

OPTIMALLY STABLE MATCHINGS FOR RESOURCE ALLOCATIONS

Le Hong Trang^{1,2,*}, Hoang Huu Viet³

¹*Faculty of Computer Science and Engineering, Ho Chi Minh City University of Technology (HCMUT), 268 Ly Thuong Kiet, District 10, Ho Chi Minh City, Viet Nam*

²*Vietnam National University - Ho Chi Minh City, Linh Trung Ward, Thu Duc, Viet Nam*

³*School of Engineering and Technology, Vinh University, 182 Le Duan, Vinh City, Viet Nam*

*Emails: lhtrang@hcmut.edu.vn

Received: 1 June 2021; Accepted for publication: 22 August 2021

Abstract. The stable marriage problem (SMP) and its variants have received much attention in the literature due to their wide range of applications. One of their applications is resource allocation in network environments. In this paper, we consider two resource allocation problems. The first one is to maximize the performance of a system in the fog computing environment while maintaining a low cost. In the problem, the resource allocation task will be rewritten as the MAX-SMTI variant (i.e., finding the maximum cardinality stable matching of the stable marriage with ties and incomplete lists). It is then formulated under an integer linear program to solve. The best allocation can then be chosen to be the lowest cost one among stable matchings. In the same manner, another variant called MAX-HRT (i.e., finding the maximum cardinality stable matching of the hospital-residents with ties) is applied for the second application regarding the virtual machine allocation. By using MAX-SMTI and MAX-HRT models which are solved via integer linear programs, we aim to not only find stable matchings for the resource allocation problems but also maximum length matchings. Consequently, a maximum number of user requests should be served at a time. The models are implemented in C++ using the SCIP solver. Numerical experiments are conducted for large datasets and results are given to show the efficiency of the models.

Keywords: stable marriage, integer linear programming (ILP), resource allocation, network computing.

Classification numbers: 4.5.3, 4.7.1, 4.10.2.

1. INTRODUCTION

The stable marriage problem (SMP), introduced by Gale and Shapley [6] in 1962, is a well-known problem where we are given a set of people and each of them ranks others in strict order of preference, saying the desired levels that the person wants to match up with. A stable matching of the problem is a pairing such that there is no blocking pair in the matching. Two people form a blocking pair if they prefer each other than their current partner in a considering matching. Gale and Shapley in [6] also presented a linear-time algorithm for finding a stable matching. In some situations, it is requested that every person needs to give a preferred level

over all the others, meaning that its preference list can be incomplete. This is the stable marriage problem with incomplete lists (SMI). On the other side, a person can rank equally a set of people, since he/she cannot decide a strict order preference among these people. In this case, the preference lists of the problem can include ties, giving another variant of SMP that is the stable marriage with ties (SMT). When both incomplete lists and ties are allowed in a problem, we have the problem with ties and incomplete lists (SMTI). It is not difficult to find a so-called weakly stable matching of a given instance by breaking all the ties and then applying a modified Gale and Shapley's algorithm. Since there are many ways to break the ties, many weakly stable matchings can be found in different sizes. It is natural to find a weakly stable matching of maximum cardinality (MAX-SMTI). This problem is NP-hard [15]. Some approximate algorithms proposed for the problem can be found in [12, 16]. The local search approach was studied for this problem, as given in [7, 8, 17, 21]. However, these methods will be challenged when considering variants with more constraints required in real-world applications. It is well known that integer linear programs have been well studied for a wide range of combinatorial optimization problems, for example, scheduling [26] and matching problems. For dealing with variants of SM problems, in 1989, Gusfield and Irving [9] and Vate [20] formulated linear programming models for SMP and SMI instances. They indicated that stable matchings are exactly extreme points of polytopes presenting the constraint domains. These models are then extended for the MAX-SMTI and HRT [14]. A comprehensive study of mathematical models for the problems has been recently presented by Delorme *et al.* in [4]. We utilize these models for the resource allocation problems in this paper. They will be reviewed in Section 2.

