Optimizing QoS for Multimedia Services in Next Generation Network Based on ACO Algorithm

Dac-Nhuong Le
Faculty of Information Technology, Haiphong University, Haiphong, Vietnam
E-mail: Nhuongld@hus.edu.vn

Abstract— In Next Generation Network (NGN), the backbone of the overall network architecture will be IP network, supporting different access network technologies and types of traffics. NGN will provide advanced services, such as Quality of Service (QoS) guarantees, to users and their applications. Factors affecting the QoS in NGN are speech encoders, delay, jitter, packet loss and echo. The negotiation and dynamic adaptation of QoS is currently considered to be one of the key features of the NGN concept. In this paper, I propose a novel Ant Colony Optimization algorithm to solve model of the optimal QoS for multimedia services in the NGN. Simulation results show that my new approach has achieved near optimal solutions. Comparison of experimental results with a recently researches shows that the proposed algorithm is better performance and it can meets the demand of the optimal QoS for multimedia services in NGN.

Index Terms— Quality of Service; Multimedia Services; QoS Matching; Next Generation Networks; Ant Colony Optimization

I. Introduction

In Next Generation Network (NGN), the backbone of the overall network architecture will be IP network, supporting different access network technologies such as wireless Local Area Network (WLAN), UMTS Terrestrial Radio Access Network (UTRAN), and WiMax. Moreover, this integrated wireless system, will have to handle diverse types of traffics: data traffics (e.g. web browsing, e-mail, ftp), voice traffic (e.g. voIP), and multimedia traffics (e.g. video conferencing, online TV, online games), etc... NGN will provide advanced services, such as Quality of Service (QoS) guarantees, to users and their applications. Factors affecting the QoS in NGN are speech encoders, delay, jitter, packet loss and echo. In standardized NGN architecture, at the service level specify requirements for entities at the rim of the network to negotiate and select common codecs for each End to End (E2E) session[1]. The 3rd Generation Partnership Project (3GPP) has specified the IP Multimedia Subsystem (IMS) procedures for negotiating multimedia session characteristics between endpoints[2]. Under these demanding conditions of

media-rich services, network service providers must carefully provision and allocate network resource capabilities to providing personalized and optimal service quality implies more advanced negotiation mechanisms matching user preferences/capabilities against service requirements, network resource capabilities, and operator policy.

Related work on QoS adaptation in IMS networks includes solutions for dynamically adapting multimedia content to fit network/system resource availability. Ozcelebi et al have presented a solution to enhancing E2E QoS for multimedia streaming in IMS[3]. Boula et al introduced an enhanced IMS architecture featuring cross-layer monitoring and adaptation mechanisms to maximizing perceived QoS[4]. The DAIDALOS[5] and ENTHRONE[6] projects is further research on multimedia session control and content adaptation. In additional related work, we can be found on profile matching[7], decision-making for the optimization of service parameters[8-9], and the description of various service and transport configurations for service negotiation[10-12], limited solutions unite these aspects to support delivery of personalized multimedia services in the NGN. Khan has proposed an heuristic algorithm for finding near optimal solutions for cases with a large number of variables to solve the quality adaptation in a multisession multimedia system problem[13]. A different approach to formulating the optimization problem may be to use tools specified as part of the MPEG DIA standard, Usage Environment Description tools, Universal Constraints Description tool, and Adaptation QoS tools has presented by Mukherjee et al[14]. In [15], the authors have proposed the high-level concept of a QoS Matching and Optimization Function (Q-MOF) to be included along the E2E signaling path in the service control layer of the NGN architecture and heuristic algorithm for finding near optimal solutions.

In this paper, I propose a novel Ant Colony Optimization (ACO) algorithm to solve the Optimal QoS for Multimedia Services (OQMS) problem. Simulation results show that my new approach has achieved near optimal solutions. I have compared to evaluate the effectiveness of the proposed algorithm with the recently researches. The rest of this paper is organized as follows. Section 2 and section 3 present Q-MOF model in the NGN architecture[15] and OQMS problem formulation. Section 4 presents our new

algorithm for OQMS problem based on ACO algorithm. Section 5 presents our simulation and analysis results, and finally, section 6 concludes the paper.

II. Q-MOF in the NGN Architecture

Fig.1 presents the ITU-T NGN release an architecture based on the concept of independence between the transport stratum and service stratum [1]. In the service stratum, service control functions are based on an IMS service environment and support the provisioning of real-time Session Initiation Protocol (SIP) based multimedia services. The application support functions and service support functions can impact sessions on behalf of services. During session negotiation, QoS requirements are extracted by service control functions and used to issue resource reservation and authorization requests to the *Resource Admission Control Functional* architecture (RACF). In order to provide enhanced QoS control mechanisms, Q-MOF have introduced in the service stratum as a new application server. The Q-MOF is designed to be included along the E2E signaling path, providing support for advanced QoS parameter matching and optimization logic within the

QoS negotiation process. Actual service adaptation may be performed at communication endpoints or in the network. While today's Internet model is based on a user-centric view of the network, the IMS model uses a more operator-centric approach[15].

