Study on Reinforcement Learning and Its Application for Delivery problem

Pham Hoang Duc Nha · Phan Quoc Minh · Hoang Tuan Anh
FPT University, Software Engineering, Ho Chi Minh City, Vietnam
Phạm Hoàng Đức Nhã (Leader) nhaphdse61869@fpt.edu.vn
Phan Quốc Minh (Contributor) minhpqse62019@fpt.edu.vn
Hoàng Tuấn Anh (Contributor) anhhtse62103@fpt.edu.vn
Đoàn Nguyễn Thành Hoà (Advisor) hoadnt@fe.edu.vn

***Abstract* – To determine the shortest path are one of the most addressed problems in the field of computer science. Many algorithms are currently available to solve the problems. However, consider between performance and accuracy of algorithms is also the dilemma which need to be solved.**
Since exact algorithms are too resources-consuming, improving heuristic solutions to obtain as closest as possible the optimal result yet keeping performance to be within the acceptable threshold is currently the better approach to solve the problems. Our thesis mainly focuses on how using reinforcement learning and to be more specific, Ant-Q would improve the accuracy of heuristic approach. Moreover, we also compare results achieved by Ant-Q with other exact and heuristic algorithms. To address further, we also apply clustering the big solution into smaller chunks as to divide and conquer the big problem performance-wise as well as to fit the current situation of the delivery problem when multiple deliverymen involve in one delivery session.

Keywords—Reinforcement Learning, Q-learning, Ant-Q, Clustering, TSP

I. INTRODUCTION

Based on underlying theory of the Ant Colony Optimization (ACO) introduced by Dorigo, Maniezzo and Colorni (Dorigo, 1992; Dorigo, Maniezzo and Colorni 1996; Colorni, Maniezzo and Dorigo, 1991; 1992) and the concept of Q-learning, Ant-Q is a reinforcement learning method where cooperating agents try to find shortest Hamiltonian tours in a weighted complete graph. By combining the 2 approaches, Ant-Q not only improve the accuracy but also prevent itself to converge toward local optimal. The basic idea of the algorithm will be introduced in section II. The visualization of the algorithm as well as the how data is structured will be included in section III. In section IV, there is the analysis of the parameters used in the algorithm. Furthermore, we will introduce how the big problem is segmented using the K-means algorithm. In section V, the experiment results will be shown to compare between Ant-Q and other heuristic algorithms. Afterwards, we conclude briefly the discussion and address the future work of this algorithm.

II. ANT-Q AND K-MEANS++ ALGORITHM

1) *Ant-Q Algorithm*

Given a set of cities where distance between each pair of cities is $d_{rs}d_{rs}$, the purpose of solving the Travelling Salesman Problem is to find the minimal closed tour where each city is only visited once. Ant-Q algorithm can be similarly applied to both symmetric TSP (where $d_{rs} = d_{sr}$) or asymmetric TSP or ATSP (where $d_{rs} \neq d_{sr}$).

The choice of posible next cities to visit is described as below:

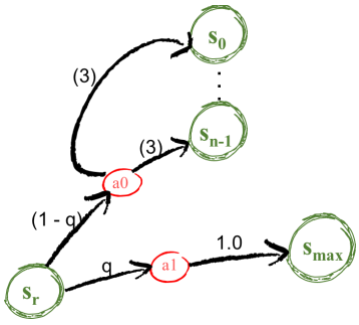

Figure 1. The MDP above describes the state transition rule of an agent from one city to another.

a_1, a_2 denotes the actions which an agent k to perform state transition from s, whereas the policy of choosing actions is represented by the probability of q_0.

s_o represent the current state of an agent k who standing in city r. With the probability of q, agent may choose action a1 which leads to the state s_{max} where s_{max} is the state where it holds the maximum value formulated in equation (1) comparing to other values in S where S is the set of cities available to be visited from city r. Otherwise, the next city will be chosen randomly from the set S where each city holds a probability of choices described in equation (3).

$$s = \begin{cases} argmax_{u \in S_k(r)} [AQ(r,s)]^{\delta} \cdot [HE(r,s)]^{\beta}, & q \leq q_0 \\ S, & q > q_0 \end{cases} \quad (1)$$

where δ and β are parameters which weigh the relative importance of the learned *AQ*-values and the heuristic values.

q is a random value which is uniform distributed in the range of [0,1] indicating how the selection of action is biased toward the argmax function with the probability of q_0.

