Ant-Q agent System Based Path Optimization Service for a Multi-Objective Mobile Robot and Real World

Hyun-Kim

Department of Computer Engineering, School of Electronics and Information, Kyunghee University, South Korea
kimhyun @khu.ac.kr

Abstract

A multi-objective mobile robot path planning algorithm based on improved Ant-Q agent system algorithm is proposed.

The most Driver has Navigation system. It is convenient if him use the Navigation system. But, Navigation systems are not able to determine optimized driving routes considering that each driver has specific driving habits and propensities and many circumstantial changes are present in every trip. Therefore, We Propose a route recommendation system as part of the personalizing information method for navigation systems by the prey pursuit problem has been put to use in multi-agent research in addition to the food chain system using multi-agents in a virtual grid space. In this paper, We suggest a limitless space like reality and a new algorithm to better explain reality using the existing grid space and obstacle environment. We suggest New algorithm is based Ant-Q(Ant Colony System – Q learning) algorithm. Ant-Q algorithms were inspired by work on the ant system (AS), a distributed algorithm for combinatorial optimization based on the metaphor of ant colonies which was recentl proposed in (Dorigo, 1992; Dorigo, Maniezzo and Colorni, 1996).

Keywords: *multi-agent, prey pursuit problem, personalization, optimized path, dynamic environment, multi-objective mobile robot, Ant-Q, learning. Circular grid space, real world, obstacle environment*

1. Introduction

In this paper, we propose a route recommendation system as part of the personalizing information method for navigation systems applied of Ant-Q algorithm.

A personalizing information service, which provides optimization information to users, is the core technology of ubiquitous computing, and it is required to construct profiles based on individual characteristics. There are many generative methods for personal profiles, such as the method of data retrieval and accumulation of user activity data (migratory routes, movements, manipulations) and the recommendation method which utilizes data of existing users with similar propensities. In additions, each user, or driver in the case of navigation systems, has specific characteristics, such as driving lane preference.

Having agents which analyze routes based on existing navigation services is not the optimal method for determining the shortest route nor does it consider environmental variables when analyzing solutions. In real driving situations, there are environmental variables, driving environments, and specific personal driving habits that factor into the optimized route, and all of these variables change frequently.

Currently, While most drivers are driving a car are two-way data transmission and reception are possible due to the generation of communication and computer technologies.

In this paper, we suggest a heuristic approach for personal route optimization by considering a variety of environmental variables. It heuristic is applied Ant-Q. Ant-Q algorithm is the basic idea underlying this algorithm, called ant system (AS), was that of using a colony of cooperating ants to find shortest Hamiltonian tours in a weighted complete graph (the so called traveling salesman problem:TSP). A family of algorithms which strengthen the connection between RL, in particular Q-learning, and AS.

We represent dynamic environmental information as factors of the agent, ***n***, and the prey (target result) sought by the agents is ***P***. We propose a heuristic approach based on experiments in which n tracks down and captures ***P*** [1, 2, 3, 7, 8].

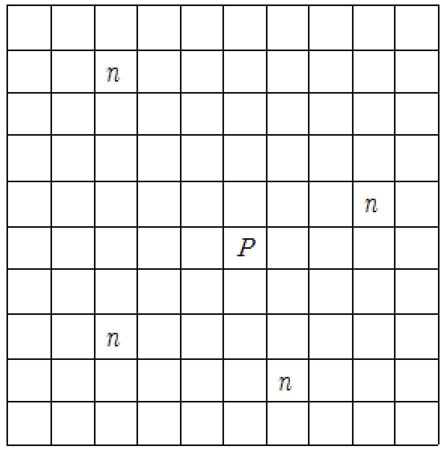

Figure 1. Prey Pursuit Problem

Our method is based on research about multi-agents developed to satisfy user needs and to solve user problems. Therefore, the multi-agent system with cooperating agents is suggested to determine solutions for complicated problems that cannot be solved by a single agent. M. Benda proposed the prey pursuit problem to express complicated reality as a typical experimental model of the efficient multi-agent system [2].