SMP and its variants have shown their important role in a wide range of applications such as the Evolution of the Labor Market for Medical Interns and Residents [19], the Student-Project Allocation problem [1], course timetabling for maximal enrollments [23], and the Stable Roommates problem [5, 10]. In the last few years, stable matching problems have been studied in the literature for resource allocation applications. In 2016, Wang *et al.* proposed a virtual machine allocation based on stable matching in a cloud data center [22]. They adopted a deferred acceptance procedure to handle conflicts among the preferences of virtual machines and physical hosts. Regarding the scheduling of computing and storing the resource in data centers, Chu *et al.* presented a three-sided stable matching scheme for arranging among computing, storage resources, and applications [3]. They showed stability and the correctness of the proposed algorithm. For the problem of device-to-device (D2D) communication in networks, relays are usually used to reduce the loss in the user data rate. A low complexity resource allocation procedure is required in the relays. Kishk *et al.* in [13] proposed a centralized method using time-sharing and a distributed low complexity allocation using the stable matching theory. To optimize the energy consumption in mobile devices in the D2D communication, Zhou *et al.* considered the joint problem of power control and spectrum resource allocation [25]. They first formulated a joint problem as two-dimensional matching between D2D pairs and cellular user equipment. Preference lists are established among two sides of the matching. They then presented an energy-efficient matching algorithm based on Gale and Shapley's algorithm. Other stable matching-based methods for resource allocation can be seen in [2]. Most of the proposed methods slightly extended Gale and Shapley's algorithm. However, due to the deferred acceptance scheme in Gale and Shapley's algorithm, it could not be ensured the optimality of the obtained matching by means of the length of the matching.

In this paper, we consider two resource allocation problems. The first problem is to maximize the performance of a system in the fog computing environment while maintaining a low cost. Recently, a micro-level cost-based method has been introduced by Batulla *et al.* for this problem [2], which includes two phases. They proposed first a micro-level cost model to

estimate the cost of the user participating in fog environments. A resource allocation algorithm based on stable matching is then applied to benefit both users and providers in the environment. They used a modified Gale and Shapley's algorithm for the stable matching problem. In our work, the problem of resource allocation will be stated as the MAX-SMTI variant of SMP. It is then formulated under an integer linear program (ILP) to solve. The best allocation can then be chosen to be the lowest cost one among stable matchings. In the same manner, another variant called the MAX-HRT model is applied for the second application regarding virtual machine allocation. By using MAX-SMTI and MAX-HRT models which are solved via integer linear programs, we aim to not only find stable matchings for the resource allocation problems but also maximum length matchings. Consequently, a maximum number of user requests should be served at a time. The models are implemented in C++ using the SCIP solver. Numerical experiments are conducted for large data sets and the results are obtained.

The rest of the paper is organized as follows. The next section recalls the background regarding ILP formulations established for the related SMP variants. We then describe the resource allocation problem and formulate the ILP model accordingly in Section 3. Experiments will be conducted and discussed in Section 4. Finally, some concluding remarks are given in Section 5.

2. MAX-SMTI AND MAX-HRT BACKGROUNDS

We now recall the mathematical models for two problems which are MAX-SMTI and MAX-HRT, detailed descriptions can be found in [4]. We keep here all terminologies given in [4] for simplicity. An instance of SMTI consists of a set C of n_1 children and a set F of n_2 families. Each child (family, respectively) ranks a subset of families (children, respectively) in an order of preference, where ties are allowed. Let $c \in C$ and $f \in F$, c (f , respectively) finds f (c , respectively) acceptable if f (c , respectively) is in c 's (f 's, respectively) preference list. (c, f) is called to be acceptable pair. A matching M in I is defined to be a subset of acceptable pairs. If a member (child or family) is not matched in M , it is said to be single (or unmatched). The size of M is called cardinality.

Let $(c, f) \in M$, we denote by $M(c)$ and $M(f)$ the partner of f and c in M , respectively, i.e., $f = M(c)$ and $c = M(f)$. A pair (c, f) blocks M if: (i) (c, f) is an acceptable pair, (ii) either c is single or prefers f to $M(c)$, and either (iii) f is single or prefers c to $M(f)$. Such a pair (c, f) is called a blocking pair. A matching M is said to be stable if it admits no blocking pair. In the setting of SMTI, as mentioned, we can find different stable matchings with different cardinalities. We aim to find a maximum cardinality stable matching in the MAX-SMTI. Let us denote by:

- $F(i)$ the set of families acceptable for child i , for $i = 1, 2, \dots, n_1$.
- $C(j)$ the set of children acceptable for family j , for $j = 1, 2, \dots, n_2$.
- $r_j^c(i)$ the rank of family j for child i , defined as the integer k such that j belongs to the k^{th} most-preferred tie in i 's list, for $i = 1, 2, \dots, n_1, j \in F(i)$. The smaller the value, the better family is ranked.
- $r_i^f(j)$ the rank of child i for family j , defined as the integer k such that i belongs to the k^{th} most-preferred tie in j 's list, for $j = 1, 2, \dots, n_2, i \in C(j)$. The smaller the value, the better child is ranked.
- $F_j^{\leq}(i)$ the set of families that child i ranks at the same level or better than family j , i.e., $F_j^{\leq}(i) = \{j' \in F: r_{j'}^c(i) \leq r_j^c(i)\}$, for $i = 1, 2, \dots, n_1, j \in F(i)$.