In an actual network implementation, the authors in [15] proposed the Q-MOF to be included in a service provider domain as a generic reusable functionality to providing a better service to users, introducing enhanced QoS support in the network as a reusable service capability would benefit both the service provider and third-party service/application providers. The service provider would have additional means to control, differentiate, and appropriately charge the QoS a particular user receives for a given multimedia service. Third-party service/application providers would have to specify a service profile stating service requirements and options and would further be relieved from implementing complex QoS decision making functionality for each new introduced service, hence leading to simplified provisioning and possibly quicker time-to-market for new services requiring such mechanisms. A multimedia service is composed of two or more media components such as audio flows, video flows, graphics data, *etc*…[15]
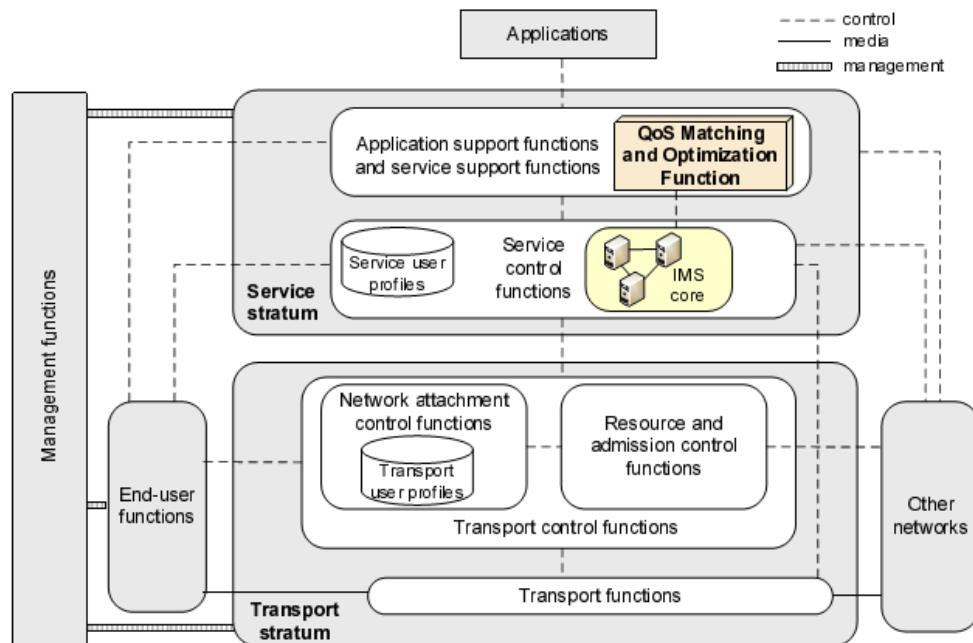

Fig. 1: Q-MOF in the NGN architecture

I assume that a multimedia service exists in one or more versions to meet heterogeneous user and network capabilities. We specify service versions as differing in the included media components. Each media component may be configured by choosing from offered alternative operating parameters (e.g., *different codecs, frame rates, resolutions, etc.*). We refer to the overall service configuration as the set of chosen operating parameters for all included media components [13, 15].

III. Model of OQMS Problem

The optimal QoS for multimedia services (OQMS) problem can be formulated as a multi-choice multi-dimension 0-1 knapsack problem (MMKP), for which I introduce the following notation:

- $F = (f_1,...,f_n)$ is set of different media flows (n is number of different media flows), of which flows

$f_1, ..., f_h$ are in the *downlink direction* and flows $f_{h+1}, ..., f_n$ are in the *uplink direction*.

- The flow f_i has p_i operating points.

- $r_{ij} = (r_{ij1}, ..., r_{ijq}, ..., r_{ijm})$ is set of required resources r for operating point p_j and media flow f_i, of which m is number of resources, $r_{ij1}, ..., r_{ijq}$ are corresponding to bandwidth assigned within different QoS classes (*differing in delay, loss, jitter, and bandwidth guarantees*). For a single operating point, we assume only one of the values $r_{ij1}, ..., r_{ijq}$ to be greater than zero, while all others are equal to zero. This is because only one QoS class is chosen per operating point for a media flow.