The basic experimental environment of the prey pursuit problem is that there are four agents and one target, and the agents aim to capture the target through cooperation [3]. The prey is the target, and the goal of multi-agent research is for each of the agents to reach the target efficiently with the lowest cost. However, the previous experimental environment is restricted to a space of **n*n**, leading to a limited reality, and is solved the problems by only capture prey of agents.

In this paper, we suggest a new enlarge experimental environment (the circular grid space) in an attempt to more adequately represent reality. Also, we express the relationships between agents, as well as those between prey and agents, using direction vectors in a new heuristic approach which considers distance, location, and directivity.

The remainder of this paper is organized as follows: In Section 2, the preliminaries for this problem is reviewed. The proposed Ant-Q based path optimization algorithm is described in Section 3. In Section 4, experimental results are presented with different criteria. Suggestions for future development are given in Section 5. Finally, conclusions are presented in Section 6.

2. Background

2.1 Circular Grid Space that Considers the Actual World and Obstacle

The experimental environment of the general prey pursuit problem is a restricted space on which all sides are blocked, which can result in an imperfect capture when the prey is driven in one direction. Even if an effective capture strategy was developed, the restricted space and imperfect capture of the model still lack actuality and efficiency [4, 5, 6, 7]. For this reason, we propose a continuous circular grid space that better reflects reality, as shown in Figure 1. In the real world, agents operate in an infinite space to capture prey. However, in the case that the transport speed of the prey is the same as that of the agents, if the agents have gathered in one location and the prey runs in only one direction, capture is impossible. We address problems stemming from this infinite space by using a continuous circular grid space. That is, from grid space between (0, 0) and (0, 9), we make (0, 9) that right is blocked by circular state and do (0,9) and (0,0) adjoin. And, Several obstacle have been added to this space. Because, This environment is the similar to the real world space format.

For example, the top, bottom, right, and left of (3, 3) are (2, 3), (4, 3), (3, 2) and (3, 4), respectively. However, the top, bottom, right, and left of (0, 0) are (9, 0), (1, 0), (0, 9) and (0, 1), respectively. In a multi-agent system, even if the transport speeds of the prey and the agents are the same, capture is still possible; although successful escape by the prey is also a possible outcome. We aimed to demonstrate the efficiency of the new capture strategy.

