

The Improved Ant Colony Optimization for the Multi-Dominated QoE Evaluation Point Selection Problem

Leixia Tian^{1,2}, Wenguo Yang^{1,2}, Suixiang Gao^{1,2}

¹(School of Mathematical Sciences, University of Chinese Academy of Sciences, China)

²(Key Laboratory of Big Data Mining and Knowledge Management, University of Chinese Academy of Sciences, China)

Abstract: *Quality of Experience (QoE) evaluation, compared with Quality of Service (QoS), can better reflect the quality of the network by considering the end-user's experience. Based on the number of users and the difference of user's behavior, the operators need to select some candidate test points which can perfectly represent all the users, and use the selected test points to simulate the user's behavior in the network. Due to users' different behavior, every candidate test point dominates different users' different behavior, which is described by corresponding dominated matrixes. In this article we present the multi-dominated QoE evaluation point selection problem and establish a 0-1 integer linear programming model. Because this problem is NP-hard, we adopt an improved ant colony optimization algorithm to solve it. And three methods of heuristic information description are proposed in the improved ACO. In order to demonstrate the effectiveness of the proposed method, some instance with 500 user points are generated and the performance of improved ACOs with three heuristic functions are given. Simulation results show that the quality of solution provided by the improved ACO has improved by 3.5% and 10% respectively compared with ACO and greedy algorithm when all the users are dominated. What's more, the average convergence speed of improved ACOs has increased by 88%, among them H1-IMACO reaches 94% and the average elapsed (running) time is significantly reduced by 57.6%.*

Keywords: *QoE evaluation, multi-dominated point selection, 0-1 integer linear programming, the improved Ant Colony Optimization, heuristic information*

I. Introduction

As the Internet constantly deeply influences people's lives, people don't focus on how to access network resources any more, but how to get better network resources. As a result, the satisfaction degree of the user's experience is the greatest concern for the current network service providers. People used to improve the network's performance and the overall utilization through improving bandwidth, latency, jitter and packet loss, which may affect the network's quality of service (QoS) in network transport. However, these studies are all about the network infrastructure and they cannot fulfill the real user's experience. QoE evaluation from the perspective of the user brings in the quality of the user's experience to describe the performance of the network, that is to hope users evaluate their own experience evaluation every time. [1,2] For example, there are a lot of domestic video sites such as sohu, iQIYI, youku and so on. Assume the user will assign a value to his every experience according to his own satisfaction degree after browsing a video website. It is easy to know that different users have different satisfaction with the same video site, and the same user satisfactions of different video sites are also different. However, not every user will comment on his experience and also a user will not comment his every experience. To get the quality of network accurately we want to simulate users' behavior and give the evaluation. But It needs not only plenty of manpower and material, but also enormous and continuous financial support upon this procedure. So we need to find a way to select part of the test points not only instead of all the users as many as possible but comprehensively considering all the different user behaviors.

Give a network only with the candidate test nodes (like workstation, server, network equipment) and user nodes (like terminal users). If a candidate node can monitor or serve some user nodes, we call the candidate node can dominate those user nodes. Here every user has kinds of behavior. Under different user behavior the same candidate node dominates different subset of user nodes. This article is to select some candidate nodes as less as possible that can dominate all the user nodes under every user behavior. It is easy to find that when there is only one user behavior, this problem can be described as a set-covering problem (SCP). So this paper tries to design an algorithm solving this problem from algorithms using in SCP. As we all know, the classical SCP has been proved to be a NP-hard problem, which cannot be solved in polynomial time. [3] Although no exact algorithms are for more complex or larger SCP, at present most researches use approximate heuristic algorithms to optimize it and get better results, and one of them is the ant colony optimization. Ant Colony Optimization (ACO) is a kind of evolutionary algorithms based on swarm intelligence. It pays attention to the cooperation of ant individuals and use the mechanism of feedback pheromones quickly and efficiently to search the optimal

solution. [4,5,6] Ant colony algorithm has successfully applied many complicated combinatorial optimization problems. [7] And its excellent optimization ability provides a new train of thought when solving SCP. Most variants of ACO have fresh perspectives from the influence of defining the heuristic information [8,9], the new transition rule when selecting the next point [10] and other new ideas of designing feasible solutions [11,12]. Those are also the aspects we consider to design an improved ACO to solve the point selection problem under many user behaviors. What's more, the setting of parameters in ACO is also an important matter [13,14], and documents [15,16,17] give some strategies when tuning them.