As an agent moves from one state to another, it will interact with the environment by leaving "trails" which reinforce other agents' decisions of choosing cities. This is presented by the update of the Ant-Q values matrix:

$$AQ(r,s) \leftarrow (1\text{-}\alpha) \cdot AQ(r,s) + \alpha \cdot (\Delta AQ(r,s) + \gamma \cdot \max AQ_{u \in S_k(s)}(s,u)) \quad (2)$$

The update term consists of the reinforcement from previous transition steps to reinforce long-term rewards and a discount evaluation of the short-term reward.

The learning rate α[0,1] indicates how an agent prefers learning from previous states (exploitation) to learning from trial and error as it is biased toward the Ant-Q value (exploration).

The discount factor γ[0, 1]is used to balance the value of the learning history of an agent via the maximum value of AQ(s, u) where s is the next city to be visited and u belongs to the set of cities available from s. s is the city which is selected by the action the agent has taken

The ΔAQ(r, s) is called the delayed Ant-Q value which has a value of 0 unless an agent finished its tour. The calculation of ΔAQ(r, s) will be present in equation (4).

The probability of the selection of each city in the exploration phase is indicated by the following equation:

$$p_k(r,s) = \begin{cases} \frac{[AQ(r,s)]^{\delta} \cdot [HE(r,s)]^{\beta}}{\sum_{u \in S_k(r)} [AQ(r,u)]^{\delta} \cdot [HE(r,u)]^{\beta}}, & s \in S_k(r) \\ 0, & s \notin S_k(r) \end{cases} \quad (3)$$

$S_k(r)$ denotes the set of cities allowed to visit from r.

We also try to obtain the random action choice rule in Q-Learning. The uniform distribution is tested for selecting next city. The comparison of the 2 policies is shown in the following table:

	Random choice rule with uniform distribution			Random choice rule distributed with equation (3)		
	Best tour	Mean	Std. Dev	Best tour	Mean	Std. Dev
Oliver30	425.02	426.24	1.22	423.74	424.44	0.62
ry48p	14848	16246	523	14422	14624	175

Table 1. Ant-Q Performance in the change of action selection rule

Delayed reinforcement values are considered between two types: Iteration-best and Global-best.

$$\Delta AQ_k(r,s) = \begin{cases} \frac{W}{L_k} & \text{if } (r,s) \in \text{tour done by agent k} \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

k is the agent who completed the tour with the best result.

$L_k L_k$is either the global-best tour or the current iteration-best tour which is completed by agent k.

W is called weight value which is derived from the formula of updating global pheromone in Ant Colony Optimization algorithms. This value is fixed with the value of 10.

As being mentioned before, the calculation of ΔAQ(r, s) is only available once in each iteration at the time all agents have finished their tours. In iteration-best, $L_k L_k$is the length of the best tour in each iteration while in global-best, in each iteration, the value of L_k $L_k$is equal to the best tour length so far.

2) K-means++ Algorithm

a) K-means

Clustering is one of the classic problems in machine learning and computational geometry. The problem is computationally difficult (NP-hard); however, there are efficient heuristic algorithms that are used to find a local optimum. Given given an integer k and a set of n data points in R^d, the clustering problem is stated as the problem of splitting these point into k similar groups.

The k-means algorithm is one of the oldest and most popular clustering algorithm. It is proposed by Lloyd and still very widely used today. The objective of K-means is partition n data points into k clusters in which each data point belongs to the cluster with the nearest center.

K-means algorithm begins with k arbitrary "centers", which are chosen uniformly at random from the data points. Each data point is then assigned to the nearest center, and each center is recomputed as the center of mass of all points assigned to it. These last two steps are repeated until the assignments no longer change. However, the result of algorithm can be arbitrarily bad compared to the optimal clustering.

b) K-means++

The k-means++ algorithm addresses that problem by choosing random starting centers with very specific probabilities before proceeding with the standard k-means optimization iterations. we are given an integer *k* and a set of *n* data points $X \subset R^d$. We wish to choose *k* centers *C*. It works as follows:

1a. Choose an initial center c_1 uniformly at random from X .

1b. Choose a new center c_i , choosing $x \in X$ with probability with $D(x)$ is the distance from a data point x to the closest center we have already chosen.

1c. Repeat Step 1b until we have chosen k centers altogether.

2. For each $i \in \{1, \ldots, k\}$, set the cluster C_i to be the set of points in X that are closer to c_i than they are to c_j for all .