- $B_{downlink}$ and B_{uplink} are maximum available downlink and uplink bandwidth determined by user terminal capabilities and access network. They represent the resource constraints for $r_{ij1}, ..., r_{ijq}$.

- $R = (R_{q+1}, ..., R_m)$ is the consumption of a particular resource across all media flows (e.g., *cost for all flows must be less than a specified amount*).

- $R_i = (R_{i1}, ..., R_{im})$ is regarding resource consumption per flow f_i (e.g., *cost for a media flow must be less than a specified amount*).

- $u_i(r_{ij})$ is the utility value for operating point p_j and media flow f_i.

- w_i is weight factors assigned to utility values to indicate the relative importance of media flows.

- $x_{ij} = \begin{cases} 1 \text{ if operating point } p_j \text{ is chosen per media flow } f_i \\ 0 \text{ otherwise} \end{cases}$
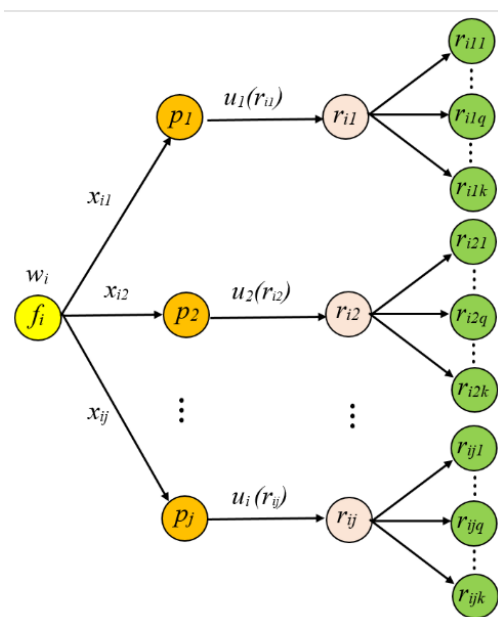

Fig. 2: Flow i^{th} in OQMMS problem

Then the formulation OQMS for multi-flow resource constrained utility maximization is:

$$\max \sum_{i=1}^{n} \sum_{j=1}^{p_i} w_i x_{ij} u_i(r_{ij})$$

Subject to

$$\sum_{i=1}^{h} \sum_{j=1}^{p_i} \sum_{k=1}^{q} x_{ij} r_{ijk} \le B_{downlink}$$

$$\sum_{i=h+1}^{n} \sum_{j=1}^{p_i} \sum_{k=1}^{q} x_{ij} r_{ijk} \le B_{uplink}$$

$$\sum_{i=1}^{n} \sum_{j=1}^{p_i} x_{ij} r_{ijk} \le R_k, \quad \forall k = \overline{q+1, .., m}$$

$$\sum_{j=1}^{p_i} x_{ij} r_{ijk} \le R_{ik}, \quad \forall k = \overline{1..m}, \forall i = \overline{1..n}$$

$$\sum_{i=1}^{n} \sum_{j=1}^{p_i} x_{ij} = 1, \quad x_{ij} \in \{0,1\}, \forall i = \overline{1..n}, \forall j = \overline{1.. p_i}$$

IV. ACO based Algorithm for the OQMS

4.1 Ant Colony Optimization

The ACO algorithm is originated from ant behavior in the food searching. When an ant travels through paths, from nest food location, it drops pheromone. According to the pheromone concentration the other ants choose appropriate path. The paths with the greatest pheromone concentration are the shortest ways to the food. The optimization algorithm can be developed from such ant behavior. The first ACO algorithm was the Ant System[16], and after then, other implementations of the algorithm have been developed[17-18].

There are some ACO based solutions for a more restricted variant of *Knapsack Problem* (KP), namely *Multi-choice Knapsack Problem* (MKP) and *Multi-choice Multi-dimension Knapsack Problem* (MMKP) when MKP resources have multiple dimensions. However there is no concept of group in MKP. As a result, MKP can be thought of as a restricted version of MMKP, which has all objects in a single group[19-22].

The exist algorithms differ in deciding which component of the problem should be regarded as the pheromone depositing component and in the mechanisms of pheromone updating:

- ***Pheromone Trails on Each Object***: The first way is to lay pheromone trails on each object belonging to the current solution set the amount of pheromone represents the preference of the object[19].

- ***Pheromone Trails on Each Pair***: In this case, pheromone trails are laid on each pair (o_i, o_j) of successively selected objects of the solution set the idea is to increase the desirability of choosing object o_j when the last selected object is o_i[20].

- ***Pheromone Trails on All Pair***: The third one is to lay pheromone trails on all pairs of different objects of the solution set. Here, the idea is to increase the desirability of choosing simultaneously two objects of S[21].