The classical SCP is for one certain coverage condition, but this article proposes to combine many coverage condition together based on users' different behaviors, what's more, for different user behavior the candidate node has different coverage. To synthetically consider these behaviors and select as less as possible candidate points this article expand from the following steps. Firstly, a network topology is given with the candidate test points and user points. If the candidate test point can dominate a user point, there is an edge between them. And under different user behavior there is a dominated matrix. This article puts forward the multi-dominated QoE evaluation point selection problem, called Multi-Dominated Point Selection Problem (MDPSP). Then from aspects of the heuristic information, the new transition rule and adaptive mechanisms this article design an improved ACO to solve MDPSP, which improved the ACO's shortcoming of slow speed of convergence and easily getting local optimum solution. Finally, utilize an instance generated by MATLAB to test the performance of the improved ACO and compared with the solution of ACO and greedy algorithm it achieves good results. The rest of the paper is organized as follows. In section 2 we give the mathematical description of the multi-dominant QoE evaluation point selection problem and establish 0-1 integer linear programming. In section 3 we give the outline of the improved ACO when using in MDPSP. Section 4 represents three heuristic function that is important for the ant's selection strategy. In section 5 we firstly set the optimal combinations of parameters for improved ACOs with three heuristic functions. Secondly using them solve the multi-dominant point selection problem and analyze the performance of improved ACO compared with greedy algorithm and traditional ACO. Section 6 concludes the article.

II. Problem Description and mathematical model

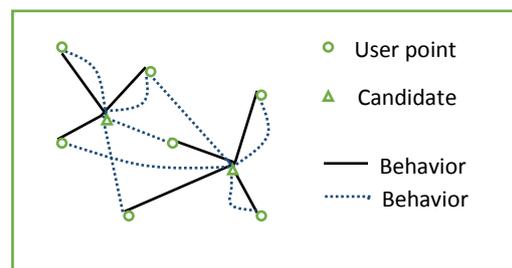
In a network exist workstation, server, network equipment and terminal users. And one workstation, sever or network equipment can monitor or server terminal users in a certain distance and in this case the operator can simulate some users' different behavior on it to give the evaluation. The final goal is to select parts of them as less as possible and to synthetically consider all the user behaviors that sometimes have different importance for operators. This article gives a network topology with two kinds of points -- candidate test points (like workstation, server, network equipment), denoted as $Y = \{y_j\}, j \in J = \{1, 2, \dots, m\}$ and user points (terminal users), denoted as $X = \{x_i\}, i \in I = \{1, 2, \dots, n\}$ and candidate test point can connect with user points. If a candidate test point y_j can connect some user points $\{x_{i_1}, x_{i_2}, \dots, x_{i_h}\} \subseteq X$, we say y_j dominates $\{x_{i_1}, x_{i_2}, \dots, x_{i_h}\}$ and there is an edge between y_j and $x_{i_{h_0}}, h_0 \in \{1, 2, \dots, h\}$. The set of users' different behavior is $U = \{1, 2, \dots, l\}$ and for one behavior t there is a dominate matrix $D_u = [d_{ij}^u], u \in U, i \in I, j \in J$, where

$$d_{ij}^u = \begin{cases} 1, & \text{if } x_i \text{ can be dominated by } y_j \\ 0, & \text{otherwise} \end{cases}$$

To better understand this model Fig.1 is given for $l = 2$. And there are seven user points and two candidate test points. At this time, we get

$$D1 = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}, D2 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}$$

Fig.1 A simple example



The problem aims to select some y_j as less as possible, which can dominate X under every dominate matrix, so this article represents the Multi-Dominated QoE evaluation Point Selection Problem, short for MDPSP. It can be formally formulated as a 0-1 integer linear program:

$$\text{Minimize } \sum_{j=1}^m y_j \tag{1}$$

$$\text{Subject to } \sum_{j=1}^m d_{ij}^u y_j \geq 1, \forall i \in I, \forall u \in U \tag{2}$$

$$y_j \in \{0, 1\}, \forall j \in J \tag{3}$$

where $y_j = 1$ represents that the candidate point y_j is in the solution, otherwise $y_j = 0$. Equation (2) ensures each user point can be dominated by at least one candidate point for every user behavior. MDPSP and SCP have the similar goal to find the least points or subsets to dominate all the members of a given set. But it's different for them when consider the number of dominate matrices in the process of generating solution. SCP is a special case of MDPSP and the latter is more complex. As is known to all, SCP is a NP-hard problem, which cannot be solved by polynomial algorithms, so the difficulty to solve MDPSP is no less than it. Currently researchers mainly use some approximate algorithms or heuristic algorithms to solve SCP. The ant colony optimization is one of them, which has been successfully applied in an instance of SCP. In this paper, we also try to adopt ACO and improve it to solve the MDPSP.