- $C_i^{\leq}(j)$ the set of children that family j ranks at the same level or better than child i , i.e., $C_i^{\leq}(j) = \{i' \in C: r_{i'}^f(j) \leq r_i^f(j)\}$, for $j = 1, 2, \dots, n_2, i \in C(j)$.

Let x_{ij} be the binary decision variables, where $x_{ij} = 1$ if child i is matched with family j , otherwise $x_{ij} = 0$. The model for MAX-SMTI is given as follows.

$$\max \sum_{i=1}^{n_1} \sum_{j \in F(j)} x_{ij} \quad (P_{\text{MAX-SMTI}})$$

s.t

$$\sum_{j \in F(i)} x_{ij} \leq 1, i = 1, 2, \dots, n_1 \quad (1)$$

$$\sum_{i \in C(j)} x_{ij} \leq 1, j = 1, 2, \dots, n_2 \quad (2)$$

$$1 - \sum_{q \in F_j^{\leq}(i)} x_{iq} \leq \sum_{p \in C_i^{\leq}(j)} x_{pj}, i = 1, 2, \dots, n_1; j \in F(i) \quad (3)$$

$$x_{ij} \in \{0, 1\}, i = 1, 2, \dots, n_1; j \in F(i) \quad (4)$$

In $(P_{\text{MAX-SMTI}})$, every agent (family or child) can engage at most a partner. This one is ensured by Eqs. (1) and (2). Eq. (3) is to make sure that there is no blocking pair in the resulting matching.

The HRT problem can be seen as a generalized form of the SMTI problem. In the HRT setting, each hospital has a capacity to host a number of doctors, we denote by c_j , for $j = 1, 2, \dots, n_2$. In the same manner, the model of MAX-HRT is given as $(P_{\text{MAX-HRT}})$. Constraints (6) and (7) include the term c_j to adapt the hosting capacity of hospitals and the blocking pair definition in the HRT problem, respectively.

$$\max \sum_{i=1}^{n_1} \sum_{j \in H(j)} x_{ij} \quad (P_{\text{MAX-HRT}})$$

s.t

$$\sum_{j \in H(i)} x_{ij} \leq 1, i = 1, 2, \dots, n_1 \quad (5)$$

$$\sum_{i \in D(j)} x_{ij} \leq c_j, j = 1, 2, \dots, n_2 \quad (6)$$

$$c_j \left(1 - \sum_{q \in H_j^{\leq}(i)} x_{iq} \right) \leq \sum_{p \in D_i^{\leq}(j)} x_{pj}, i = 1, 2, \dots, n_1; j \in H(i) \quad (7)$$

$$x_{ij} \in \{0, 1\}, i = 1, 2, \dots, n_1; j \in H(i) \quad (8)$$

Here, we replace the term ‘‘family’’ with ‘‘hospital’’, then change $F(i)$, $r_i^f(j)$, and $F_j^{\leq}(i)$ into $H(i)$, $r_i^h(j)$, and $H_j^{\leq}(i)$, respectively. Similarly, the term ‘‘child’’ is replaced by ‘‘doctor’’, and $C(j)$, $r_j^c(i)$, and $C_i^{\leq}(j)$ are changed into $D(j)$, $r_j^d(i)$, and $D_i^{\leq}(j)$, respectively. We now apply two models given by $(P_{\text{MAX-SMTI}})$ and $(P_{\text{MAX-HRT}})$ for the problem of resource allocations in networking environments.

3. FORMULATIONS FOR RESOURCE ALLOCATIONS

This section considers two use-cases arising in fog computing which are depicted in Fig. 1. Basically, a fog architecture can be composed of three layers, including IoT, fog, and cloud layers [18]. The IoT layer at the bottom is to collect data via sensors. Fog devices are located in the fog layer to store and process streaming collected data, and communicate to the others. The cloud layer plays the role of big data processing which can be performed via a set of virtual machines. The first will deal with the resource allocation problem in a fog layer where we need to allocate necessary resources to respond to user requests at a time. We then consider the problem of virtual machine allocations in the cloud layer in the second use case.