- ***Pheromone Diffusion Model***: The forth approach follows the same principle as the first one. Additionally it uses a pheromone diffusion scheme where pheromone trails are laid on objects that tend to occur together in previous solutions[22].
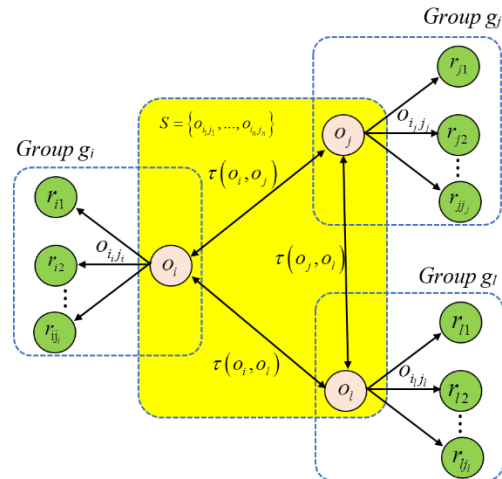
4.2 Solving the OQMS Based on ACO

In this section, I present application of ACO technique with the dynamic local heuristic information for the OQMS problem. My new algorithm is described as follows:

Let $\tau_{\min}$ and $\tau_{\max}$ are lower and upper bounds on pheromone values and pheromone $\tau_{\min} < \tau < \tau_{\max}$. I initialize all pheromone trails to $\tau_{\max}$. The algorithm chooses first group g_i has been randomly chosen. For each candidate group o_j, the pheromone factor $\tau_{S_k}(g_j)$ is initialized to $\tau(o_i, o_j)$. Each time a new group g_l is added to the solution S_k, for each candidate group g_j, the pheromone factor $\tau_{S_k}(g_j)$ is incremented by $\tau(o_l, o_j)$. And the highest pheromone factor can be defined as

$$\tau_{S_k}(g_i) = \sum_{g_j \in S_k} \tau(o_i, o_j)$$

When the algorithm starts, ants can perform search in order of group that impose the lowest restriction on the next choices. The key point is to decide which components of the constructed solutions should be rewarded, and how to exploit these rewards when constructing new solutions.

A solution is a set of selected objects $S = \{o_{ij} \mid x_{o_{ij}} = 1\}$. Say $x_{o_{ij}} = 1$ is the mean an object o_{ij} is selected and the corresponding decision variable x_{ij} has been set to 1. We must constructed solutions $S = \{o_{i_1 j_1}, ..., o_{i_n j_n}\}$, pheromone trails are laid on each objects selected in S. So pheromone trail τ_{ij} will be associated with object o_{ij}. Fig.3 show an example of the solution S in my algorithm.

Fig. 3: Solution S in our algorithm

Let S_k be the set of the selected objects at the kth Iteration. The heuristic factor $s_k(O_{ij})$ is determined by:

$$s_k(O_{ij}) = \frac{w_i u_i(r_{ij})}{\sum_{i=1}^{m} \dfrac{r_{ijl}}{d_{S_k}(l)}}$$

where,

- $d_{S_k}(l) = R_l - \sum_{o_{ij} \in S_k} r_{ijl}$ (10)

- l is index of resource R across all media flows.

- S_k is the dynamic heuristic information will be changed from step to step.

My algorithm select groups in the same way as it selects the objects. I have unique idea of a separate pheromone trail for groups to save the ordering of groups that lead to a good solution. The group pheromone trail also follow a min-max ant system approach and initialized to the max pheromone value[24].

Once each ant has constructed a solution, pheromone trails laying on the solution objects are updated according to the ACO meta-heuristic. First, all amounts are decreased in order to simulate evaporation. This is done by multiplying the quantity of pheromone laying on each object by a pheromone persistence rate $(1-P)$ such that $0 \le P \le 1$. Then, pheromone is increased for all the objects in the best solution of the iteration. When constructing a solution, an ant starts with an empty knapsack. At the first construction step an ant selects a group randomly and at all the latter steps, groups are selected according to their associated pheromone value.

After selecting a group, the algorithm removes all the bad *Candidates* that violates resource constraints. It then updates the local heuristic information of the remaining candidate objects of the group and selects an object according to the following probability equation:

$$P_{S_k(O_{ij})} = \frac{\left[\tau_{S_k}\left(O_{ij}\right)\right]^{\alpha}\left[\eta_{S_k}\left(O_{ij}\right)\right]^{\beta}}{\sum\limits_{O_{ij} \in Candidates}\left[\tau_{S_k}\left(O_{ij}\right)\right]^{\alpha}\left[\eta_{S_k}\left(O_{ij}\right)\right]^{\beta}}$$

in which,

- *Candidates* are all items from the currently selected group which do not violate any resource constraints.