III. Improved ACO for MDPSP

Ant Colony Optimization (ACO) is a kind of intelligent bionic algorithm through mimicking natural foraging behavior of ants. The colony-level behavior on the exploitation of positive feedback can be used by ants to find the shortest path between a food source and their nest. This article designs an improved ACO to solve the MDPSP. In the improved ACO, artificial ants are introduced to search the least candidate points to dominate all the user points for every user behavior through simulating the foraging behavior of real ants. At the beginning, the search process in the improved ACO is composed of a series of iterations, and the algorithm sets a maximal iteration number. Then in an iteration, the ants are randomly distributed in candidate points. The ant selects the next candidate point into its solution with a hybrid transition rule, which associated with pheromone trails and heuristic information distributed in candidate points. [18,19] The ant will not select a candidate point any more until candidate points it selected into the solution can dominate all user points for every user behavior and the ant gets a feasible solution. And after an iteration every ant obtains a feasible solution. While only local optimal ones among them are recorded, which have the least number of candidate points in the solution. Based on them ants will give feedback by depositing pheromone on the candidate points which are in these local solutions. And the pheromones in every candidate points are accumulated after an iteration during the whole search process. Thus these candidate points will receive a higher amount of pheromone and will be more likely to be selected by ants in the future iterations of the algorithm. To avoid the search getting stuck, all the pheromones are decreased by an evaporation factor before they get reinforced. In the improved ACO the evaporation factor adopts adaptive mechanism, that is, it will vary with the local optimal solution. The algorithm loops like this getting local optimal solutions and finally obtains the optimal one among these local optimum. The algorithm ends when a maximal number of iterations have been performed or the local optimal solution has not been changed consecutively for a certain number.

3.1 Outline Of The Improved Aco For MdpSP

When the improved ACO is applied to MDPSP, u dominated matrixes should be emphatically considered. And this information is showed in heuristic information η_j distributed in candidate point y_j . The heuristic information η_j and pheromone trails τ_j respectively represent the heuristic desirability and learned desirability of including candidate point y_j into an ant's solution. The hybrid selection strategy works based on their synthetic value. While constructing a feasible solution, each ant starts with an empty set and selects candidate points iteratively until the whole user set X is dominated. In each iteration we set the total number of ants is m_0 , so after an iteration m_0 feasible solutions will be obtained. Then the algorithm records the local optimums among them to update the pheromone trails. Set the maximal iterations is NC_{max} and the number of unchangeable local solution is n_0 . Then the outline of improved ACO for MDPSP is given in Table 1.

3.2 The Hybrid Selection Strategy

When the classic ACO applied in MDPSP, at the step t of constructing feasible solution, ants randomly choose the candidate point y_j from all the unselected candidate points with probability

$$p_j^k(t) = \begin{cases} \frac{\tau_j^\alpha \eta_j^\beta}{\sum_{l \in H} \tau_l^\alpha \eta_l^\beta}, & \text{if } j \in H \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

where pheromone trails τ_j is the degree of pheromone deposited by other ants when the candidate point y_j has been selected as a part of their solutions. Heuristic information η_j is obtained by a function associated with the objective function, and the greater its value is, the more likely candidate point y_j is selected into the optimal solution. α and β are two non-negative parameters that determine the relative importance of pheromone trails and heuristic information. H is the set of candidate points that is not contained in the current partial solution. The random mechanism of selecting next candidate point always needs a long time to converge optimal solution. To accelerate the rate of convergence the improved ACO adopts a hybrid selection strategy combined random mechanism and deterministic choice. The next candidate point y' is selected by (5).

$$y' = \begin{cases} \arg \max_{h \in H} \{p_h^k(t)\} & , \text{ if } p \leq p_0 \\ \text{randomly select } s \text{ based on (9),} & \text{ otherwise} \end{cases} \quad (5)$$

where p is a random variable obeyed uniform distribution in $(0,1)$. p_0 is a adaptive variable concerned with the local optimal solution. If $p \leq p_0$, ants will select the candidate point y_j which has the maximal p_h^k for every $h \in H$ at step t . Otherwise ants randomly select y_j with probability p_j^k . That means p_0 is a key parameter to determine the proportion of deterministic select strategy. If p_0 is too big, ants tend to select the candidate point of maximum probability, thus the algorithm has a greater convergence rate and easily gets local optimum. However, if p_0 is too small, the random select strategy plays a major role when ants select next y_j . It increases the diversity of solutions and leads to a longer running time of the algorithm. Therefore the setting of p_0 is an important matter. This paper adopts an adaptive mechanism to set it. The value of p_0 is not static and unchangeable. It is dynamic and varying with the local optimal solution. At begin set a bigger p_0 to quicken convergence rate, then decrease the value of p_0 by $p_0 = p_0 - \Delta p_0$ to slow convergence rate and increase the solution diversity when the local optimum has not changed for n_0 times and guarantee $p_0 \geq 0$. This setting of p_0 makes the proportion of deterministic mechanism vary with local optimum to balance the convergence rate and solution diversity.