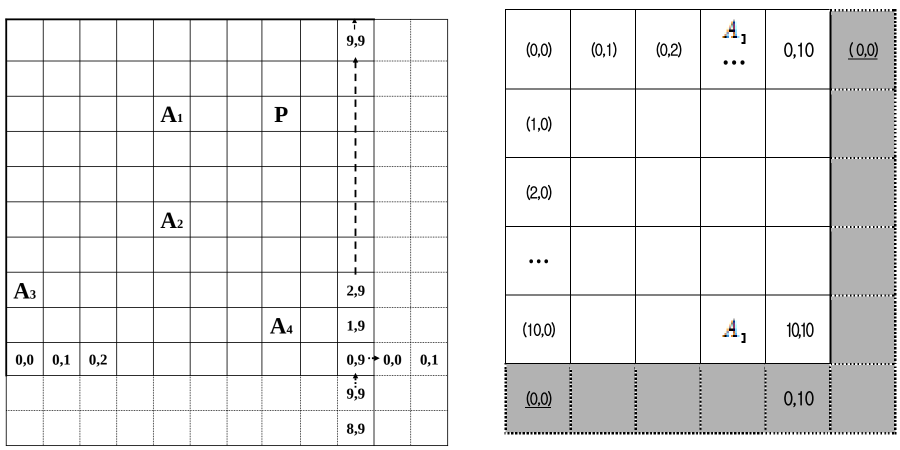

Figure 2. Circular Grid Space

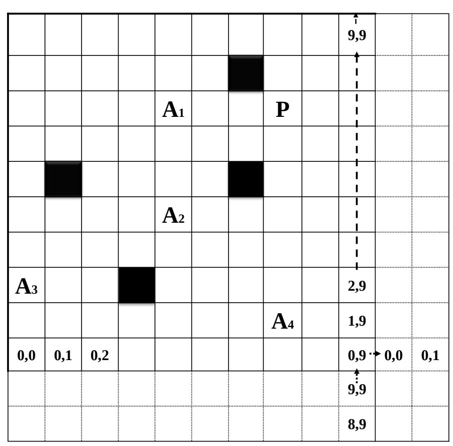

Figure 3. Circular Grid Space and Obstacle

2.2 The Ant-Q Family of Algorithm

This part introduce the Ant-Q algorithm by its application to the traveling salesman problem(TSP). Given a set of n cities, and for each pair of cities a distance d_{ij}, the TSP is stated as the problem of finding a minimal length closed tour that visits each city once. An instance of the TSP is given by a graph (N,E), where N, $|N| = n$, is the set of cities and E is the set of edges between cities (a fully connected graph in the Euclidean TSP). In case $d_{ij} \neq d_{ij}$ we have the more general asymmetric traveling salesman problem (ATSP). Ant-Q algorithms apply indifferently to both problems. Let AQ(i, j), read Ant-Q-value, be a positive real value associated to the edge (i, j). It is the Ant-Q counterpart of Q-learning Q-values, and is intended to indicate how useful it is to make move s (i.e., to go to city j) when in city i. AQ(i, j)'s are changed at run time. When Ant-Q is applied to a symmetric TSP, then AQ(i, j)=AQ(i, j).Otherwise, when Ant-Q is applied to an asymmetric TSP,AQ(i, j)'s can be different from AQ(j, i)'s. Let HE(i, j)be a heuristic value associated to edge (i, j)which allows an heuristic evaluation of which moves are better (in the TSP we chose the inverse of the distance).

$$\mathrm{s} = \begin{cases} \underset{j \in J_k(i)}{\arg\max}\left\{[AQ(i,j)]^{\delta} \cdot [HE(i,j)]^{\beta}\right\} & if \quad q \leq q_0 \\ S & otherwise \end{cases} \qquad (1)$$

$$j_k(r,u) = \frac{AQ(i,j)^{\delta} \ \cdot \ HE(i,j)^{\beta}}{\sum \ AQ(i,j)^{\delta} \ \cdot \ HE(i,j)^{\beta}}$$

Let k be an agent whose task is to make a tour: visit all the cities and return to the starting one. Associated to k there is the list $J_k(i)$ of cities still to be visited, where i is the current city. This list implements a kind of memory, and is used to constrain agents to make feasible tours, that is, tours which visit all cities once and only once. An agent k situated in city i moves to citys using the following rule, called action choice rule (or state transition rule) Eq. (1)

where δ and β are parameters which weigh the relative importance of the learned AQ-values and the heuristic values, q is a value chosen randomly with uniform probability in [0,1], q_0 ($0 \leq q_0 \leq 1$) is a parameter such that the higher q0 the smaller the probability to make a random choice, and S is a random variable selected according to a probability distribution given by a function of the AQ(i, j)'s and HE(i, j)'s, with $j \in J_k(i)$.

In equation (1), as it is the case in the following equation (3), we multiply the AQ-value AQ(i, j)by the corresponding heuristic value HE(i, j). This choice was meant to favor those AQ-values belonging to shorter edges, and was mainly motivated by our previous work on the ant system. Other composition functions, different from multiplication, are possible and will be the subject of future work.

In Ant-Q m agents cooperate to learn AQ-values such that they can favor, in probability, the discovery of good TSP solutions. AQ-values are updated by the following rule Eq.(2)

$$\mathrm{AQ(i,j)} \leftarrow (1-\alpha)\cdot AQ(i,j) + \alpha \cdot \left(\Delta AQ(i,j) + \gamma \cdot \underset{z\epsilon J_k(j)}{Max} \; AQ(i,z) \right) \quad (2)$$

The update term is composed of a reinforcement term and of the discounted evaluation of the next state. Parameters α and γ are the learning step and the discount factor. In AS, and in all the Ant-Q algorithms presented here, the reinforcement ΔAQ is always zero except after each agent has completed its tour. The update rule of formula (2) is the same as in Q-learning, except for the fact that the set of available actions in state *s*, that is, the set $J_k(i)$, is a function of the previous history of agent *k*. An iteration of the generic Ant-Q algorithm can be described in words as follows.