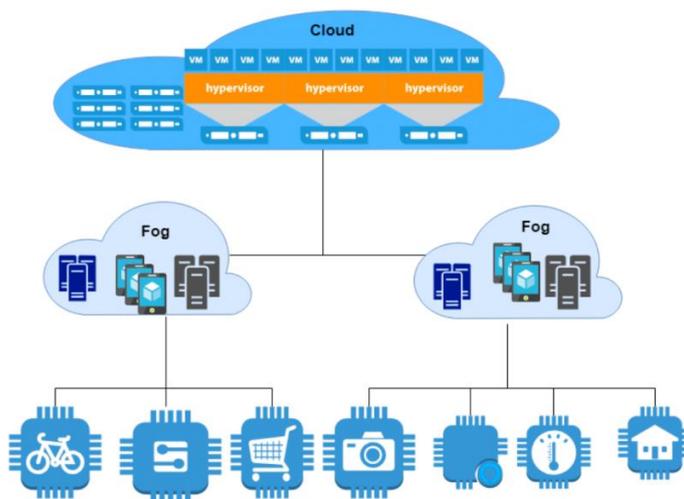


Figure 1. An overview architecture of fog computing [2].

3.1. Minimal cost resource allocating in fog computing

In a fog environment, many user devices can be involved. Every user can send requests to be served and fog devices will need to respond to users as providers. It is required to determine suitable resources to serve as many requests as possible so that the cost is minimum, while still keep the benefits of both users and providers. In particular, we first determine how many available providers can meet the user requirements, then decide to accept or reject to serve. If accepting a user request, we need to allocate resources among currently available fog devices so that it fulfills the request while keeping a minimum cost. There are several cost estimation models presented in the literature. A micro-level compensation-based cost model is proposed by Battula *et al.* in 2019 [2]. After estimating the cost, the authors also presented a stable marriage-based algorithm for resource allocation. We focus on this task by summarizing the problem statement, then formulate the under an ILP form. The method presented in [2] is to match the best-suited resource to all user requests and the available resources. To this end, they used a modified Gale and Shapley’s algorithm for the following stable marriage problem.

- Input: a list of user requests denoted by UR , and a list of free devices denoted by fd . The preference of the user list is formed due to the requirements and the capacity of the fog devices.

- Output: a stable matching in the form $\langle UR, fd \rangle$.

The algorithm first determines available devices which meet the user requirements. It then iteratively performs the matching task as follows. The device fd checks the UR 's preference. If df can satisfy the requirement, fd is mapped to the user request. Then it compares the fd 's preference with the fog device previously assigned to the user request. If it prefers fd to the assigned one, fd will be assigned to the user request. The algorithm continues to check for all user requests or until all available resources are allocated. This gives a stable matching between user requests and available fog devices but does not shows maximum length ones. We next formulate the problem under an integer linear program for the MAX-SMTI form of the problem.

As such setting of a stable marriage problem, we now formulate it under an ILP form for efficiently solving. Let I be an instance consisting of $UR = \{ur_1, ur_2, \dots, ur_{n_1}\}$ and $FD = \{fd_1, fd_2, \dots, fd_{n_2}\}$. We use the binary decision variable x_{ij} to express an acceptable pair in I which is formed by a user request ur_i and a fog device fd_j , for $i = 1, 2, \dots, n_1$ and $j = 1, 2, \dots, n_2$. x_{ij} indicates that either ur_i can be matched with fd_j or not. If $x_{ij} = 1$ in a matching x , ur_i is matched with fd_j , otherwise ur_i is not suitable for fd_j . Let $rank(ur_i, fd_j)$ be the rank of fd_j in ur_i 's preference list. We define $rank(ur_i, fd_j)$ to be $k + 1$, where k is the number of fog devices that ur_i prefers to fd_j . For fds , all user requests in fd_j 's preference list are equally preferred. We also note that $rank(ur_i, fd_j) = rank(fd_j, ur_i) = \infty$ for an unacceptable pair (ur_i, fd_j) . For each pair (ur_i, fd_j) , we also define

$$U_{i,j} = \{ur_p \in UR : rank(fd_j, ur_p) \leq rank(fd_j, ur_i)\},$$

and

$$F_{i,j} = \{fd_q \in FD : rank(ur_i, fd_q) \leq rank(ur_i, fd_j)\}.$$

Applying the model given by $(P_{MAX-SMTI})$ results in the ILP model for the resource allocation as shown in (P_{RA}) .