Table 1 The Outline of improved ACO for MDPSP

1	Initialize Parameter and pheromone value
2	while (termination condition is not satisfied) do
3	for ($t = 2$ to m)
4	for (the ant $k = 1$ to m_0)
5	while (all x_i are not dominated, for every u) do
6	use the hybrid select strategy to choose the next candidate point
7	endwhile
8	endfor
9	end for
10	record the local optimal solution
11	while (the local optimums has not changed for n_0 times) do
12	if $p_0 \geq 0$, $p_0 = p_0 - \Delta p_0$ else $p_0 = 0$
13	if $\rho \leq 0.9$, $\rho = \delta\rho$ else $\rho = 0.9$
14	endwhile
15	update the pheromone in all candidate points
16	endwhile
17	Output the final optimal solution

3.3 The Adaptive Pheromone Update

In traditional ACO, ants modify the pheromone trails also after every iteration combined evaporation and accumulation. At every candidate point its pheromone is updated by

$$\tau_j(tt + 1) = (1 - \rho)\tau_j(tt) + \sum_{k=1}^M \rho \Delta \tau_j^k(tt, tt + 1), \quad (6)$$

$$\Delta \tau_j^k(tt, tt + 1) = \begin{cases} \frac{Q}{L_{tt}}, & \text{if the candidate point } y_j \text{ is} \\ & \text{in this iterative optimal solution} \\ 0, & \text{otherwise} \end{cases} \quad (7)$$

where $\tau_j(tt)$ is intensity of pheromone at candidate point y_j calculated until tt th iteration, $\Delta \tau_j^k(tt, tt + 1)$ is the increment from tt th iteration to $(tt + 1)$ th iteration. From the formula (6) we can see no matter accumulation by other ants or increment in current iteration pheromones are both evaporated before deposited, and the evaporation degree is different and competing. M is the number of ants that include y_j in its solution after tt th iteration, ρ reflects the evaporation of pheromone trails, called evaporation rate, Q is a constant, and L_{tt} is the number of candidate points in the optimal solution after tt th iteration.

The difference between traditional ACO and improved ACO is the setting of evaporation rate ρ . In traditional ACO, ρ is a constant, while improved ACO takes adaptive mechanism. It varies with the local optimal solution. If ρ is greater, the evaporation degree of the accumulation, priori knowledge learned by other ants, is bigger compared with increment obtained by the optimal solution after current iteration. Otherwise is smaller. Initially set a smaller ρ to guide ants search solution based on the priori information. To avoid trapping in local optimum, increase the value of ρ to improve the importance of new solution. Update ρ by $\rho = \delta\rho$, $\delta > 1$ when the local optimal solution has been unchanged for consecutive n_0 iterations and guarantee $\rho \leq 1$.

IV. Heuristic Information

Heuristic information is crucial for good performance of ACO algorithms. The use of it can guide the artificial ants towards the most promising solutions. [21] The heuristic information η_j in candidate point y_j is obtained by a function that benefits the goal--getting least candidate points to dominate the whole user point set X for every user behavior u . The value of η_j is dynamic, that is, every time the ant selects next candidate point, η_j is different relied on its partial solution. In this paper we tried different ideas as heuristic information.

- 1) H1-IMACO: set S_k is the ant k 's partial solution at the step t and $X_{S_k}^u$ is a subset of X that S_k can dominate for the user behavior u . Y_{uj} is a subset of X that the candidate point y_j can dominate for the user behavior u . $\Delta X_{(S_k,j)}^u$ means the increment of user points the solution $S_k \cup \{y_j\}$ can dominate compared with S_k for user behavior u and $|\cdot|$ is the number of elements in a set. In this case we use the proportion of total increment in all user points as heuristic information given by formula (8).

$$\eta_j^{(1)} = \frac{\sum_{u \in U} |\Delta X_{(S_k,j)}^u|}{n} \tag{8}$$

- 2) H2-IMACO: firstly, we merge the u dominated matrixes into one dominated matrix DD by using OR operation. Set Y_j^{DD} is the subset of X that y_j can dominate by new dominated matrix DD . And $\Delta X_{(S_k,j)}^{(DD,u)}$ is the increment of user points between $X_{S_k}^u \cup Y_j^{DD}$ and $X_{S_k}^u$. In this case the heuristic information is

$$\eta_j^{(2)} = \frac{\sum_{u \in U} |\Delta X_{(S_k,j)}^{(DD,u)}|}{n} \tag{9}$$

$\eta_j^{(2)}$ is proposed to improve the deficiency of H1-IMACO. The candidate point y_j will be selected with a high probability in H1-IMACO if it has a bigger summation of u different increment corresponding to u kinds of different user behavior. It is reasonable. If we want least candidate points to content every user behavior, we should select ones that can dominate most user points. However, the performance of heuristic information obtained by only accounting the number of increment is not very ideal. It can be illustrated in Example.

