International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. #, No.# (Mon-20..)

A Novel Matching Framework For One-Sided Markets In Fog Computing

Prarena Shroff[1], Anjan Bandyopadhyay[1, 2]

[1] Department[1] of Computer Science and Engineering, Amity University, Kolkata, India
[2]Department of Computer Science and Engineering, National Institute of Technology, Durgapur, India

E-mail address: pshroff76@gmail.com, anjanmit@gmail.com

Received ## Mon. 20##, Revised ## Mon. 20##, Accepted ## Mon. 20##, Published ## Mon. 20##

Abstract: Resource allocation is a challenging task for every researcher in fog and cloud computing. This problem has been widely studied in a fog computing environment where there is a constant need for resource allocation. Many mechanisms have been proposed for buying and selling of resources where one or both sides give their preferences over each other. However, the challenge of allocating resources becomes difficult if the service provider wants to pick their most preferred user. In such cases, the users may also consider forming a coalition to improve their overall welfare in a cooperative environment. In this scenario, we have proposed a novel matching mechanism named as stable fog resource allocation (SFRA) in the fog computing environment. To the best of our knowledge, this highly possible framework of allocating resources based on preferences in cooperative and non-cooperative fog environment has not been proposed before. First, the proposed mechanism is studied in the paper then its implementation is done-with and without a coalition.

Keywords: Fog computing, resource allocation, coalition, mechanism design, stable matching.

1. Introduction

Cloud computing has emerged as one of the most sought after technologies in the world. It provides a wide array of resources tailored to the user's demands over the web and on very flexible pricing schemes. However, with the huge amount of data being produced, there is a need for a technology that brings the processing closer to the end-user [1]. To solve this issue, CISCO proposed a new era of upcoming technology called fog computing [2] to reduce latency and response time. Fog computing provided a complement to the cloud solution, to adjust to the emerging IoT vision [3]. In fog computing, each fog node is connected to IoT devices which create the need for an efficient resource allocation framework as in cloud [4], where buying and selling can be done. In this paper, we are dealing with a setup where there are many sellers for a particular service and consequently many buyers who want to avail that service. This can be put into one-sided markets i.e. where only one side has their preferences on the other side [4]. Moreover, agents (or users) may try to work cooperatively in an environment where they can benefit by forming a coalition such that no agent is better off alone. The use of coalitions in such environments has been an active area of research as in [5] which provides a game-theoretic framework for forming stable allocations. But to our knowledge, we surprisingly found out that there is hardly a mechanism that matches the users (single or in coalition) to the service providers in a one-sided market [6].

Thus we see how such a matching mechanism will be important for proper allotment of resources. The real-life applications of resource allocation are in various domains like healthcare, educational institutions, communication networks and electronic marketplaces.

To provide a solution to this problem we have presented an algorithm called Stable Fog Resource Allocation (SFRA) in this paper which can be used to form service provider and user pairs in both cooperative and non-cooperative settings.

The remainder of the paper is organized as follows: some prior works in this area with different setups are discussed in Section 2. It is followed by the system model and the problem formulation in Section 3.

E-mail: pshroff76@gmail.com, anjanmit@gmail.com

Then we present our algorithm in both the settings in detail in Section 4. A detailed analysis of the experimental result is carried out in Section 5 and finally, we present a summary of our work and highlight some future directions in Section 6.

2. LITERATURE REVIEW

The problem of allocating resources in an economic setting has been a challenge in a buyer-seller environment. The problems are based on different kinds of markets where buyers and sellers give their preferences over each other. In these kinds of scenarios arises a need to match the buyers and sellers according to some matching mechanism in that particular market. The concepts of such markets are found in game theory [6] where there are one-sided [6] and two-sided markets [7, 8] depending upon whether one or both sides give their preferences. The paper [9] provides a comprehensive overview and difference between one-sided and two-sided markets. The one-sided matching model can be found in [4] which returns matched service-provider pairs where users had a strict set of preferences. The idea of optimality for one-sided preference matching models was put forward in [10] which maximized the area under the profile curve ratio metric. The applications of two-sided matching models can also be found in [11, 12, 13]. In [14], a many-to-one matching was designed as a joint user association and resource allocation problem to handle the increasing demand for quality of services (QoS) requirements in the downlink of the fog network. Moreover, there are other techniques like auction mechanisms [15] which have emerged as a popular price determination technique in diverse computing environments like in cloud [16,17,18] and edge computing [19, 20, 21]. In the field of fog computing, there is a major requirement for methods that compensate the service providers for the use of their resources by the users. Such an approach implementing combinatorial auctions [22] was found in [23] and in [24] which presented a double sided combinatorial auction for maximizing the welfare of both users and service providers. The study of welfare has been an area of active research in the past which has led to many discussions especially for one-sided [25, 26, 27] and two-sided markets [28]. Furthermore, there may be a scenario where users or service providers choose to form a coalition [29] so that they can benefit by increasing their profits and handle the problem of some fog nodes having low computational capabilities. The concept of cooperative game theory [30] has been widely explored and has led to many algorithms in cloud [31] and fog [5, 32, 33, 34, 35] computing.