- τ_{S_k} is the pheromone factor of the dynamic heuristic information S_k and computed by equation (8)

- η_{S_k} is the desirability of S_k.

- The influence of the pheromone concentration to the probability value is presented by the constant α, while constant β do the same for the desirability. Parameters α and β control the relative importance of pheromone trail versus local heuristic value.

More precisely, let $S_{internalbest}$ be the best solution constructed during the current cycle. The quantity of pheromone increased for each object is defined by:

$$G\left(S_{\text{internalbest}}\right) = \frac{1}{\sum\limits_{k=1}^{n} P_{S_k(O_{ij})}} \times \sum\limits_{o_{ij} \in S_{\text{internalbest}}} w_i u_i\left(r_{ij}\right)$$

After each iteration, group pheromone values have much better than the original pheromone trail for ants. The best solution updates the group pheromone trail. All the adjacent groups get the highest amount of pheromone value that gradually diminishes as the distance between groups increases.

The *Candidategroups* data structure maintains a list of feasible candidate groups which can be considered next. After each ant has constructed a solution, the best solution of that iteration is identified and a random local search procedure and a random item swap procedure is applied to improve it. Then pheromone trail is updated according to the best solution. Also it maintains a database of top k solutions. After each iteration a small amount of pheromone is deposited in the pheromone trails of the objects belonging to the top k solutions. The motivation behind this strategy is to ensure quick convergence on good solutions and to explore better areas more thoroughly. The algorithm stops either when an ant has found an optimal solution (when the optimal bound is known), or when a maximum number of cycles has been performed.

Our algorithm based on ACO for OQMS described as follows:

ACO_ OQMS Algorithm
1.   Initialize pheromone trails to τ_{max} for both item and group
2. $Topksolution \Leftarrow \emptyset$ {hold topmost k solutions}
3.   **repeat**
4. $S_{globalbest} \Leftarrow \emptyset$
5.   **for** each ant $k = 1...N_{Max}$ **do**
6. $S_{internalbest} \Leftarrow \emptyset$
7. $Candidategroups \Leftarrow$ all the groups
8.   **while** $Candidategroups \neq \emptyset$ **do**
9. $C_g \Leftarrow$ Select a group from $Candidategroups$ according to group pheromone trail.
10. $Candidates \Leftarrow \{o_{ij} \in$ objects in C_g that do not violate resource constraints in (2-5)}
11.   Update local heuristic values by equation (9)
12.   Choose an object $o_{ij} \in Candidates$ with   probability computed by equation (12)
13. $S_k \Leftarrow \{S_k \cup o_{ij}\}$
14.   Remove C_g from $Candidategroups$
15.   **end while**
16.   **if** $profit(S_k) > profit(S_{internalbest})$ **then**
17. $S_{internalbest} \Leftarrow S_k$
18.   **end if**
19.   **end for**
20.   **if** $profit(S_{globalbest}) < profit(S_{internalbest})$ **then**
21. $S_{globalbest} \Leftarrow S_{internalbest}$
22.   **end if**
23.   Update database and pheromone trails and lower and upper bounds on pheromone τ_{min} and τ_{max}.
24. **Until** maximum number of cycles reached or optimal solution found

V. Experiments and Results

In the first experiment, I implemented the similar prototype Audio/Video Call (AVC) service allow two end users to engage in an audio/video call in a laboratory IMS testbed[15,25]. The AVC service matching parameter and service requirement with conversational audio and video is shown in table 1. We assume the uplink direction from the call initiating user A to the terminating user B. The user capabilities and operator constraints is shown in table 2.

For illustration, I assume a specification of constraints follow:

- Set of different media flows:

$$F = \left(f_1, f_2, f_3, f_4\right).$$

- Flows f_1, f_2 are audio downlink and video downlink and flows f_3, f_4 are audio uplink and video uplink.

- Operating points:

$$P = \left(p_1, p_2, p_3, p_4\right) = (3,8,3,8)$$

- Utility value $u_i(r_{ij})$ in the range of [0,1]

- $B_{downlink} = 1200$ kbps

- $B_{uplink} = 800$ kbps

- A hypothetical price: 10 [monetary unit/bit].

- Maximum cost 20000 [monetary unit/s]

- Cost as bandwidth class q [bit/s] x price class q [monetary unit/b].

Our operating parameters, resource vectors and corresponding utilities are specified in the service profile and summarized in Table 3.