Example: set $n = 7, m = 3, l = 2$, $X = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7\}, Y = \{y_1, y_2, y_3\}$ and $D1 = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$,

$$D2 = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, DD = D1|D2 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \end{bmatrix}.$$

Utilize H1-IMACO to solve the problem, we obtain the optimal solution $\{y_3, y_2, y_1\}$ or $\{y_3, y_1, y_2\}$. Utilizing H2-IMACO we get $\{y_1, y_2\}$ or $\{y_2, y_1\}$. The sort of y_j is the selection order in the algorithm. Obviously, the optimal solution is $\{y_1, y_2\}$. The H2-IMACO has better performance by giving more chance to y_1 and y_2 .

- 3) H3-IMACO: sufficiently use the optimal solution for one dominated matrix to combined ideas of H1-IMACO and H2-IMACO. For a given user behavior $u \in U$, we select least candidate points to dominate the user point set X only using the dominated matrix Du , which can be described as a SCP, denoted this problem as Du -SCP, $\forall u \in U$. Then adopt improved ACO to respectively solve these SCP problems. And let the increment of dominated user points when including a new candidate point into current partial solution, which is similar to the idea of H1-IMACO. Considering the randomness of ACO, for every Du -SCP calculate for N_0 times and obtain N_0 optimal solutions. Set N_j^u is the times that candidate point y_j appears in the N_0 optimal solutions. In H3-IMACO, we consider both $|\Delta X_{(S_k,j)}^{(DD,u)}|$ and N_j^u when setting heuristic information, which is given in (10).

$$\eta_j^{(3)} = \frac{\sum_{u \in U} |\Delta X_{(S_k,j)}^{(DD,u)}|}{n} + \frac{\sum_{u \in U} N_j^u}{n} \tag{10}$$

$\eta_j^{(3)}$ is proposed to improve H2-IMACO. When many candidate points have equivalent column vectors in DD , H3-IMACO prefers to select one that is most frequent in the $N_0 \times l$ optimal solution.

V. Experimental Study

According to the detailed description of the improved ACO applied in MDPSP in this section the article will evaluate algorithmic performance through an example. The example is given by MATLAB. 500 user points and 50 candidate points are randomly generated in a plane coordinate. Suppose two user behaviors are considered. Also randomly generate their corresponding 0-1 dominated matrixes and guarantee the amount of 1 is no more than 10% of the total scale, that is, less than or equal to 2500, which means that each candidate test point could control about 50 user points on the average.

5.1 Parameter settings

Many parameters are involved in the improved ACO. And the setting of parameters determines the behavior of ants searching for a solution and it is critical for fast convergence to near optimal solutions. [16] The improved ACO is parameterized by $\alpha, \beta, \rho, Q, p_0, \Delta p_0, \delta, m_0$. Firstly, analyze the influence of some parameters in brief. α and β are respectively determine the weight of pheromone trails and heuristic factor in the selection rule. If α is too greater than β , the ant most utilizes the priori knowledge deposited by other ants to choose the next candidate point. That increases the diversity of the algorithm but needs a long running time to find an optimal solution as the ant doesn't have a sufficient guide. Contrarily, with too higher β and too small α , the algorithm has a fast convergence rate while easy gets into local optimums because ants are too greedy when constructing a solution. ρ and Q influence the value of updated pheromone trails. ρ reflects the evaporation rate of pheromone trails before deposited. Q means the importance of current suboptimal solutions when updating pheromones. Through increasing ρ and decreasing Q the influence of pheromone will be weakened. In this way, ants are more likely to explore a solution by itself so that they may find good solutions. As a counterpart, it takes ants a longer time to achieve optimal solutions. p_0 determines the proportion of deterministic selection in the hybrid selection strategy. Δp_0 and δ are respectively applied in the update of p_0 and ρ . Their influence in improved ACO is analyzed in detail in Section III. m_0 , the number of ants, determines the number of feasible solutions obtained in each iteration. With a too small value, the diversity of solution cannot be guaranteed leading to poor solution quality. However, a great value can improve the global optimality and the stability of the algorithm. When the value is too great, the improved ACO, whose complexity is $T(m) = O(m \cdot m_0 \cdot NC_{max})$, runs a long time without improvement in solution quality as the quality of local optimal solutions cannot be markedly enhanced.