However, we could not find a matching mechanism that was suited for one-sided markets and was adaptable to be implemented in both cooperative and non-cooperative environments. Hence in this paper, we have proposed a novel allocation technique that is stable, Pareto-optimal and returns a perfect matching as the solution.

3. SYSTEM MODEL AND PROBLEM FORMULATION

The proposed model can be described using a one-sided preference-based matching market model. The model consists of n number of users denoted by $U = \{u_1, u_2, \ldots, u_n\}$ who want to gain access to resources from n number of service providers denoted by $S = \{s_1, s_2, \ldots, s_n\}$. Each service provider here gives a complete set of preferences over the n number of users. As this is one-to-one matching, one user can be allocated to at most one service provider and vice-versa. Each service provider has a total ordering on the users in the system. The ordering of service providers is done arbitrarily. For example, if there are three users u_1, u_2, u_3 and three service providers s_1, s_2, s_3 one of the possible total ordering could be:

1) $u_3 \succ_{s_1} u_1 \succ_{s_1} u_2$
2) $u_2 \succ_{s_2} u_1 \succ_{s_2} u_3$
3) $u_2 \succ_{s_3} u_3 \succ_{s_3} u_1$

Hence, the preference matrix for the above-illustrated example is given in Table I. The preferences are in descending order (left to right). Hence s_1 prefers u_3 first then u_1 and lastly u_2. The same kind of ordering follows for s_2 and s_3. However, in our paper, the ties in rankings are not considered that is no two or more users can have the same preference ranking from that service provider.

TABLE I. PREFERENCE MATRIX OF SERVICE PROVIDERS

s_1	u_3	u_1	u_2
s_2	u_2	u_1	u_3
s_3	u_2	u_3	u_1

Users want to get allocated as fast as possible while service providers want to get their most preferred available pick amongst the users. This leads to a one-to-one matching scenario for which we have proposed an approach.

4. PROPOSED APPROACH

At first, we see our proposed mechanism which is called SFRA, where each user gets allocated to a service provider and we get service provider and user matched pairs as a result.

Algorithm 1 Stable Fog Resource Allocation Mechanism (SFRA)

Input: $U = \{u_{1,} u_{2\,,} \dots, u_n\}$, $S = \{s_1, s_2, \dots, s_n\}$, $\succ_s = \{\succ_{s_1}, \succ_{s_2}, \dots, \succ_{s_n}\}$, $\forall_i = \{0, 1, 2, \dots, n\}$, $\forall_j = \{0, 1, 2, \dots, n\}$

Output: $A \leftarrow \emptyset$

1: **begin**
2: ***for*** each $i \in S$ do
3: $A[i] \leftarrow \emptyset$
4: ***end***
5: ***while S′*** $\neq \emptyset$ **do**
6: $S^* \leftarrow S_i$
7: $U^* \leftarrow \emptyset$
8: $count \leftarrow 0$
9: **for** each $u_j \in \succ_S$
10: **if** $u_j \in A$
11: $count \leftarrow count + 1$
12: **end**
13: **if** $count == \emptyset$ **then**
14: $U^* \leftarrow u_j$
15: $A[S^*] \leftarrow U^*$
16: $U' \leftarrow U' \backslash U^*$
17: $S' \leftarrow S' \backslash S^*$
18: **break**
19: **end**
20: **end**
21: $i \leftarrow i + 1$
22: **end**
23: **end**

A. Outline of SFRA

1) Creating the preference matrix: In the considered scenario of n service providers and n users, each service provider has a total ordering ($\succ_s$) over the n number of users. Now, this preference list is randomly generated where the preferences are in between the range 1 to n considering each number occurs only once in each combination of those n digits. No repetitions are allowed however two or more same combinations can be allowed.

By this, we mean that two or more service providers may have the same list of preferences. The input to the algorithm is a $n\ x\ n$ preference matrix as shown in *Fig. 1* which is used to obtain service provider and user pair which is stored in a data structure A.

2) Running the SFRA mechanism: Each service provider has a list of preferences as shown in Section III. In line 2-4, no service provider has an allocation i.e. they are assigned a null value. Line 5-23 allocates users to service providers in a greedy fashion until all service providers are allocated. By this, we mean that the algorithm first searches the matrix for that service provider's first choice and checks if it has been already allocated i.e. if it is in A , as in line 10-12. If found in A , the next best choice's availability is searched and so on until the service provider gets allocated. If the first choice is not found in A, it allocates its first choice to it, as in line 13-19.

3) Updating the data structure: Now after each allocation, the system has to be updated so that future allocations can be done. In line 15-17, the allocated user is added to A and the service provider and user pair are removed from the unallocated service provider (S') and user data structure (U'). After updating, the control breaks out of the loop and the system gets ready for the next allocation. The while loop in line 5 runs until all the service providers have been allocated. Thus a one-to-one matching is obtained and we get the service provider and user pair as an output A.

Our proposed SFRA mechanism can be run on a coalition environment as well for which we have put forward Algorithm 2 named as Coalition Fog Resource Allocation Mechanism (CFRA) which presents how can the SFRA mechanism can be adapted into a cooperative setting. The outline of the CFRA mechanism is as follows.