$$\max \sum_{i=1}^{n_1} \sum_{fd_j \in FD} x_{ij} \quad (P_{RA})$$

s.t

$$\sum_{fd_j \in FD} x_{ij} \leq 1, i = 1, 2, \dots, n_1,$$

$$\sum_{ur_i \in UR} x_{ij} \leq 1, j = 1, 2, \dots, n_2,$$

$$1 - \sum_{fd_q \in F_{i,q}} x_{iq} \leq \sum_{ur_p \in U_{p,j}} x_{pj}, i = 1, 2, \dots, n_1; j \in FD,$$

$$x_{ij} \in \{0, 1\}, i = 1, 2, \dots, n_1; j \in FD.$$

3.2. Virtual machine allocation

This section considers a virtual machine allocation problem in the cloud layer depicted in Fig. 1. We use the scenario given in [11] as our case study, which is briefly described as follows. Let us consider a cloud data center consisting of n_2 physical servers with different resource capacities. At a time, a number of users send requests to n_1 virtual machines. The system needs

to determine all possible servers which are available to serve, then perform to allocate virtual machines to the best suitable servers.

Let $V = \{v_1, v_2, \dots, v_{n_1}\}$ be the virtual machines, $S = \{s_1, s_2, \dots, s_{n_2}\}$ be the sever in the cloud. The capacity of $s_j \in S$ is denoted by c_j , for $j = 1, 2, \dots, n_2$. Each virtual machine $v_i \in V$ ranks the servers in an increased order of transmission costs, which is computed by

$$\phi(v_i, s_j) = d_{s(v_i), s_j} / \theta_{s(v_i), s_j},$$

where $\phi(v_i, s_j)$ denotes the transmission cost, $s(v_i)$ is the server which v_i is residing, $d_{s(v_i), s_j}$ is the hop distance from $s(v_i)$ to s_j , and $\theta_{s(v_i), s_j}$ is the bandwidth between $s(v_i)$ and s_j . We also note that ties in the preferences of virtual machines are allowed due to the equal transmission costs. Similarly, each server $s_j \in S$ also ranks the virtual machines in increasing order of migration overhead, defined by

$$\omega(s_j, v_i) = d_{s(v_i), s_j} \cdot \zeta_{s(v_i), s_j},$$

where, $\zeta_{s(v_i), s_j}$ denotes the VM disk image size in bytes [24]. Let x_{ij} be the binary decision variable, indicating whether virtual machine v_i is assigned to the server s_j . Replacing the roles of H and D in $(P_{\text{MAX-HRT}})$, by S and V , respectively, we obtain the following equation for the virtual machine allocation.

$$\begin{aligned} & \max \sum_{i=1}^{n_1} \sum_{j \in \mathcal{S}(j)} x_{ij} && (P_{\text{VM}}) \\ & \text{s.t} \\ & \sum_{j \in \mathcal{S}(i)} x_{ij} \leq 1, i = 1, 2, \dots, n_1 \\ & \sum_{i \in \mathcal{V}(j)} x_{ij} \leq c_j, j = 1, 2, \dots, n_2 \\ & c_j \left(1 - \sum_{q \in \mathcal{S}_j^{\leq}(i)} x_{iq} \right) \leq \sum_{p \in \mathcal{V}_i^{\leq}(j)} x_{pj}, i = 1, 2, \dots, n_1; j \in \mathcal{S}(i), \\ & x_{ij} \in \{0; 1\}, i = 1, 2, \dots, n_1; j \in \mathcal{S}(i). \end{aligned} \quad (9)$$

It differs from the algorithm proposed in [11] that (P_{VM}) finds not only a stable virtual machine allocation but also the maximum length matching in the resulting allocation. This consequently helps to improve the performance of the system.

4. EXPERIMENTS RESULTS

In this section, we concentrate on the task of resource allocation for a given number of user requests and free fog devices. Unlike in [2], our method aims to respond as maximum as possible requests. Furthermore, we also conduct experiments for large data sets which can be randomly generated. One of the methods which can be used to generate random SMTI instances was given in [8]. Our models were implemented in C++ using SCIP¹ for ILP solutions and run

¹<https://www.scipopt.org>

on the Ubuntu platform, CPU Intel Core i5 – 3320M 2.60GHz, 2456 MHz, 2 Cores, and 8GB RAM. In all experiments, we used the default setting of the SCIP solver. In this setting, for solving relaxed linear programs, SoPlex is used. It performs first some preprocessing steps to reduce the size of the problem before applying the simplex algorithm, such as reducing the singleton rows and detecting the linear dependencies. For branching and cutting steps, the branch rule was applied to create two subproblems by splitting a single variable's domain, while the cutting plane separation method is used.

Table 1. Detail matching results of equal number of user requests and free fog devices.