3. For each $i \in \{1, \ldots, k\}$, set c_i to be the center of mass of all points in C_i:

4. Repeat Steps 2 and 3 until C no longer changes.

III. PLAN IMPLEMENTATION

We decide to use 2 2D-arrays to store heuristics values and ant-q values. As an edge consists of 2 nodes, the equivalent heuristic values and the ant-q value will be stored as HE[r][s], and AQ[r][s] with r as starting node and s as end node, respectively.

The flow of the AntQ algorithm can be described with the figure 2.

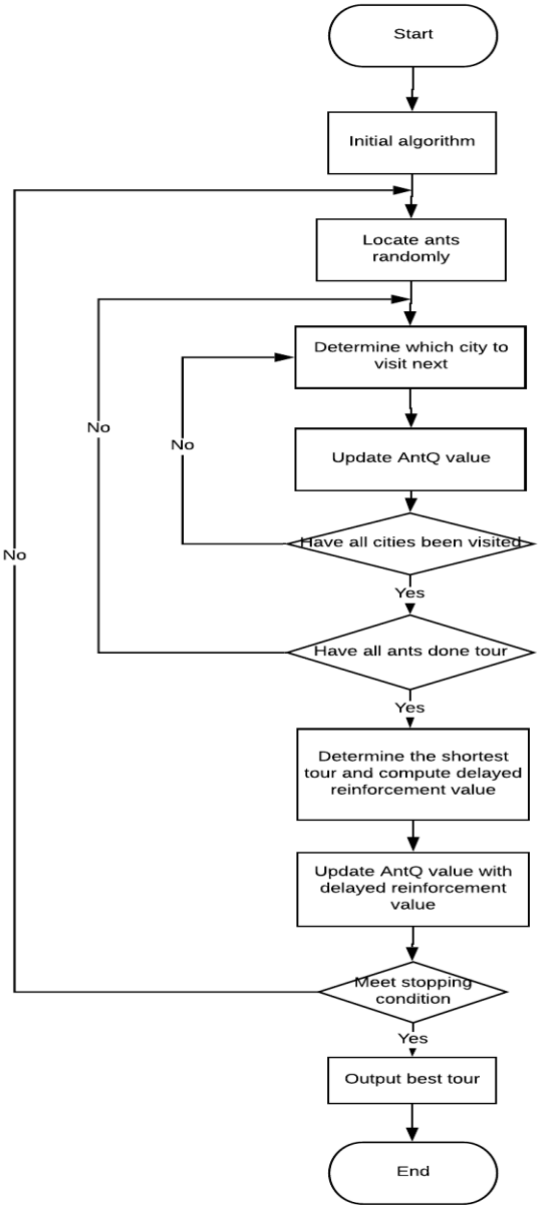

Figure 2. Ant-Q Algorithm workflow

IV. ANALYSIS

The experimentally found best values are δ=1, β=2,

$q0$=0.9, α=0.1, γ=[0.2, 0.6], W=10, AQ_0= ~0. The number of ants should be in the range of [0.6n, n]. However, to optimize the result accuracy, we decided to use m=n. Test problem are Oliver30, the 30 nodes symmetrical TSP and ry48p, the 48 nodes asymmetrical TSP.

The reason to choose such parameters will be determined with the following experiment results.

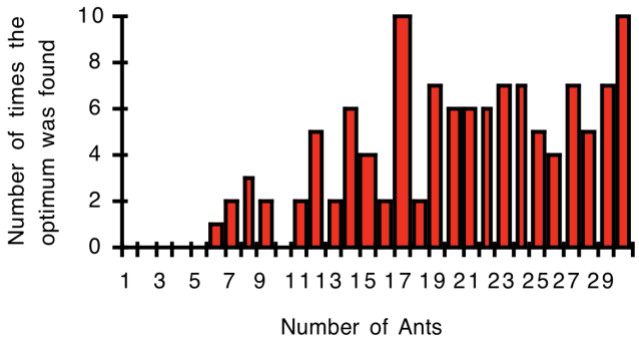

Figure 3. Ant-Q Performance for different value of m. Test tour: Oliver30. Average on 30 trials, 200 iterations each. q_0 = .9, α = .1, γ= .4.

q_0 is a parameter indicates the probability of choosing between best learned values and values in the exploration phase. During testing, it is shown that, the higher the q_0, the better the results. However, the results obtained with q_0 = [0.9, 1] drastically decrease in accuracy since the algorithm converge toward the current best-known result.