B. Outline of CFRA

1) Formation of coalition: The next stage in our proposed mechanism is Algorithm 2, which forms the coalition, updates the matrix and runs Algorithm 1 on it. To form a coalition there should be two or more members willing to participate in the matching cooperatively. In our algorithm, we have not focused on the optimization problem of choosing the players of the coalition and the size in regards to coalition formation. However, that is considered to be the system's input to the algorithm. Under the assumption that the coalitions requested are valid, the coalitions are formed. In Algorithm 2, in line 5-9, each player in a coalition has a rank according to the allocation in a non-coalition scenario obtained from Algorithm 1.

In line 10-15, the ranks of all the users in a coalition are then compared and whoever has the least rank i.e. whoever was allocated first becomes the winner. In line 16, the winner is added to the data structure w which stores the winner of each coalition c_k .

Algorithm 2 Coalition Fog Resource Allocation Mechanism (CFRA)

Input: $C[a][b] =$ where each row has the players forming the coalition, a =maximum number of coalitions, b = maximum number of players that can take part in a coalition, $\forall_i = \{0, 1, 2, \dots, n\}, \forall_k = \{0, 1, 2, \dots, a\}$, $\forall_j = \{0, 1, 2, \dots, b\}$

Output: $A' \leftarrow \emptyset$

1: **begin**

2: Obtain A from Algorithm 1

3: $k \leftarrow 0$

4: $p^* \leftarrow b$

5: **for** each $i \in A$

6: find index of i where $A[u_i] = i$

7: store index in rank $[k]$

8: $k \leftarrow k + 1$

9: **end**

10: **for** each $c_k \in C$

11: **for** each $p_j \in c_k$

12: **if** rank $(p_j) <$ rank (p^*) **then**

13: $p^* = p_j$

14: **end**

15: **end**

16: $w[c_k] \leftarrow p^*$

17: $l[c_k] \leftarrow c_i \backslash w[c_k]$

18: replace each $l[c_k]$ as a dummy in C

19: **end**

20: shift dummy variables at the end in C

21: run Algorithm 1 on updated matrix

22: Obtain A' from Algorithm 1

23: **end**

In line 17, the rest are added to the data structure l representing the losing/supporting side. This is done for all the coalitions and thus we get a winner for each coalition. Now in line 18, all the losers or supporting users are replaced in the original preference matrix as dummy users. We assume here that a user can take part in only one coalition. Now in line 20, the dummy users have to be shifted towards the end of each list so that the preferences can be updated. The input still remains a $n\ x\ n$ matrix thus not changing the input size for Algorithm 1.

2) Running the SFRA mechanism on the coalition matrix: Now in line 21, we run our SFRA on the updated coalition matrix and obtain service provider-pair and store it in a data structure A'.The service providers who get dummy users are not included in the final result and are discarded.

C. Illustrative Example

The mechanism first generates the $n \text{ x } n$ matrix randomly for $n = 5$ users and $n = 5$ service providers. *Fig. 1* shows the 5 x 5 matrix where each service provider's strict preferences are given over users.

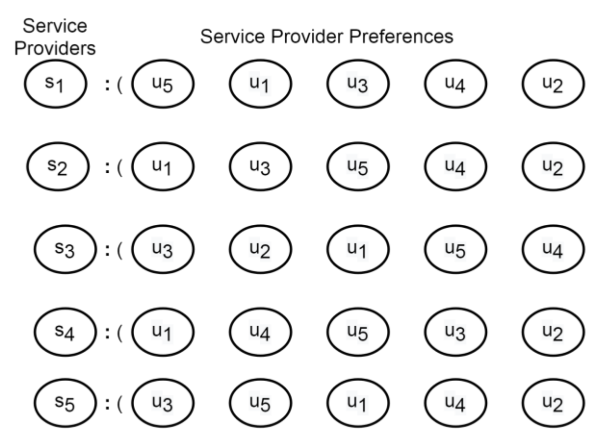

Figure 1. Service provider preference matrix in descending order from left(best) to right(worst)

Fig. 2 is the representation of the flow of control of the mechanism.

In the illustrated example(See *Fig. 2*) s_1 has the first preference as u_5. Since it was not already allocated to any other service provider, $s_1 - u_5$ becomes our first allocated service provider and user pair.

Now in the second iteration, similarly $s_2 - u_1$ becomes our second allocated pair. The third allocated pair following the same fashion is $s_3 - u_3$. Now in the fourth iteration, service provider s_4 has its first preference as $u_1$which is already allocated to s_2.

The second preference of s_4 is u_4 which is available and thus it is allocated making $s_4 - u_4$ the fourth allocated service provider and user pair.

In the final iteration we see that s_5 's first four preferences u_3, u_5, u_1, u_4 are already allocated to s_1, s_2, s_3 and s_4 and hence it is only with u_2. Thus the final service provider and user pair becomes $s_5 - u_2$.

The allocated service provider and user pairs are shown in *Fig. 3*.