URs	fds	Limited preference list	Results		Runtime (second)
			Pair number	% (per UR)	
100	100	10	94	94	0.448
		30	98	98	0.776
		50	100	100	1.427
200	200	10	103	51.5	0.482
		30	168	84	7.627
		50	188	94	14.043
300	300	10	143	47.67	1.132
		30	251	83.67	7.647
		50	279	93	14.043
400	400	10	151	37.75	1.311
		30	286	71.5	7.019
		50	345	86.25	15.78
500	500	10	158	31.6	1.979
		30	295	59	25.873
		50	390	78	59.880

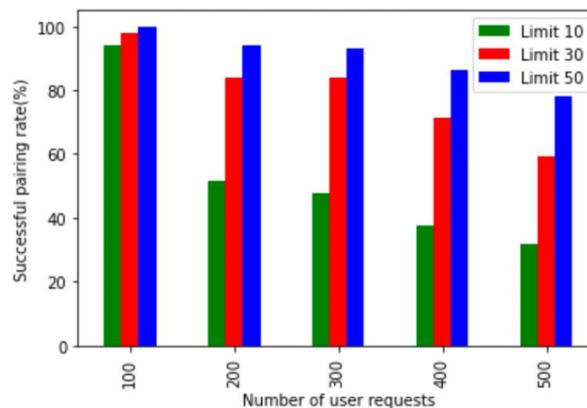


Figure 2. Matching results compared between three limited lengths of preference lists.

Table 1 shows the experiment results for several data sets of an equal number of user requests and free fog devices. The sizes of data sets are 100, 200, 300, 400, and 500. For each data set, we tested for three different limited lengths of the preference lists which are 10, 30, and 50. We report here the results of matching numbers and runtime. Generally, smaller numbers of *URs* and *fds* get better matchings, even when the preference list length is small. When the preference list length is 50, most of the cases obtain more than 80 % of user requests matched

(see Fig. 2). This indicates that ILP models for solving the MAX-SMTI based resource allocation are useful.

Regarding the execution time, as shown in Fig. 3, it takes about a second to complete pairing *URs* and *fds* for the preference list length of 10, for all the sizes from 100 to 500. This is acceptable to obtain maximum cardinality matchings. Naturally, the execution time is increased as the preference list length gets higher.

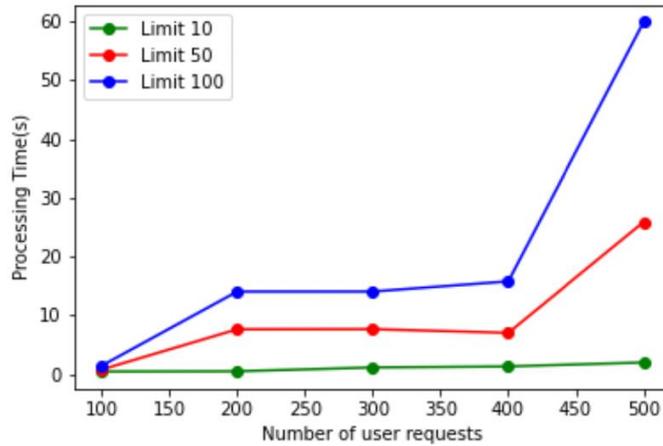


Figure 3. The runtime of three limited lengths of preference lists.

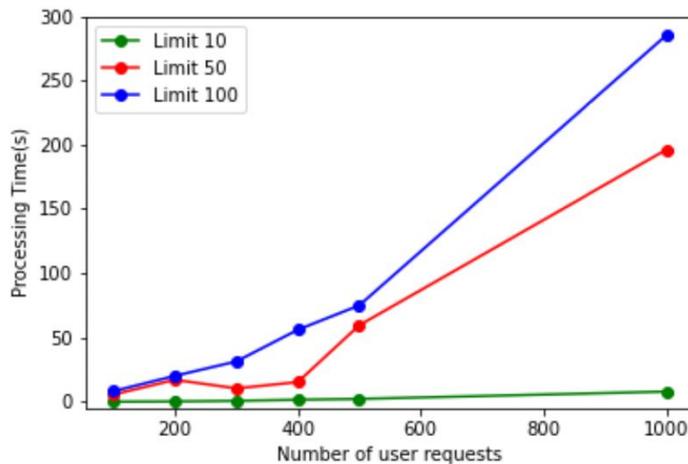


Figure 4. Runtime for a large number of user requests.

Figure 4 shows the execution times for larger data sets. It indicates that when the size of *URs* and *fds* is large, the time grows quickly for completing the matching. This is because of the time required for solving large ILP models. However, we also note that the models we are using in this work are in basic forms. We have not yet exploited any properties of stable matching structures of given SMTI instances which can help to eliminate unnecessary blocking pairs during searching. Consequently, it can help to have better ILP formulations for quickly obtaining solutions. This has been shown in [4], where the authors introduced some pre-processing tasks for such eliminations. On the other hand, they also improved ILP models by reducing the

number of non-zero elements and used alternative stability constraints. Computational results showed that the execution time is significantly reduced for those hard variants of SMPs. It is straightforward to apply these techniques in our work to improve performance.