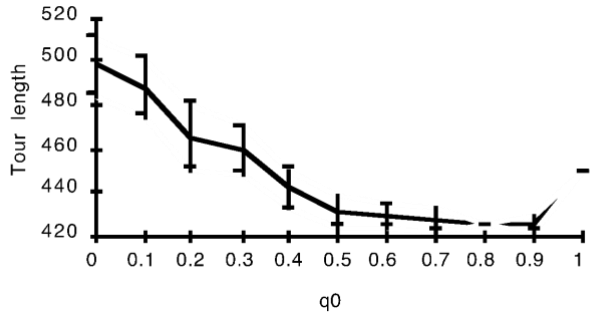

Figure 4. Ant-Q performance with the change of q0. Test problem: Oliver30. Std dev. and Mean Length in 15 trials, 200 iterations each.

During the testing phase, interesting to observe that, the optimal parameters to obtain the best results remain the same in all symmetric and asymmetric problems.

Since the change of α does not clearly affect the result of the simple symmetric TSP like Oliver30, we decide to use the ry48p ATSP to better distinguish the results of the algorithm with different values of α and γ.

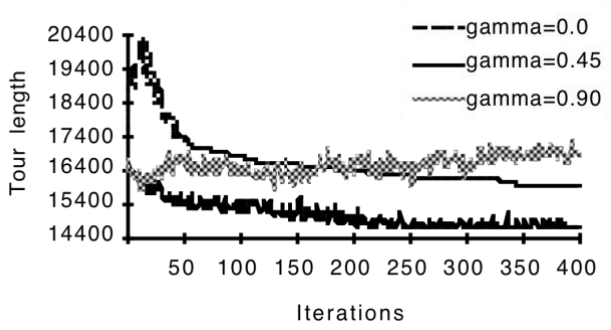

Figure 5. Ant-Q Performance affected by the change of gamma. Test problem: ry48p. Average in 5 trials. 500 iterations each.

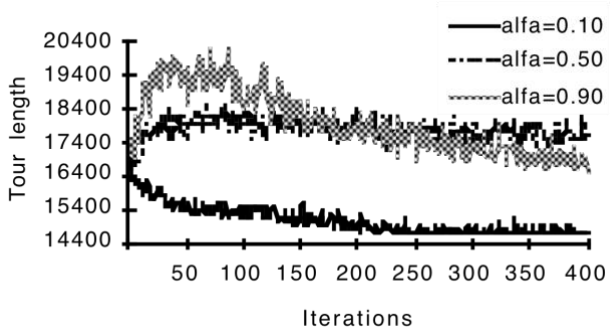

Figure 6. Ant-Q Performance affected by the change of alpha. Test problem: ry48p. Average in 5 trials. 500 iterations each.

All analyzed graphs above are referenced from the experimental research by Gambardella L. and M. Dorigo (Gambardella L. and M. Dorigo, 1995. Ant-Q: A Reinforcement Learning approach to the traveling salesman problem). We will introduce our results produced by our implementation in section V.

V. EXPERIMENTAL RESULTS AND CONCLUSION

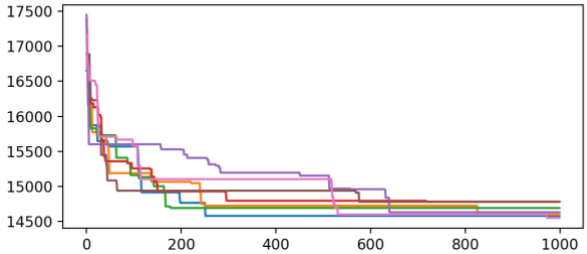

Figure 7. Ant-Q results in Problem:ry48p, 7 trials, 1000 iteration each. All results are in [14500, 15000]

For the ry48p ATSP problem, the best result is 14422. Our best result for this problem so far is 14446. Above is a chart indicating the result produced by 7 trials.

With the solutions lying between 14446 and 14943. The algorithm has proved that it is consistent enough to produce decent results.

It is observable that the solution for this specific problem is found when the iteration reach around the number of 700.