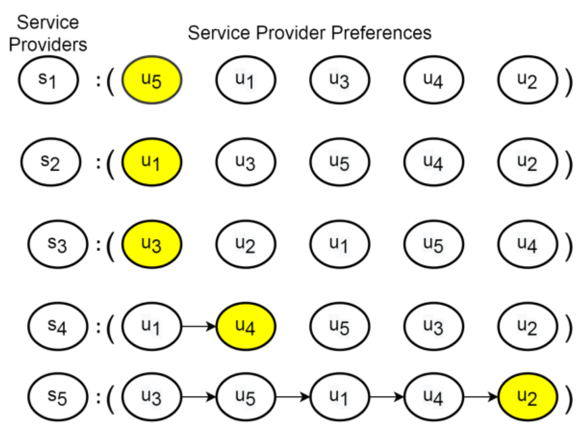

Figure 2. SFRA mechanism on service providers' preference matrix.

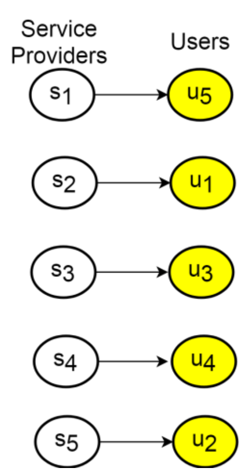

Figure 3. The allocated service provider and user pair after the execution of the SFRA mechanism.

We have already obtained a ranking of allocation of the users from running the SFRA mechanism on the original preference matrix. The ranking of users u_1 to u_5 in terms of who gets allocated earlier is 2, 5, 3, 4, 1 respectively. Now we see that coalition will be beneficial if u_5 is replaced as u_{53} as $rank(u_5) < rank(u_3)$ Hence, u_3 becomes the losing/supporting side and u_5 becomes the winner i.e the contributing player to the coalition. Thus a coalition u_{53} is formed as shown in *Fig. 4*. Consequently, all indexes where u_5 is there is changed to u_{53}. Whereas, all indexes where u_3 is there is replaced with a dummy variable u_d. Now, since u_d is still taking up a position in the preference list, we shift the dummy users to the end of each row so that the preference matrix can be updated. This updation is shown in *Fig. 5*. This is done to better the position of other users as well in the preference matrix without changing the n x n input matrix type.

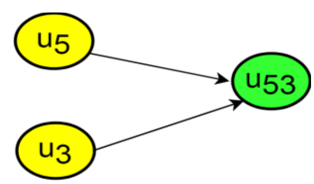

Figure 4. Two users u_5 and u_3 want to form a coalition.

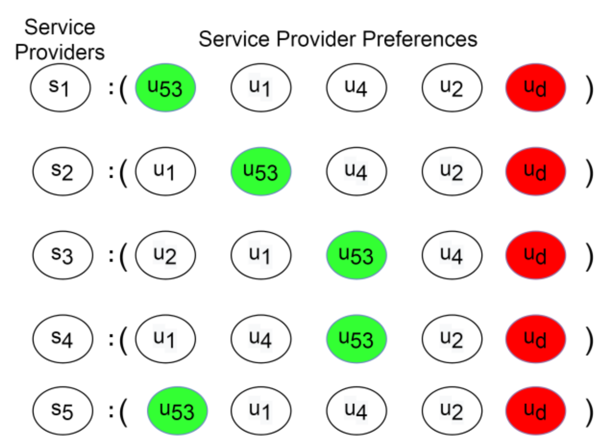

Figure 5. The updated preference matrix using CFRA.

Now, we run the SFRA mechanism on the updated matrix as shown in *Fig. 6*. In the illustrated example s_1 has the first preference as u_{53}. Since it was not already allocated to any other service provider, $s_1 - u_{53}$ becomes our first allocated service provider and user pair. Now in the second iteration, similarly $s_2 - u_1$ becomes our second allocated pair. The third allocated pair following the same fashion is $s_3 - u_2$. Now in the fourth iteration, service provider s_4 has its first preference as $u_1$which is already allocated to s_2. The second preference of s_4 is u_4 which is available and thus it is allocated making $s_4 - u_4$ the fourth allocated service provider and user pair.

In the final iteration, we see that s_5 's first four preferences u_{53}, u_1, u_4, u_2 are already allocated to

s_1, s_2, s_3 and s_4 and hence it is only with u_d. Thus the final service provider and user pair becomes $s_5 - u_d$.

The allocated service-provider pairs are shown in *Fig 7*. The pairs with dummy users will however be discarded in the final solution as getting a dummy user is of no value to the system providers. Thus, again a one-to-one matching is obtained by using our proposed SFRA mechanism. Note that even if multiple users form a coalition, they still resemble a single user in the matching.

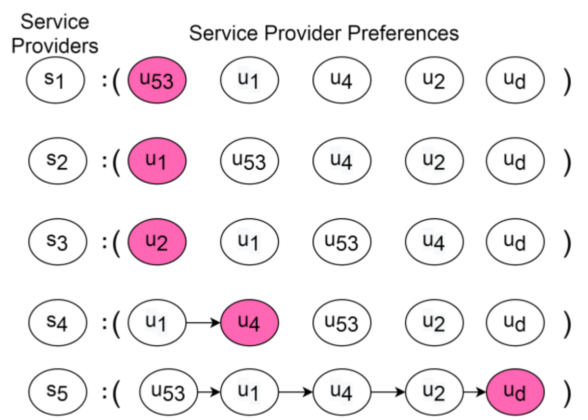

Figure 6. The CFRA mechanism: The execution of SFRA on the coalition matrix.