Table 2. Results obtained when the number of requests is greater than that of free devices.

URs	fds	Limited preference list	Pair number	Results		Runtime (second)
				% (per UR)	% (per fd)	
100	90	10	60	60	66.667	0.203
		30	80	80	88.889	0.76
		50	81	81	90	1.456
200	180	10	100	50	55.556	0.443
		30	159	79.5	88.333	2.232
		50	160	80	88.889	5.293
300	270	10	107	35.667	39.63	0.893
		30	206	68.667	76.3	7.58
		50	238	79.333	88.148	11.517
400	360	10	112	28	31.111	1.72
		30	225	56.25	62.5	7.485
		50	286	71.5	79.444	17.14
500	450	10	118	23.6	26.222	2.379
		30	268	53.6	59.556	24.648
		50	340	68	75.556	28.081

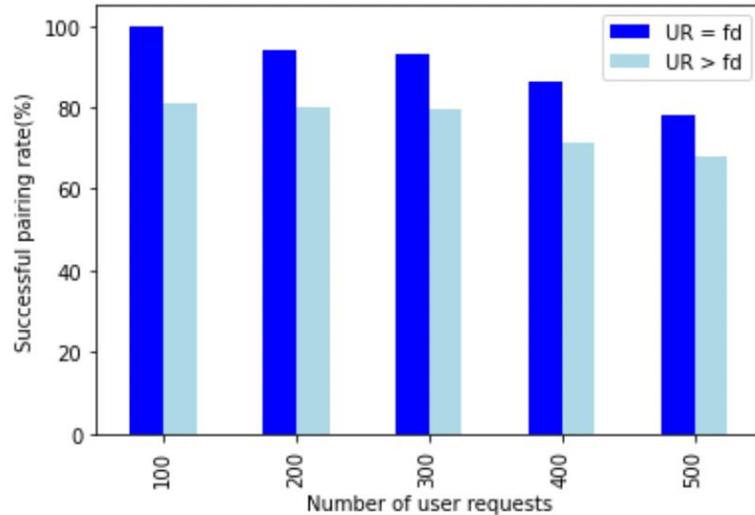


Figure 5. Successfully pairing ratio compared between two cases of $|URs| = |fds|$ versus $|URs| > |fds|$ with respect to the user request.

We next perform experiments for the case of unequal numbers of user requests and fog devices. We consider here percentages of successfully pairing in both user requests and free fog devices. Table 2 shows the results obtained. It can be observed that for all data sets, the ratio of successfully pairing in the fog device side is higher than that on the user side. This is because of the smaller number of free fog devices which are available to serve the user requests. For the same reason, on the other hand, the ratio of successfully pairing is lower when compared with

that given in Table 1 (see also Fig. 5 for a comparison we plotted for the case of limited preference list length of 50). All of these show the correctness of the proposed method.

5. CONCLUSIONS

The paper studied an approach to use stable marriages for the resource allocation problem in some modern networking environments. We considered two basic problems that are device allocations for user requests in the fog and virtual machine allocation in cloud computing layers of an IoT architecture. We aim to address the problem for not only large datasets but also as maximum as possible pairing number of obtained matchings. The experimental results for large data sets show the correctness as well as the usefulness of the proposed approach. The ILP models used in this work are given in the basic formulation of MAX-SMTI and MAX-HRT. This can be improved using the structural feature of the stable matchings as well as applying a better formulation of ILP forms. This will be another content in the next work to have a comprehensive study in this application of SMP.

Acknowledgments. This research is funded by the Vietnam National Foundation for Science and Technology Development (NAFOSTED) under grant number **102.01-2017.09**.

CRedit authorship contribution statement. Le Hong Trang: Methodology, Investigation, Experiment design, Analysis, Funding acquisition. Hoang Huu Viet: Implementation, Result evaluation.