The parameters of our algorithm are set as follows:

- $N_{MAX} = 50$, $k = 10$, $\alpha = 1$, $\beta = 5$

- $P = 0.01$, $\tau_{min} = 0.01$, $\tau_{max} = 8$

- Pheromone deposited = 0.1

- Diffusion rate = 0.5

- Maximum number of cycles: 500

The results of objective value and resource configuration of our algorithm updated step by step show in table 4. The experimental results with the similar input data sets in [15] show that resource configurations of our algorithm is similar results solved by GLPK (*GNU Linear Programming Kit*). So, this is a promising approach to solve this problem.

In the second experiment, we generated random input data to verify the effectiveness of our algorithm follow constraints below. The experiment was conducted on Genuine Intel® CPU DuoCore 3.0 GHz, 2 GB of RAM machine. I ran experiment our ACO algorithm. The problems tackled, object value, and time processing shown in table 5, fig.4 and fig.5.

VI. Conclusions

In this paper, I propose a novel Ant Colony Optimization algorithm to solve OQMS problem. Simulation results show that our new approach has achieved near optimal solutions. The effectiveness and the efficiency of our algorithm by comparing with the recent researches and it can meets the demand of QoS matching for multimedia services in NGN. Optimizing QoS negotiation models in cases involving multiple providers in service delivery is our next research goal.

Table 1: Matching parameter set for AVC service [15]

AVC service matching parameter set		Service requirements
p_1	Media Component	audio, video
p_2	Codecs	audio: mpeg, gsm. video: mpeg, h26
p_3	Min Bandwidth Downlink	46
p_4	Max Delay Downlink	150
p_5	Max Jitter Downlink	10
p_6	Max Loss Downlink	1
p_7	Min Bandwidth Uplink	46
p_8	Max Delay Uplink	150
p_9	Max Jitter Uplink	10
p_{10}	Max Loss Uplink	1
p_{11}	Resolution Local	176x144
p_{12}	Resolution Remote	176x144

Table 2: Matching parameter set for AVC service [15]

Parameters	User A capabilities	User B capabilities	Operator constraints
p_1	audio, video, data, image, model	audio, video, data, image, model	User A and B: audio, video, data, image, text
p_2	audio: peg, pcm, gsm; video: mpeg, mjpeg, h263	audio: peg, pcm, gsm; video: mpeg, mjpeg, h263	Not Allowed : audio, G729, audio G723
p_3	1200	1300	1400
p_4	150	150	N/A
p_5	N/A	N/A	N/A
p_6	N/A	N/A	N/A
p_7	800	1000	1400
p_8	150	150	N/A
p_9	N/A	N/A	N/A
p_{10}	N/A	N/A	N/A
p_{11}	1204x768	N/A	N/A
p_{12}	N/A	1204x768	N/A

Table 3: Operating parameters, resource vectors and utilities for AVC flows [15]

Flow	Operating parameter	Resource vectors	Bandwidth [kpbs] r_{ij1}	Cost r_{ij1}	Utility value $u_i(r_{ij})$
f_1	p_1	r_{11}	21	210	0.5
	p_2	r_{12}	34	340	0.8
	p_3	r_{13}	64	640	1.0
f_2	p_1	r_{21}	25	900	0.2
	p_2	r_{22}	90	900	0.4
	p_3	r_{23}	370	3700	0.5
	p_4	r_{24}	400	4000	0.6
	p_5	r_{25}	781	7810	0.7
	p_6	r_{26}	1015	10150	0.8
	p_7	r_{27}	1400	14000	0.9
	p_8	r_{28}	2000	20000	1.0
$f3$	p_1	r_{31}	21	210	0.5
	p_2	r_{32}	34	340	0.8
	p_3	r_{33}	64	640	1.0
f_4	p_1	r_{41}	25	900	0.2
	p_2	r_{42}	90	900	0.4
	p_3	r_{43}	370	3700	0.5
	p_4	r_{44}	400	4000	0.6
	p_5	r_{45}	781	7810	0.7
	p_6	r_{46}	1015	10150	0.8
	p_7	r_{47}	1400	14000	0.9
	p_8	r_{48}	2000	20000	1.0

Table 4: Resource configurations for the ACO_ OQMS Algorithm updated step by step

Number of cycles	Resource Configuration	Objective value
93	$(r_{13}, r_{23}, r_{33}, r_{44})$	2.770
128	$(r_{13}, r_{25}, r_{33}, r_{43})$	2.839
187	$(r_{13}, r_{26}, r_{33}, r_{42})$	2.840
264	$(r_{13}, r_{24}, r_{33}, r_{44})$	2.840
331	$(r_{13}, r_{25}, r_{33}, r_{44})$	2.909
375	$(r_{13}, r_{26}, r_{33}, r_{43})$	2.909
417	$(r_{13}, r_{26}, r_{33}, r_{44})$	**2.980** (Optimal)