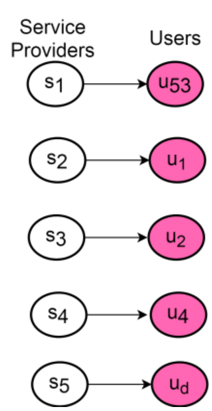

Figure 7. The allocated service provider and user pair after the CFRA mechanism.

D. Time Complexity of SFRA mechanism

Our proposed SFRA has a worst-case time complexity of $O(n^3)$, where n is the number of service providers or users in the system. The SFRA mechanism takes $O(n^2)$ to traverse the n x n matrix and another $O(n)$ to perform a linear search inside each traversal to check if the user is already allocated or not. Thus, the total complexity stands at $O(n^3)$ for our proposed approach.

E. Interesting Properties of the SFRA mechanism

Our proposed SFRA mechanism has some interesting properties:

Proposition 1. The algorithm SFRA is stable.

Proof. A matching is stable if there does not exist a pair for $(s_1 \rightarrow u_1)$, $(s_2 \rightarrow u_2)$ where $s_1, s_2 \in S$ and $u_1, u_2 \in U$ such that either s_1 prefers its current match to s_2 or s_2 prefers its current match to s_1. Here that does not happen as any $s \in S$ gets it's most preferred allocated user and would not swap with any other service provider. Hence, this results in a stable matching.

Proposition 2. The algorithm SFRA is Pareto-Optimal.

Proof. An outcome $o^*$is Pareto-Optimal if there is no other outcome that Pareto-dominates it.

Sometimes, one outcome o is at least good for every other agent as another outcome o', and there is some agent who strictly prefers o to o'. Consequently, it can be said that the situation of Pareto-Optimality indicates that no party can be made better off without making any other party worse off. Here the service provider who gets allocated first gets its top available pick and hence for the successive service providers they may have to choose a lower-ranked user even if it is the topmost pick for it at that point. Hence, one party has to sacrifice which leads to the conclusion that our SFRA mechanism is Pareto-Optimal.

Proposition 3. The algorithm SFRA returns a perfect matching.

Proof. Let us assume that the matching set A is not perfect. Then there must be at least one unallocated service provider s. The service provider may have expressed an interest to the user, as it has a total ordering of preference list. Now, each user also wants to get allocated to a service provider. Therefore all n users are allocated, so there must be n service providers. This contradicts our assumption that s was unallocated.

Proposition 4. The algorithm SFRA is not core allocation.

Proof. An allocation is said to have the core property if no coalition can improve upon it. The core is the set of all feasible allocations with the core property. But in this paper, we see in CFRA (*see Algorithm 2*), the formation of coalitions is improving the allocation of non-coalition members as well. Thus we can say that even if SFRA is stable but is not the core allocation.

F. Social Welfare

In a setting where users are allocated through some matching mechanism, there is always social welfare associated with it. This welfare can be measured using a social welfare function which is the benefit of each user present in the system. In our system model, each service provider has a list of preferences over all the other users and each user has a goal to get allocated as early as possible.

Upon executing the proposed SFRA mechanism, we obtain service provider and user matched pairs as the solution to the matching problem. Now let each user's rank be allotted according to the position at which it was allocated through SFRA. For example, the user who was allocated first will get rank 1, the user allocated second will get rank 2 and so on. The lower the rank is better is the allocation. This rank is denoted by $\mathfrak{R}_{u_i}$ for the i^{th} user. Thus each user has a rank value of $\mathfrak{R}_{u_i}$ ranging from $1, 2, \ldots, n$.

Now in a cooperative setting, users formed a coalition and then participated as a unit in the system. This coalition led to an overall improvement of all the users in a system whether they participated in the coalition or not. The utility for each user after the execution of CFRA is measured in comparison with the ranks obtained after the execution of the SFRA mechanism. Let the rank obtained after the CFRA mechanism of each user be denoted by $\mathfrak{R}_{u_i}{}^{*}$.

The individual utility of the user in such a scenario is:

$$\mho_{u_i} = \mathfrak{R}_{u_i} - \mathfrak{R}_{u_i}{}^{*} \tag{1}$$

Now, the total welfare for the users (denoted by $\mathbb{U}$) in the system becomes:

$$\mathbb{U} = \sum_{i=1}^{n} \mho_{u_i} = \sum_{i=1}^{n} (\mathfrak{R}_{u_i} - \mathfrak{R}_{u_i}{}^{*})$$

$$= n.\mathfrak{R}_{u_i} - \sum_{i=1}^{n} \mathfrak{R}_{u_i}{}^{*} \tag{2}$$

The average utility for each user in the system (denoted by $\mathbb{U}_{avg}$) in the system becomes:

$$\mathbb{U}_{avg} = \left(\frac{1}{n}\right) n.\mathfrak{R}_{u_i} - \sum_{i=1}^{n} \mathfrak{R}_{u_i}{}^{*} = \mathfrak{R}_{u_i} - \left(\frac{1}{n}\right) \sum_{i=1}^{n} \mathfrak{R}_{u_i}{}^{*} \tag{3}$$

5. EXPERIMENTAL RESULTS

In this section, we have presented some simulations based on our SFRA mechanism. To show the performance of our approach, we test it on both the scenarios i.e. with and without coalition. The preference matrix is generated randomly for all the experiments. Through these simulations, we study the improvements in the system when users decide to form a cooperative group. The experiment is done using Java.