Declaration of competing interest. The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

REFERENCES

1. Abraham D. J., Irving R. W. and Manlove D. F. - The student-project allocation problem, in Proceedings of the 14th International Symposium, Kyoto, Japan, Dec. 2003, pp. 474–484.
2. Battula S. K., Garg S., Naha R. K., Thulasiraman P. and Thulasiram R. - A micro-level compensation-based cost model for resource allocation in a fog environment, *Sensors* **19** (13) (2019) 2954.
3. Chu Q., Cui L. and Zhang Y. - Joint Computing and Storage Resource Allocation Based on Stable Matching in Data Centers, IEEE 3rd international conference on big data security on cloud, 2017, pp. 207-212.
4. Delorme M., García S., Gondzio J., Kalcsics J., Manlove D. F., and Petterson W. - Mathematical models for stable matching problems with ties and incomplete lists, *European Journal of Operational Research* **277** (2) (2019) 426-441.
5. Fleiner T., Irving R. W., and Manlove D. F. - Efficient algorithms for generalized stable marriage and roommates problems, *Theoretical Computer Science* **381** (1-3) (2007) 162-176.
6. Gale D. and Shapley L. S. - College admissions and the stability of marriage, *The American Mathematical Monthly* **9** (1) (1962) 9-15.
7. Gelain M., Pini M. S., Rossi F., Venable K. B. and Walsh T. - Local search approaches in stable matching problems, *Algorithms* **6** (1) (2013) 591-617.

8. Gent I. P. and Prosser P. - An empirical study of the stable marriage problem with ties and incomplete lists, Proceedings of the 15th European Conference on Artificial Intelligence, Lyon, France, Jul. 2002, pp. 141-145.
9. Gusfield D. and Irving R. W. - The stable marriage problem: structure and algorithms, MIT Press Cambridge, 1989.
10. Irving R. W. - An efficient algorithm for the “stable roommates” problem, Journal of Algorithms **6** (1)(1985) 577-595.
11. Kim G. and Lee W. - Stable matching with ties for cloud-assisted smart tv services, Proceedings of the 2014 IEEE International Conference on Consumer Electronics (ICCE), 2014, pp. 558–559.
12. Kiraly Z. - Linear time local approximation algorithm for maximum stable marriage, Algorithms, **6** (3) (2013) 4.
13. Kishk S., Almofari N. H., and Zaki F.W. - Distributed resource allocation in D2D communication networks with energy harvesting relays using stable matching, Ad Hoc Networks **61** (2017) 114-123.
14. Kwanashie A. and Manlove D. F. - An integer programming approach to the hospitals/residents problem with ties, Operations Research Proceedings, 2014, pp. 263-269.
15. Manlove D. F., Irving R. W., Iwama M., Miyazaki S., and Morita Y. - Hard variants of stable marriage, Theoretical Computer Science, **276** (1-2) (2002) 261-279.
16. McDermid E. J. - A $3/2$ -approximation algorithm for general stable marriage, Proceedings of International Colloquium on Automata, Languages and Programming (ICALP), 2009, pp. 689-700.
17. Munera D., Diaz D., Abreu S., Rossi F., Saraswat V. and Codognet P. - Solving hard stable matching problems via local search and cooperative parallelization, Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, Austin, Texas, Jan. 2015, pp. 1212-1218.
18. Naha R. K., Garg S. and Chan A. - Big Data-Enabled Internet of Things, Institution of Engineering and Technology (IET) Digital Library, Fog-computing architecture: survey and challenges, 2019, pp. 199-224.
19. Roth A. E. - The evolution of the labor market for medical interns and residents: A case study in game theory, Journal of Political Economy **92** (6) (1984) 991-1016.
20. Vate J. H. V. - Linear programming brings marital bliss, Operations Research Letters **8** (3) (1989) 147-153.
21. Viet H. H., Uyen N. T., Lee S. G., Chung T. C., and Trang L. H. - Amax conflictsbased heuristic search for the stable marriage problem with ties and incomplete lists, Journal of Heuristics **27** (2021) 439-458.
22. Wang J. V., Fok K., Cheng C., and Tse C. K. - A stable matching-based virtual machine allocation mechanism for cloud data centers, IEEE World Congress on Services, 2016, pp. 103-106.
23. Long D. T. - An enhanced genetic algorithm-based courses timetabling method for maximal enrollments using maximum matching on bipartite graphs, Vietnam Journal of Science and Technology **57** (6) (2019) 734-748.

24. Xu H. and Li B. - Egalitarian stable matching for VM migration in cloud computing, Proceedings of the IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), 2011, pp. 631-636.
25. Zhou Z., Gao C., Xu C., Chen T., Zhang D. and Mumtaz S. - Energy-Efficient Stable Matching for Resource Allocation in Energy Harvesting-Based Device-to-Device Communications, IEEE Access **5** (2017) 15184-15196.
26. Toan P. T. and Loc N. T. - A branch and bound algorithm for workflow scheduling, Vietnam Journal of Science and Technology **56** (2) (2018) 246-256.