First, at Step 1 there is an initialization phase in which an initial value is given to AQ-values, and each agent k is placed on a city r_{k1} chosen according to some policy (discussed in Section 3). Also, the set $J_k(j$)of the still to be visited cities is initialized. Then, at Step 2, a cycle, in which each of the m agents makes a move and the AQ(i, j)'s are updated using only the discounted next state evaluation, is repeated until each agent has finished its tour and is back in the starting city. At Step 3, the length L_k of the tour done by agent k is computed, and is used to compute the delayed reinforcements ΔAQ(i, j)'s. Then AQ(i, j)'s are updated using formula (2). Finally, Step 4 checks whether a termination condition is met, and if it is not the case the algorithm returns to Step 2. Usually the termination condition is verified after a fixed number of cycles, or when no improvement is obtained for a fixed number of cycles. (In experiments in which the optimal value was known a priori the algorithm was stopped as soon as the optimum was found.)

2.3 Capture strategy using directional vectors

In infinite space, when the prey moves with equal speed in a direction opposite that of the agent, capture is impossible. As a circular grid space is used to represent an endless environment similar to infinite space, it is hard to capture using a general strategy. Therefore, we suggest a new strategy that expresses a distance relationship between prey and agents using a directional vector.

The agent confirms the positions of the other agents and the prey and moves in the capture direction. The prey calculates the distance from the agent in order to escape and moves as far as possible from the agents. If the transfer speed of the prey is the same as that of the agent, and the prey moves faster than the agents, then capture is difficult. The agents need a strategy aimed at an effective capture, and the prey needs a strategy that can recognize the neighborhood state space and utilize it to escape from the agents. We introduced a directional vector to design these strategies using a vector that reflects the distance between the agents and the prey as well as the distances between the agents.

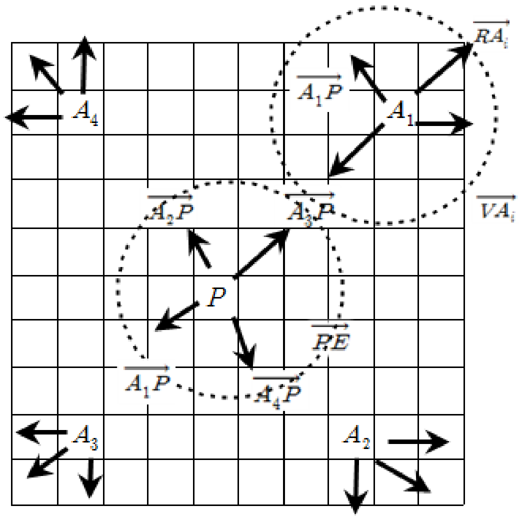

Figure 4. Transfer Strategy using Direction Vector

A. Escape Directional Vector

Prey (P) moves in the opposite direction of the agents (A_i) to escape. As the distance between the agents and the prey decreases, the transfer directional vector of the prey increases. The prey escapes from the agents using the resultant vector of Eq. (3)

$$\overrightarrow{PE} = \overrightarrow{A_1E} + \overrightarrow{A_2E} + \overrightarrow{A_3E} + \overrightarrow{A_4E} = \sum_{i=1}^{n} \overrightarrow{A_iE} \qquad (3)$$

$$\left|\overrightarrow{A_iE}\right| = \frac{1}{(Px - A_ix)^2 + (Py - A_iy)^2}$$

B. Transfer direction vector of agents to capture prey

The directional vector ($\overrightarrow{A_iP}$) reflects the distance between the prey (P) and the agent (A_i). The length of the directional vector (($\overrightarrow{A_iP}$) between each agent and the prey is inversely proportional to the distance between them. However, the problem of collisions between agents is possible during movement. We address this problem by designing a vector function that considers the distance relationship between the agents, as in Eq. (4). This vector that incurs more weight in the opposite direction as it approaches another agent. The repulsion vector ($\overrightarrow{RA_i}$) value of agent (A_i) is expressed as the sum of the directional vectors between each agent.