A. *Simulation set-up*

The simulation has been performed on an n x n preference matrix where n users are to be matched to n service providers. For simulation purposes, the preference list of each service provider is generated randomly.

We have set a limit of service provider allocation to be a minimum of 70% consequently meaning coalition allowance to be of 30% to ensure fair allocation on the part of service providers. Here, rank indicates the position at which the user was allocated. A lower rank signifies better allocation i.e. the first allocated user has rank 1 and so on. Based on this setup, the following simulations are performed.

B. *Performance*

We compare the rank of the allocated users generated via SFRA and CFRA respectively and simulate for two scenarios with and without coalition i.e. Algorithm 1 and Algorithm 2(see *Fig. 8).* The ranks are allotted starting from 1 to n. This numbering is the same for all the service providers in the system.

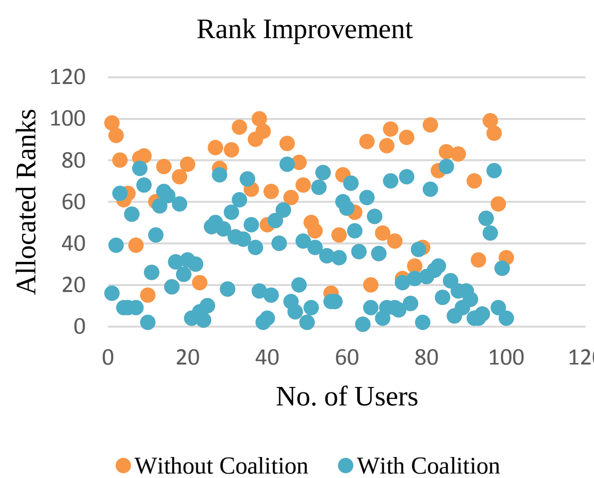

Figure 8. The graph shows the rank improvement of users in both settings.

We see in *Fig. 8,* that the cloudy region of the orange bubbles, which represent allocation with coalition is closer to the x-axis denoting the improved rank of the users. Whereas, the majority of the blue bubbles are scattered much higher denoting a difference between the allotted ranks. It is evident that our SFRA mechanism in a cooperative environment leads to the betterment of the users.

The SFRA mechanism when simulated on a coalition matrix, leads to the reassignment of users as compared to the SFRA mechanism without a coalition. The number of reassignments after coalition for users who did not take part in any coalition but still were reassigned can be studied in *Fig. 9* where n = number of users and n = 10, 20 , 30,…, 100.

Furthermore, we observe the coalitions formed also lead to the improvement of the users who did not participate in the coalition.

The utility of an individual is the number of jumps in the ranks i.e. difference between the value of the previous rank (via SFRA) and newly allocated rank (via CFRA).

In *Fig. 10*, we see a significant increase in the utility of users in terms of the number of ranks a user jumped to get reallocated to its new rank. The orange line denotes the utility improvement without coalition and the blue line denotes the utility with coalition. It is seen that the utility improvement for the members participating in the coalition is almost always more than with coalition.

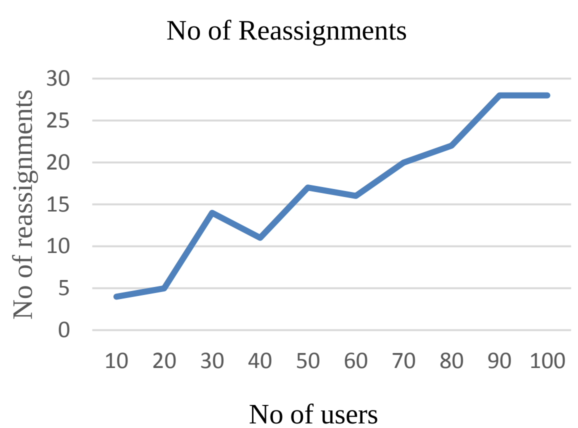

Figure 9. The graph shows the number of reassignments for the users who did not take part in any coalition

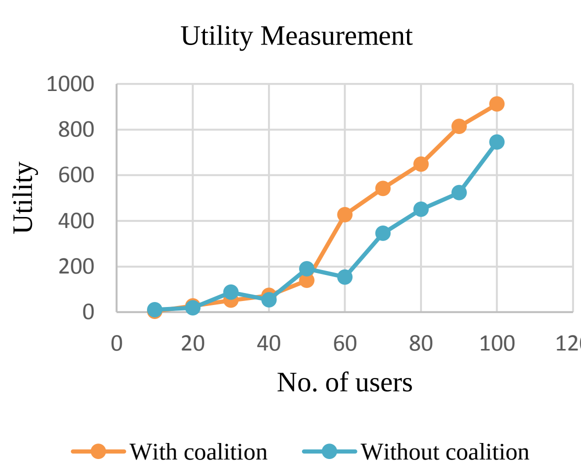

Figure 10. The graph shows the improved utility of users in both settings.

In *Fig. 11*, we see the average utility with and without coalition for n users taken at a time where n = 10, 20, 30, …, 100. In almost all scenarios the utilities of both settings are increasing linearly.