So far, there exists no analytic method to direct the setting of these parameters. Generally, the values of these parameters are obtained in an experimental way that is also used in this paper. [11] We implemented the improved ACO in MATLAB R2015a and ran it under Windows 7 system. For each parameter, we set a set of values and $\alpha \in \{0, 1, 2, \dots, 10\}$, $\beta \in \{0, 1, 2, \dots, 10\}$, $\rho \in \{0.1, 0.2, \dots, 0.9\}$, $Q \in \{10, 20, \dots, 100\}$, $p_0 \in \{0, 0.1, 0.2, \dots, 1\}$, $\Delta p_0 \in \{0, 0.1, 0.2, \dots, 0.6\}$, $\delta \in \{1, 1.05, 1.10, \dots, 1.30\}$, $m_0 = \{10, 20, 30, 40, 50\}$. These parameters are divided into four groups: (α, β) , (ρ, Q) , $(\Delta p_0, \delta)$ and (p_0, m_0) . We change values of one group, while keeping others fixed. For example, we have eleven discrete values for parameters α and β respectively. This yielded 121 different parameter combinations and we ran the improved ACO five times for each of these combinations because of the algorithm's randomness. Then based on the numbers of minimum, mean and mean elapsed time of the five optimal solutions determine the optimal parameter combination. Finally, we found the optimal parameters of the improved ACO are given in Table 2. As an example, three surfaces about evolutions of K_{min} , K_{mean} and $Elapsedtime$ with respect to different (α, β) in H1-IMACO are given in Fig.2.

Table 2 The Optimal Parameters Combination of improved ACO

	α	β	ρ	Q	p_0	Δp_0	δ	m_0
H1-IMACO	5	3	0.7	50	0.7	0.1	1.25	20
H2-IMACO	8	1	0.8	40	0.5	0.5	1.25	20
H3-IMACO	2	10	0.1	40	0.7	0.3	1.25	20

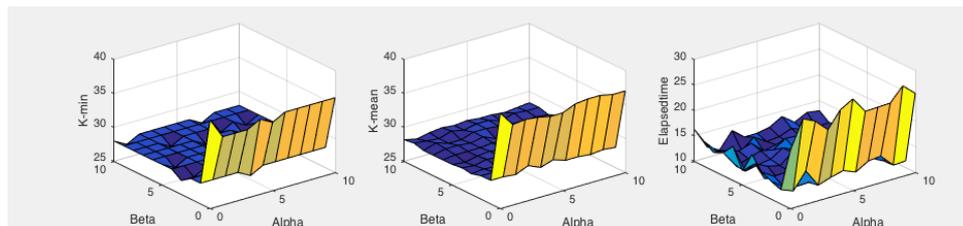


Fig.2 Evolutions of the optimal solutions' minimum and mean and corresponding mean elapsed time when (α, β) changes in H1-IMACO. Other parameters were set to $\rho = 0.7, Q = 100, p_0 = 0.9, \Delta p_0 = 0.1, \delta = 1.15, m_0 = 20$. And $NC_{max} = 50, n_0 = 5$.

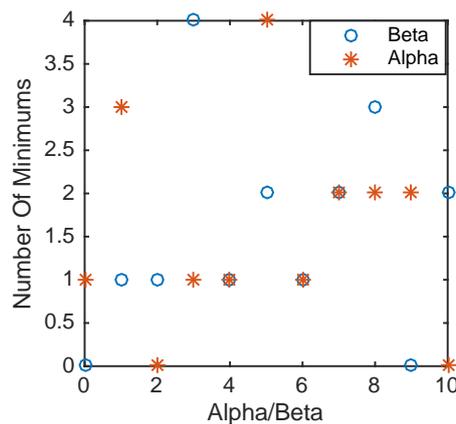


Fig.3 Number of Minimums in K_{min} when α and β vary respectively.

In Fig.2, it can be observed that, when $\beta = 0$, the quality of solution does not satisfy as ants need more time to explore a optimal solution without heuristic information. From the surface of K_{min} the minimums only appear in part points of (α, β) . In Fig.3 we calculate the number of minimums varying with α and β respectively. Select sets of α and β when the number is more than 3, 25% of total. We get $\alpha \in \{1,5\}$ and $\beta \in \{3,8\}$. Then exist four combinations of values of (α, β) . When $(\alpha, \beta) = (5,3)$, K_{mean} is smaller than others and the *Elapsedtime* is least. Therefore, we obtained the optimal combination $\alpha = 5, \beta = 3$.