$$\overrightarrow{RA_i} = \sum_{j \neq i}^{n} \overrightarrow{A_jA_i}$$

$$\left|\overrightarrow{A_jA_i}\right| = \frac{1}{(A_jx - A_ix)^2 + (A_iy - A_jy)^2} \qquad (4)$$

Here $\overrightarrow{A_jA_i}$ is the directional vector between agent (A_j) and agent (A_i) which is more heavily weighted in the opposite direction as the distance between the agents decreases. The length of the directional vector (A_jA_i) between agents is inversely proportional to the distance between agents. When this vector is applied, the distance between agents increases, thus solving the problem of collisions between agents. The transfer directional vector ($\overrightarrow{VA_i}$) of an agent is the resultant vector of A_iP and $\overrightarrow{RA_i}$ expressed in Eq. (5).

$$\overrightarrow{VA_i} = \alpha \cdot \overrightarrow{A_iP} + \overrightarrow{RA_i} \qquad (5)$$

In the design of the transfer directional vector ($\overrightarrow{VA_i}$) of the agent, we applied the weighted values to $\overrightarrow{A_iP}$ and $\overrightarrow{RA_i}$ as in Eq. 4. It was deemed suitable for agents to move in a direction according to weighted values (α, β) such that α > β because it is more important to prevent collisions between agents than it is to move toward the prey.

$$\overrightarrow{VA_i} = \alpha \cdot \overrightarrow{A_iP} + \beta \cdot \overrightarrow{RA_i} \qquad (6)$$

In the prey pursuit game, the prey or agents may choose not to move. Agents may choose not to move if the transfer direction moves them farther from the prey than is the current position, as determined by the $\overrightarrow{VA_i}$ value. Agents move to the next position if conflict occurs. Figure 4 shows the transfer strategy using the directional vector, and Figure 3 shows the prey pursuit problem algorithm that uses the directional vector.

3. Estimation

Usually, We are applied AQ-value the performance evaluation is based on the capture success rate and each agent's state transition in the prey pursuit problem.

In this paper, we wish to achieve a 100% capture rate in a collision-avoidance strategy between agents using the standard of estimation through a new type of circular grid space and obstacle environment. Also, this experimental environment removes the element of imperfect capture considering actuality and the initial positions of the prey and agents are randomly assigned. The final result illustrated the perfect capture of prey.

3.1. Environment Transformation

Table 1. The Result of the Prey Pursuit Problem with Weighted Values

		50×50		100×100		100×100(add obstacle)	
(α)	**(β)**	**Capture probability**	**State transition**	**Capture probability**	**State transition**	**Capture probability**	**State transition**
0.1	**0.1**	**3 %**	**407**	**1 %**	**2230**	**0.2 %**	**5072**
0.2	**0.1**	**42 %**	**398**	**19 %**	**2251**	**8 %**	**4564**
0.3	**0.1**	**55 %**	**391**	**26 %**	**2223**	**11 %**	**4020**
0.4	**0.1**	**78 %**	**379**	**32 %**	**2210**	**23 %**	**3795**
0.5	**0.1**	**84 %**	**327**	**54 %**	**2182**	**31 %**	**3757**
0.6	**0.1**	**92 %**	**299**	**62 %**	**2171**	**42 %**	**3593**

0.7	**0.1**	**100 %**	**263**	**75 %**	**2144**	**50 %**	**3610**
0.8	**0.1**	**100 %**	**264**	**81 %**	**2130**	**62 %**	**3321**
0.9	**0.1**	**100 %**	**261**	**92 %**	**2128**	**69 %**	**3212**
1	**0.1**	**100 %**	**262**	**100 %**	**2111**	**89 %**	**2902**

Examining the experimental results in Table 1, we illustrated the efficiency of the algorithm that considers the directional vector affected by weighted values (α, β) of prey and agents.

Various results were observed based on the size of the grid space. When α was greater than 4 in a 100×100 grid space, obstacle environment, and more than 0.7, 0.1 from **Table 1**, an effective state transition and a 100% capture were observed. These results indicate that **α is impotent in this paper.** Table 2 compares the original agent control strategy and the reinforcement learning strategy in the new environment [5,6,7,8,9,10].