Moreover we also experiment on the delayed Ant-Q value updating policy using iteration-best and global-best:

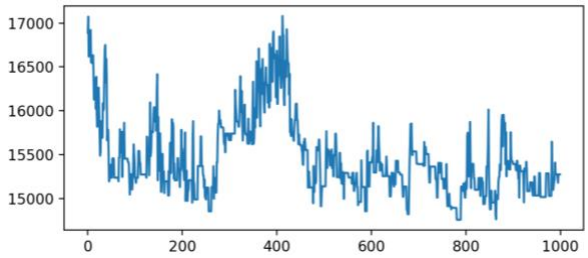

Figure 8. Best results produced by each iteration using Iteration-best policy.

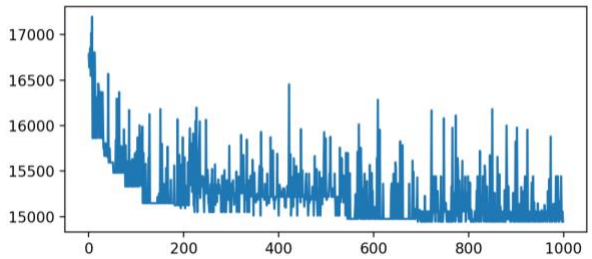

Figure 9. Best results produced by each iteration using Global-best policy.

The best route found in each iteration when using the iteration-best approach oscillates more drastically than the global-best approach. This indicates the when using iteration-best approach, agents has a tendency to widen their search range which can avoid the likelihood of converging toward local optima. Therefore, agents in the former approach are more likely to produce better and more consistent final results.

We compared the average behavior of AntQ with following well-known heuristic algorithms: Simulated Annealing(SA) and Ant Colony Optimization(ACO). The comparison was run "on ry48p"(Dataset 1) dataset of Fischetti and "att48"(Dataset 2) dataset of Padberg Rinaldi. Dataset 1 is the ATSP problems and dataset 2 is the TSP problems.

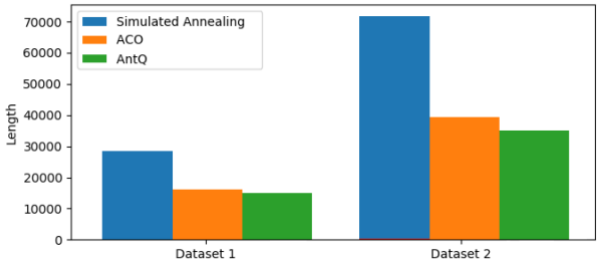

Figure 10. Best results produced by Ant-Q, ACO and SA algorithm

The obtained result is show that Ant-Q was almost always the best performing algorithm. Although ATSP problems are much more difficult than the TSP problems, Ant-Q still has better result than other heuristic algorithms.

In conclusion, the Ant-Q algorithm can produce decent solutions for both TSP and ATSP problems. The results obtained from the experiment work show that applying reinforcement learning to solving the delivery problem will produce the most good results. Further, we plan to extend the Q-learning applications, applying it to other optimization problems.

ACKNOWLEDGMENT

We would like to send our appreciation to Mr. Doan Nguyen Thanh Hoa for providing essential materials for our study on the problem.

REFERENCES

[1] Colorni A., M. Dorigo and V. Maniezzo, 1991. Distributed Optimization by Ant Colonies. *Proceedings of ECAL91 - European Conference on Artificial Life*, Paris, France, F.Varela and P.Bourgine (Eds.), Elsevier Publishing, 134–142.

[2] Colorni A., M. Dorigo and V. Maniezzo, 1992. An Investigation of some Properties of an Ant Algorithm. *Proceedings of the Parallel Problem Solving from Nature Conference (PPSN 92)*, Brussels, Belgium, R.Männer and B.Manderick (Eds.), Elsevier Publishing, 509–520.

[3] Gambardella L. and M. Dorigo, 1995. Ant-Q: A Reinforcement Learning approach to the traveling salesman problem. *Proceedings of ML-95, Twelfth International Conference on Machine Learning*, Tahoe City, CA, A. Prieditis and S. Russell (Eds.), Morgan Kaufmann, 252–260.

[4] Péter Stefán, László Monostori, Ferenc Erdélyi. Reinforcement learning for solving shortest-path and dynamic scheduling problems.

[5] Csaba Szepesvásri, 2009. Algorithms for Reinforcement Learning. *Draft of the lecture published in the Synthesis Lectures on Artificial Intelligence and Machine Learning series by Morgan & Claypool Publishers*

[6] David Arthur, Sergei Vassilvitskii. k-means++: The Advantages of Careful Seeding