Table 5: The problems tackled and results of ACO_OQMS Algorithm

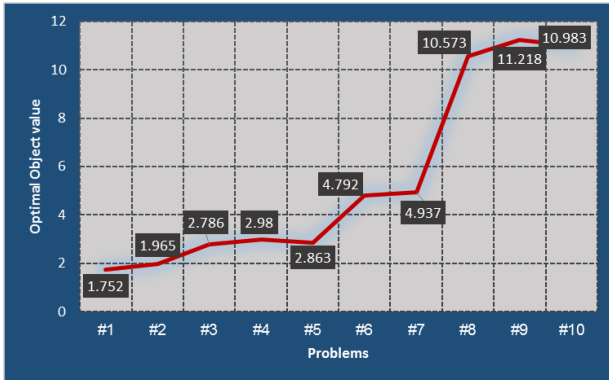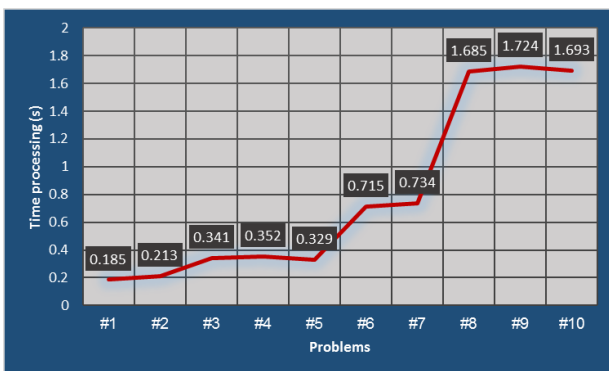| Problems	No. of flows n	Operating parameter	
		Downlink $(p_1,...,p_h)$	Uplink $(p_{h+1},...,p_n)$
#1	2	(2)	(2)
#2	2	(3)	(4)
#3	4	(4)	(2, 5, 7)
#4	4	(3, 8)	(3, 8)
#5	4	(3, 7, 4)	(9)
#6	6	(6, 8, 9)	(8, 5, 4)
#7	6	(10, 15)	(8, 10, 12, 13)
#8	8	(10, 15, 20, 25)	(10, 15, 20, 25)
#9	8	(15, 20, 30)	(20, 10, 30, 25, 15)
#10	8	(20, 25, 30, 35, 45)	(40, 35, 50)

Fig. 4: Problems tackled's optimal object values

Fig. 5: Comparing times processing

References

[1] ITU-T Recommendation Y.2012. *Functional requirements and architecture of the NGN release 1,* 2006

[2] 3GPP TS 23.228*: IP Multimedia Subsystem (IMS); Stage 2, Release 8* 2008

[3] Ozcelebi, T., Radovanovic, R., Chaudron, M.*: Enhancing End-to-End QoS for Multimedia Streaming in IMS-Based Networks*. In: ICSNC, pp. 48–53, 2007.

[4] Boula, L., Koumaras, H., Kourtis, A.*: An Enhanced IMS Architecture Featuring Cross-Layer Monitoring and Adaptation Mechanisms*. ICAS, Spain (2009)

[5] IST DAIDALOS - EU FP6,http://www.ist-daidalos.org/

[6] IST ENTHRONE - EU FP6,http://www.ist-enthrone.org/

[7] Houssos, N., et al.: *Advanced Adaptability and Profile Management Framework for the Support of Flexible Service Provision*. IEEE Wireless Communications, pp.52–61, 2003.

[8] Khan, S.: *Quality Adaptation in a Multisession Multimedia System: Model, Algorithms and Architecture*. PhD Thesis, Univ. of Victoria, 1998

[9] Wang, Y., Kim, J.-G., Chang, S.-F., Kim, H.-M.: *Utility-Based Video Adaptation for Universal Multimedia Access (UMA) and Content-Based Utility Function Prediction for Real-Time Video Transcoding*. IEEE Trans. on Multimedia 9(2), pp.213–220, 2007

[10] Guenkova-Luy, T., Kassler, A.J., Mandato, D.: End-to-End Quality-of-Service Coordination for Mobile Multimedia Applications. IEEE J. Selec. Areas Com-mun. 22(5), pp.889–903, 2004

[11] Rosenberg, J., et al.: SIP: Session Initiation Protocol. IETF RFC 3261 2002.

[12] Handley, H., Jacobson, V.: *SDP: Session Description Protocol*. IETF RFC 2327, 1998.