/* Initialization phase */
Set initial positions for the prey and four agents.
/* Main algorithm */
FOR
{
1. Initialize weighted agent directional values α and β
// α : agent to prey , β : agent to another agent
2. Prey applies the prey escape vector $\overrightarrow{PE}$ using (Eq. 3) to choose the escape position.
3. Agent (A_i) applies the vector $\overrightarrow{RA_i}$ using (Eq. 4) to prevent a conflict with each other agent.
4. Applies weighted values α and β to (Eq. 3), and Agent (A_i) applies the real moving vector $\overrightarrow{VA_i}$ using (Eq. 4) to move to a position to capture the prey
}

Figure 4. Prey Pursuit Algorithm

Table 2. Applied Original Strategy in the New Environment

Grid Space of Circular Type	**100×100**	
Agent control strategy	*Capture probability*	2.2 %
	State transition	32.6 %
Reinforcement learning strategy	*Capture probability*	41 %
	State transition	34.2 %

4. Conclusions and Future Works

Various experiments in multi-agent systems are carried out in current research. The prey pursuit problem is used often in these various experiments because it is able to adequately represent reality. However, an experiment that is achieved in a limited grid space differs significantly from the actual world, which is an environment of infinite space. We need to achieve more research through an experimental environment that can express complex reality. The circular grid space and circular grid obstacle environment proposed in this paper is a continuous space that adequately represents the real world.

We can prevent collisions between agents and incur an efficient capture through the heuristic using the Ant-Q and directional vector in the proposed experimental environment. Also, through the capture strategy that uses the proposed directional vector, we can verify the efficiency in a random arrangement state.

However, because the probability of prey capture is low in specific situations (such as those in which the initial position is assigned to one direction), more research is required to increase this probability. Also, further research is required to develop an agent system that can be applied to the new environment.

References

[1] Kim S, Kim BC and Yoon BJ, “Multi-agent Coordination Strategy using Reinforcement Learning”, The Proceedings of the 22th KIPS Fall Conference, vol. 7, no. 2, **(2000)**, pp. 285-288.

[2] Stone P and Veloso M, "Multiagent System: A Survey from a Machine Learning, "Technical Report CMU-CS-97-193, The University of Carnegie Mellon, **(1997)** December.

[3] Haynes T and Sen S, "Evolving behavioral strategies in predators and prey", In Gerhard Weiband Sandip Sen, editors, Adaptation and Learning in Muiltiagent Systems, Springer Verlag, Berlin, **(1996)**.

[4] Sen S, Sekaran M and Hale J, “Learning to coordinate without sharing information”, National Conference on Artificial Intelligence, **(1994)** July, pp. 426-431.

[5] Stephens LM, Merx MB, “The effect of agent control strategy on the performance of a DAI pursuit problem”, In Proceeding of the 1990 Distributed AI Workshop, **(1990)** October.

[6] de Jong E, “Multi-Agent Coordination by Communication of Evaluation”, In Proceeding of the 8th Europeon Workshop on MAAMAW 97, **(1997)**.

[7] Lee H and Kim BC, “Multiagent Control Strategy using Reinforcement Learning”, The KIPS Transactions 10-B 3, **(2003)**, June.

[8] Levy and Rosenschein JS, "A game theoretic approach to the pursuit problem", In Working Papers of the 11th International Workshop on Distributed Artificial Intelligence, **(1992)** February.

[9] Cammarata S, McArthur D and Steeb, "Strategies of Cooperation in Distributed Problem Solving", Proceedings of Eighth International Joint Conference on Artificial Intelligence, Karlsruhe West Germany, **(1993)** August.

[10] Stephens LM and Merx MB, "The effect of agent control strategy on the performance of a DAI pursuit problem," In Proceeding of the 1990 Distributed AI Workshop, **(1990)** October.

Authors

Ph.D. Kim Hyun

Fields of Research

Computer and Information Science , Mobile Robot, Multi Robot Agent, Path planning

Experience

FM Robotics & Kyunghee University

Education

2005 – 2011 Artificial intelligence, computer engineering, Kyunghee University, Seoul, Korea