In *Fig. 12* the total utility of the system is shown. After n = 40, the graph increases almost linearly.

Thus these simulations present our approach in both the settings i.e. it can be modified to work in a cooperative environment as well. The overall one-to-one matching is not hampered due to the allocation of dummy users. The proposed approach is observed to work well with the coalition and can be tweaked to be applied in various other settings as well.

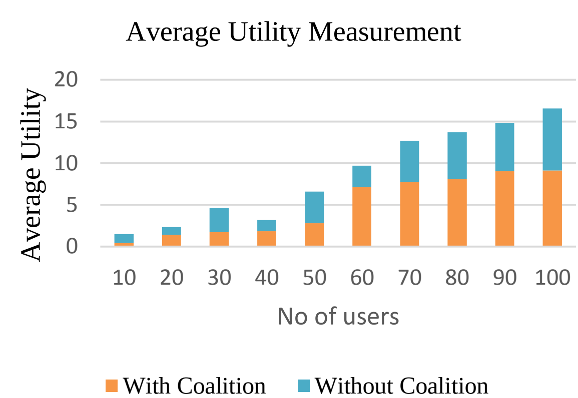

Figure 11. The graph shows the average utility of users in both settings.

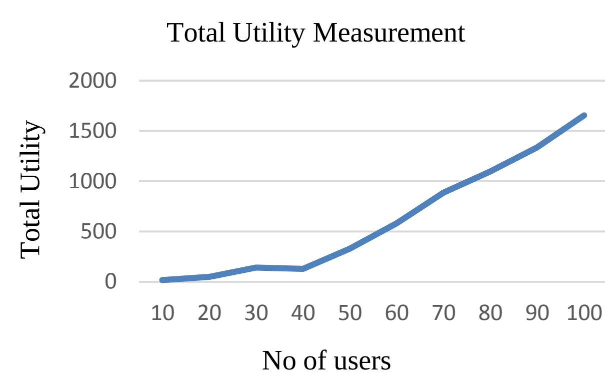

Figure 12. The graph shows the total utility of users in both settings.

6. Conclusion and Future Work

The discussion reveals that an economic environment can be created for buying and selling of resources in a fog computing environment.

Our work showed that the overall welfare of the system increases in a cooperative environment especially for the players who did not take part in any coalition. In this paper, we have considered a setting that did not involve any money i.e. allocation of resources was done free of cost. Thus in our future work, we aim to implement our approach for a monetary setting where resource allocation is done using money.

7. REFERENCES

[1] G. Eason, S. P. Singh, A. Nayyar, R. Kumar and A. Sharma, "Fog computing: from architecture to edge computing and big data processing," The Journal of Supercomputing, vol. 75, p. 2070–2105, 2019.

[2] F. Computing and others, "Fog computing and the internet of things: Extend the cloud to where the things are," in Technical Report, Cisco Systems, 2016.

[3] Z. Mahmood and M. Ramachandran, "Fog Computing: Concepts, Principles and Related Paradigms," in Fog Computing, Springer, 2018, p. 3–21.

[4] Bandyopadhyay, S. Mukhopadhyay and U. Ganguly, "Allocating resources in cloud computing when users have strict preferences," in 2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2016.

[5] Anglano, M. Canonico, P. Castagno, M. Guazzone and M. Sereno, "A game-theoretic approach to coalition formation in fog provider federations," 2018 Third International Conference on Fog and Mobile Edge Computing (FMEC), Barcelona, 2018, pp. 123-130, doi: 10.1109/FMEC.2018.8364054.

[6] Abdulkadiroglu and T. Sönmez, "Matching markets: Theory and practice," Advances in Economics and Econometrics, vol. 1, p. 3–47, 2013.

[7] E. Roth and M. Sotomayor, "Two-sided matching," Handbook of game theory with economic applications, vol. 1, p. 485–541, 1992.

[8] G. Demange and D. Gale, "The strategy structure of two-sided matching markets," Econometrica: Journal of the Econometric Society, p. 873–888, 1985.

[9] T. Quint, "On one-sided versus two-sided matching games," Games and Economic Behavior, vol. 16, p. 124–134, 1996.

[10] R. Ramesh, S. Sridhar, V. Manoharan and others, "AUPCR Maximizing Matchings: Towards a Pragmatic Notion of Optimality for One-Sided Preference Matchings," arXiv preprint arXiv:1711.09564, 2017.

[11] F. Chiti, R. Fantacci, F. Paganelli and B. Picano, "Virtual Functions Placement With Time Constraints in Fog Computing: A Matching Theory Perspective," IEEE Transactions on Network and Service Management, vol. 16, p. 980–989, 2019.

[12] Jia, H. Hu, Y. Zeng, T. Xu and Y. Yang, "Double-matching resource allocation strategy in fog computing networks based on cost efficiency," Journal of Communications and Networks, vol. 20, p. 237–246, 2018.

[13] Bandyopadhyay, S. Mukhopadhyay and U. Ganguly, "On free of cost service distribution in cloud computing," in 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2017.