[13] Khan, S.: *Quality Adaptation in a Multisession Multimedia System: Model, Algorithms and Architecture*. PhD Thesis, Univ. of Victoria, 1998.

[14] Mukherjee, D., Delfosse, E., Kim, J.-G., Wang, Y.: *Optimal Adaptation Decision-Taking for Terminal and Network Quality of Service*. IEEE Trans. on Multime-dia 7(3), pp.454–462, 2005

[15] Lea Skorin-Kapov, and Maja Matijasevic, *Modeling of a QoS Matching and Optimization Function for Multimedia Services in the NGN*, MMNS 2009, LNCS 5842, pp.55–68, 2009.

[16] M. Dorigo, V. Maniezzo, and A. Colorni, *Ant system: Optimization by a colony of cooperating agents,* IEEE Trans. on System, MAN, and Cybernetics-Part B, vol. 26, pp. 29-41, 1996.

[17] E. Rajo-Iglesias, O. Quevedo-Teruel, *Linear Array Synthesis using an Ant Colony Optimization based Algorithm*, IEEE Trans. on Antennas and Propagation, 2005.

[18] M. Dorigo, M. Birattari, and T. Stitzle, *Ant Colony Optimization: Arificial Ants as a Computational Intelligence Technique*, IEEE computational intelligence magazine, November, 2006.

[19] Leguizamon, G., Michalewicz, Z.: *A new version of ant system for subset problems. In: Evolutionary Computation*, 1999. CEC 99. Proceedings of the 1999 Congress on. Volume 2. 1999

[20] Fidanova, S.: *Aco algorithm for mkp using different heuristic information*. Lecture Notes in Computer Science 2542, pp.438–444, 2003

[21] Alaya, I., Solnon, C., Gh`edira, K.: *Ant algorithm for the multi-dimensional knapsack problem*. In: International Conference on Bioinspired Optimization Methods and their Applications (BIOMA 2004) pp.63–72, 2004.

[22] Ji, J., Huang, Z., Liu, C., Liu, X., Zhong, *N.: An Ant Colony Optimization Algorithm for Solving the Multidimensional Knap-sack Problems. In: Proceedings of the 2007 IEEE/WIC/ACM International Conference on Intelligent Agent Technology*, IEEE Computer Society, pp.10–16, 2007.

[23] Shahrear Iqbal, Md. Faizul Bari, Dr. M. Sohel Rahman, *A novel ACO technique for Fast and Near Optimal Solutions for the Multi-dimensional Multi-choice Knapsack Problem*, Proceedings of 13th International Conference on Computer and Information Technology (ICCIT 2010), Dhaka, Bangladesh, pp.33-38, 2010.

[24] Stutzle, T., Hoos, H.H.: *Max–min ant system.* Future Generation Computer Systems 16, pp.889–914, 2000.

[25] Skorin-Kapov, L., Mosmondor, M., Dobrijevic, O., Matijasevic, *M.: Application-level QoS Negotiation and Signaling for Advanced Multimedia Services in the IMS.* IEEE Comm. Magazine 45(7), pp.108–116, 2007.

[26] Dac-Nhuong Le, *Optimizing Resource Allocation to Support QoS Requirements in Next Generation Networks using ACO Algorithm*, International Journal of Computer Science and Information Technology & Security (IJCSITS), Vol.2, No.5, pp.931-938, 2012.

[27] Dac-Nhuong Le, *Optimizing the cMTS to Improve Quality of Service in Next Generation Networks based on ACO Algorithm*, International Journal of Computer Network and Information Security (IJCNIS), Vol.5, No.4, 2013

[28] Dac-Nhuong Le, *PSO and ACO Algorithms Applied to optimal Resource Allocation to Support QoS Requirements in Next Generation Networks*, International Journal of Information & Network Security (IJINS), Vol.2, No.3, pp.216-228, 2013.

[29] Dac-Nhuong Le, *PSO and ACO Algorithms Applied to Optimizing Location of Controllers in Wireless Networks,* International Journal of Computer Science and Telecommunications, Vol.3 No.10, pp.1-7.

[30] Dac-Nhuong Le, Nhu Gia Nguyen, and Vinh Trong Le, *A Novel Ant Colony Optimization-based Algorithm for the Optimal Centralized Wireless Access Network*, Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering (LNICST), Springer 2013.

Author's Profiles

Dac-Nhuong Le received the BSc degree in computer science and the MSc degree in information technology from College of Technology, Vietnam National University, Vietnam, in 2005 and 2009, respectively. He is a lecturer at the Faculty of information technology in Haiphong University, Vietnam. He is currently a Ph.D student at Hanoi University of Science, Vietnam National University. His research interests include algorithm theory, computer network and networks security.