5.2 Comparison with Some Related Algorithm

In this part, the paper will respectively apply H1-IMACO, H2-IMACO, H3-IMACO, the traditional ACO and the greedy algorithm to the MDPSP. Use the above instance to test their performance. In H3-IMACO, we set $N_0 = 20\%m$ by an experimental way. In the traditional ACO we use the first heuristic information and the same way used in improved ACO to find its optimal parameters combination. For greedy algorithm the idea of most increments is also adopted in the experiment. We run the related ACO algorithms fifteen times and record the minimum and mean of optimums, the first iteration number that the algorithm converges optimal solution, denoted as *FirstIT* and the mean elapsed time, denoted as *ET*. What’s more, we obtained further results about MDPSP through slacking the constraints that the selected candidate points only need to dominate a certain proportion of user pointset X not the whole X . In fact, it is more practical when the scale of the problem is too major and material and financial resources are limit. The results of all algorithms listed in Table 3 are minimal and average values of solutions obtained in 15 independent runs.

Table 3 The results for all algorithms

	100% of X		95% of X		90% of X	
	<i>K_Min</i>	<i>K_Mean</i>	<i>K_Min</i>	<i>K_Mean</i>	<i>K_Min</i>	<i>K_Mean</i>
GA	30	30	19	19	15	15
ACO	28	30.1	19	20.3	15	16.2
H1-IMACO	27	28.2	17	17.8	14	14.7
H2-IMACO	27	28	17	17.8	14	14.8
H3-IMACO	27	27.7	17	17.1	14	14

It can be seen that performance of improved ACO is the best, traditional ACO comes second. Compared with traditional ACO the quality of solution obtained by improved ACO has improved 3.5% when the whole X is dominated. Changing constrains though dominating 95% and 90% user points of X the optimal solution of ACO has been enhanced 10.5% and 6.7% respectively. When compared with greedy algorithm quality of solution has been improved by improved ACO all about 10% for the three proportion of X . Besides, when we solve MDPSP with only one user behavior for each user, they can be described as SCP and are respectively denoted as D1-SCP and D2-SCP. The optimal solution 22 and 23 has been obtained by the improved ACO, that is, average 45% of total candidate points are needed to dominate all X for one user behavior. That means when the constrain is 100% of X , the optimal number of candidate points for all user behaviors, obtained by improved ACO, is just average 9% larger than it for only one user behavior. For the improved ACO algorithms with three different heuristic functions, H3-IMACO has performed most stably as its mean of 15 solutions is least. And H2-IMACO is better than H1-IMACO. In respects of the convergence speed and the computation time between traditional ACO and improved ACO the results are listed in Table 4 that contains the first iteration when the algorithm converged to optimal solution and the total elapsed time. And each value listed in Table 4 is the average over 15 runs.

Table4 Convergence speed and computation time between traditional ACO and improved ACO

	100% of X		95% of X		90% of X	
	FirstIT	ET(s)	FirstIT	ET(s)	FirstIT	ET(s)
ACO	34.1	35.1	9.4	12.8	10.8	12.3
H1-IMACO	2	13.9	1.7	9.74	3	8.5
H2-IMACO	6.8	15.59	2.3	9.49	3.5	8.19
H3-IMACO	3.4	14.58	2.8	10.05	6.0	9.59

Two conclusions can be obtained from Table 4. Firstly, an apparent contrast has appeared between traditional ACO and improved ACO. Compared with the traditional ACO, the average convergence speed of improved ACOs has increased 88%, among them H1-IMACO reaches 94% and the average elapsed time is significantly reduced by 57.6% when covered all the users. Secondly, the differences among the three IMACO algorithms are light. H2-IMACO and H3-IMACO converge to optimal solution a little slowly compared with H1-IMACO, because they consider more candidate points when the ant selects next point.

VI. Conclusion

This paper utilizes the improved ant colony optimization to solve the multi-dominated QoE evaluation point selection problem, which introduces multiple dominant matrices on the basis of the classical SCP when considering a variety of user behaviors. And this feature is embodied in heuristic factor of the improved ACO. Three different heuristic functions are represented to solve it and when used in improved ACO the algorithms all get good result. Ant Colony Optimization, ACO, has been first successfully used in solving the famous traveling salesman problem (TSP). It adopts the distributed computing system, is easy to combine with other methods and has strong robustness, but longer searching time and easily falling into local optimal solution are its prominent disadvantages. To deal with them, a hybrid selection strategy and adaptive updating mechanisms have been designed in the improved ACO, which improves the performance of the traditional ACO and achieves good results. QoE evaluation, associated with the feedback of users' real experience, can more accurately reflect the quality of network service. Allowing for interests, it is vital for operators to know how to accurately find out least candidate test points to test out users' real experience for every kind of user behavior. This paper tries to solve this problem. In the future the following three perspectives can be considered to do more deep reach in solving MDPSP: a) considering a weight for every kind of user behavior or every candidate point that sometimes is the importance operators require, b) considering the varied weight under different scenarios and robust optimization to solve it.