[14] S. F. Abedin, M. G. R. Alam, S. A. Kazmi, N. H. Tran, D. Niyato and C. S. Hong, "Resource allocation for ultra-reliable and enhanced mobile broadband IoT applications in fog network," IEEE Transactions on Communications, vol. 67, p. 489–502, 2018.

[15] V. Krishna, Auction theory, Academic press, 2009.

[16] L. Zhang, Z. Li and C. Wu, "Dynamic resource provisioning in cloud computing: A randomized auction approach," in IEEE INFOCOM 2014-IEEE Conference on Computer Communications, 2014.

[17] W. Shi, C. Wu and Z. Li, "RSMOA: A revenue and social welfare maximizing online auction for dynamic cloud resource provisioning," in 2014 IEEE 22nd International Symposium of Quality of Service (IWQoS), 2014.

[18] N. Kumar and S. Saxena, "A preference-based resource allocation in cloud computing systems," Procedia computer science, vol. 57, p. 104–111, 2015.

[19] Kiani and N. Ansari, "Toward hierarchical mobile edge computing: An auction-based profit maximization approach," IEEE Internet of Things Journal, vol. 4, p. 2082–2091, 2017.

[20] S. Prasad, M. Arumaithurai, D. Koll and X. Fu, "Raera: A robust auctioning approach for edge resource allocation," in Proceedings of the Workshop on Mobile Edge Communications, 2017.

[21] N. C. Luong, Z. Xiong, P. Wang and D. Niyato, "Optimal auction for edge computing resource management in mobile blockchain networks: A deep learning approach," in 2018 IEEE International Conference on Communications (ICC), 2018.

[22] Pekeč and M. H. Rothkopf, "Combinatorial auction design," Management Science, vol. 49, p. 1485–1503, 2003.

[23] L. Fawcett, M. Broadbent and N. Race, "Combinatorial auction-based resource allocation in the fog," in 2016 Fifth European Workshop on Software-Defined Networks (EWSDN), 2016.

[24] J. Wang, A. Liu, T. Yan and Z. Zeng, "A resource allocation model based on double-sided combinational auctions for transparent computing," Peer-to-Peer networking and Applications, vol. 11, p. 679–696, 2018.

[25] Filos-Ratsikas, S. K. S. Frederiksen and J. Zhang, "Social welfare in one-sided matchings: Random priority and beyond," in International Symposium on Algorithmic Game Theory, 2014.

[26] G. Christodoulou, A. Filos-Ratsikas, S. K. S. Frederiksen, P. W. Goldberg, J. Zhang and J. Zhang, "Social welfare in one-sided matching mechanisms," in International conference on autonomous agents and multiagent systems, 2016.

[27] Bhalgat, D. Chakrabarty and S. Khanna, "Social welfare in one-sided matching markets without money," in Approximation, randomization, and combinatorial optimization. Algorithms and techniques, Springer, 2011, p. 87–98.

[28] W. Bolt and A. F. Tieman, "Social welfare and cost recovery in two-sided markets," Review of Network Economics, vol. 5, 2006.

[29] J. Yamamoto and K. Sycara, "A stable and efficient buyer coalition formation scheme for e-marketplaces," in Proceedings of the fifth international conference on Autonomous agents, 2001.

[30] R. Branzei, D. Dimitrov and S. Tijs, Models in cooperative game theory, vol. 556, Springer Science & Business Media, 2008.

[31] M. Guazzone, C. Anglano and M. Sereno, "A game-theoretic approach to coalition formation in green cloud federations," in 2014 14th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, 2014.

[32] M. Sharaf and T. El-Ghazawi, "Preference-based and Homogeneous Coalition Formation in Fog Computing," in 2019 IEEE Conference on Standards for Communications and Networking (CSCN), 2019.

[33] Z. Zhou, P. Liu, J. Feng, Y. Zhang, S. Mumtaz and J. Rodriguez, "Computation resource allocation and task assignment optimization in vehicular fog computing: A contract-matching approach," IEEE Transactions on Vehicular Technology, vol. 68, p. 3113–3125, 2019.

[34] S. F. Abedin, M. G. R. Alam, S. A. Kazmi, N. H. Tran, D. Niyato and C. S. Hong, "Resource allocation for ultra-reliable and enhanced mobile broadband IoT applications in fog network," IEEE Transactions on Communications, vol. 67, p. 489–502, 2018.

[35] Z. Ennya, M. Y. Hadi and A. Abouaomar, "Computing Tasks Distribution in Fog Computing: Coalition Game Model," in 2018 6th International Conference on Wireless Networks and Mobile Communications (WINCOM), 2018.

Prarena Shroff is a final year student of Computer Science and Engineering at Amity University Kolkata. Her research interests include Quantum Computing, Algorithmic Game Theory, Fog Computing and Mechanism Design.

Anjan Bandyopadhyay is an Assistant Professor of Computer Science and Engineering at Amity University, Kolkata and also a part-time Research Scholar of the Department of Computer Science and Engineering (CSE) at the National Institute of Technology (NIT), India. He has completed his M.Tech in Information Security from the Department of Information Technology, NIT Durgapur (2010-12). He is broadly interested in Algorithmic Mechanism Design and its Applications (Game Theory). He has published many conference and journal papers in esteemed conferences and publications. His current research interests include Cloud Computing and Fog Computing.