Acknowledgements

This work is supported by the National 863 Plan project under Grant No.2011AA01A102, the National Natural Science Foundation of China (11571015, 11331012), the "Strategic Priority Research Program" of the Chinese Academy of Sciences (XDA06010302), and Huawei Technology Co. Ltd.

References

- [1] Q Dai, *A Survey of Quality of Experience* (Berlin: Springer Berlin Heidelberg, 2011, 146-156).
- [2] M Varela, L Skorin-Kapov, T Ebrahimi, *Quality of Service Versus Quality of Experience* (Springer International Publishing, 2014, 85-96).
- [3] MR Garey, DS Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W.H. Freeman and Company, 1979.
- [4] M Dorigo, GD Caro, The ant colony optimization meta-heuristic, *New Ideas in Optimization*, 28(3), 1999, 11-32.
- [5] M Dorigo, M Birattari, T Stützle. Ant Colony Optimization: Artificial Ants as a Computational Intelligence Technique, *IEEE COMPUT. INTELL. MAG*, 2010, 28-39.
- [6] M Dorigo, C Blum. Ant colony optimization theory: A survey, *Theoretical Computer Science*, 344(2-3), 2005, 243-278.
- [7] M Dorigo, T Stützle, *The Ant Colony Optimization Metaheuristic: Algorithms, Applications, and Advances* (US: Springer US, 2001, 250-285).
- [8] L Lessing, I Dumitrescu, T Stützle, A Comparison Between ACO Algorithms for the Set Covering Problem, *Proc. 4th International Workshop Ant Colony Optimization and Swarm Intelligence*, ANTS 2004, Brussels, Belgium, 2004, 1-12.
- [9] D Alexandrov, D Alexandrov, Y Kochetov, et al. Behavior of the Ant Colony Algorithm for the Set Covering Problem, *Symposium on Operations Research*, 1999, 250-1.
- [10] B Crawford, C Castro, E Monfroy, A New ACO Transition Rule for Set Partitioning and Covering Problems, *Proceedings of the 2009 International Conference of Soft Computing and Pattern Recognition*, IEEE Computer Society, 2009, 426-429.
- [11] Ren Z, Feng Z, Ke L, et al. A fast and efficient ant colony optimization approach for the set covering problem, *IEEE on Evolutionary Computation*, 2008, 1839-1844.
- [12] Crawford, Broderick, et al. *The Impact of a New Formulation When Solving the Set Covering Problem Using the ACO Metaheuristic* (Springer International Publishing, 2015, 209-218).
- [13] Shunmugapriya, P., et al. *Investigation on the Effects of ACO Parameters for Feature Selection and Classification* (Berlin: Springer Berlin Heidelberg, 2012, 136-145).
- [14] P. Pellegrini, T. Stützle, M. Birattari, A critical analysis of parameter adaptation in ant colony optimization, *Swarm Intelligence, Suppl1(4)*, 2009, S16-21.

- [15] Hao, Zhi Feng, R. C. Cai, and H. Huang, An Adaptive Parameter Control Strategy for ACO, *International Conference on Machine Learning and Cybernetics IEEE*, 2006, 203-206.
- [16] D. Gaertner, K. L. Clark, On Optimal Parameters for Ant Colony Optimization Algorithms, *Proceedings of the International Conference on Artificial Intelligence 2005*, 2005, 83-89.
- [17] Wong, Kuan Yew, and Komarudin, Parameter tuning for ant colony optimization: A review, *International Conference on Computer and Communication Engineering*, 2008, 542-545.
- [18] Xu Zhao, *Based on ant colony optimization research on wireless sensor network coverage problem*, University of electronic science and technology, Cheng Du, MA, 2007.
- [19] Yang Gao, Solving set covering problem and its application based on ant colony algorithm, Jiangnan university, Wu Xi, MA, 2007.
- [20] Yu Fu. The improvement and application of ant colony optimization algorithm, Shanghai maritime university, Shang Hai, MA, 2006.
- [21] Abd-Alsabour, Nadia, H. Hefny, and A. Moneim, Heuristic information for ant colony optimization for the feature selection problem, *IEEE Conference Anthology*, 2013, 1-5.