The CSI Journal on
Computer Science and Engineering
Vol. 15, No. 1, 2017
Pages 1-14
Regular Paper

Data Center Selection Based on Cloud Customers' Preferences of Quality Attributes for Federated Clouds

Maryam Pourreza[1] Hamid Haghshenas[1] Jafar Habibi[1]

[1]Department of Computer Engineering, Sharif University of Technology, Tehran, Iran

Abstract

Evolution of cloud computing has introduced a new paradigm called Federated Clouds. Using this concept, even small cloud providers can collaborate with other cloud providers and offer a much huger pool of resources with a much higher quality. Finding an optimized solution for the data center selection problem in a federated environment can help all cloud providers to have an increase in their customers' satisfaction. Therefore, in this paper we investigate the data center selection problem for federated clouds with the aggregated architecture in order to enhance the satisfaction of all cloud providers' customers in this environment. So, we first propose an optimization problem which considers all cloud customers' quality requirements along with their own priorities. We also show that this problem is NP-hard and inspect the complexity of different variations of the proposed problem. Furthermore, five algorithms are introduced for solving the proposed problem. Evaluation of these algorithms reveals that the proposed Ant Colony System algorithm provides the highest satisfaction for the customers in a reasonable amount of time. Moreover, the overall customers' utilities obtained by using this algorithm is at least 1.232 times higher than those obtained in previous approaches in this context.

Keywords: Data Center Selection, Federated Clouds, Aggregated Architecture, Cloud Computing.

1. Introduction

Cloud computing is a new computational paradigm which has evolved rapidly during the recent decades, thanks to the growth of the Internet. This paradigm has helped many organizations and companies to provide services with no concern about the upfront costs and also infrastructural management issues. The resources provided in this paradigm [20], not only can be leased and released expeditiously, but also can be configured and accessed in an on-demand fashion and without any further interaction with the service provider. Consequently, the appealing features [34] of cloud computing such as reduced operational and maintenance costs, has encouraged many companies to change their business models in order to get benefits from this paradigm.

What is important to note about cloud computing is that in cloud customers' view, providers offer services from an unbounded pool of resources. However, each cloud provider can actually run out of resources, if its customers' demands exceed the maximum capacity. Therefore, the cloud provider will not be able to provide any additional resources for their customers. Consequently, any further services will not be provided with the previous quality and hence, there is a probability for the violation of service level agreements (SLAs).

Cloud federation is a solution for the mentioned problem which has been emerged within the recent years. Using this concept, a group of cloud providers can get together and offer a much bigger pool of resources with a better quality [18]. As a result, the services provided in a federated environment [30] are more reliable, scalable and also cost effective.

For every cloud provider, a crucial decision is how to assign customers' requests to the available data centers. This key decision can highly affect the overall quality received by the cloud customer. Therefore, numerous researchers have investigated various approaches for the data center selection problem. However, all of the previous works have focused on a single or at most two quality attributes to be optimized

during the selection of data centers. But an important question arises that what if cloud customers have different priorities for the quality attributes. Thus, optimizing a specific quality attribute (e.g. performance) for all the customers will not be a huge advantage.

In this paper, we aim to maximize cloud customers' satisfaction via considering all of their quality attributes along with their own priorities in the selection of data centers. Our goal is to investigate this problem in a federated environment so that the users of all cloud providers in the federation can receive the benefits provided. Therefore, in the next section related works are first introduced. Then in section 3, the federated architecture for solving the problem is explained in detail. Section 4 also describes the proposed data center selection problem defined for maximizing users' satisfaction and then in the section 5 the solutions for this problem are proposed. Finally, section 6 presents the evaluation of the solutions and then the conclusion is given in section 7.

2. Related Work

There have been numerous papers in recent years investigating different approaches for the data center selection problem. Among the primitive approaches presented for this problem, Limbani et al. [16] propose a solution based on the proximity according to the network latency. In this approach, the nearest data center is selected and if more than one data center exists in a region, a random data center is chosen. Ahmed [17] also presents an enhanced version of the Limbani et al.'s solution for the data center selection. Since in the proximity based selection of data centers, a high load is imposed on the nearest data center, in the solution presented by Ahmed the load is forwarded to the second nearest data center. In this solution also a random data center is selected when there are several candidates for the solution. On the other hand, Sharma et al. [26] present a round-robin approach for the data center selection problem. However, this solution has major drawbacks such as high response time and cost.

Some of the researchers working on the data center selection problem focus on cost as an important quality attribute and try to optimize this metric during the selection. For example, Chudasmsa et al. [2] present a solution for the data center selection problem which selects the most cost effective data center. Like Limabani's solution, in this paper also if more than one candidate data center exists, then one data center is selected randomly. In another paper also, Limbani et al. [19] propose another alternative approach in which this random selection is replaced by a cost effective selection.

Performance is another quality attribute which is in focus in some of the data center selection papers. For example, Xu et al. [32] propose a framework for the data center selection for cloud services. In this framework, the goal is to have a balanced performance in the whole system so that near and far cloud customers can receive fair performance in the solution provided for the data center selection. In another paper, Mishra et al. [21] enhance the round robin approach based on the data centers' processing times. Therefore, data centers' speeds are used as their priorities and then the round robin approach based on these priorities are used for the data center selection problem. Gil et al. [9] also use predictions based on neuro-fuzzy inference systems for the data center selection problem. In their approach, the future load of the data centers are predicted and then a data center with the lowest load is selected. Using this approach, the performance is enhanced.

Performance is the main concern in some other data center selection papers. For example, Patel et al. [24] present an approach in which a data center with the least number of customers is chosen in a region instead of selecting a random candidate data center. Using this technique, they aim to optimize response time in the selection of data centers. Performance is also the main concern in the Yan et al.'s research [33]. In this paper, they present two policies for the data center selection problem including steady state policy and also the network state policy. These approaches are based on probability so an algorithm for computing these probabilities is also presented.

Some researchers also focus on a combination of two quality attributes for the selection of data centers. For example, Gupta et al. [12] present an enhanced version of the closet data center selection policy. In this paper, a latency matrix is utilized along with a metric indicating the virtual machines cost per hour. Hence, the overall processing time and also the cost are enhanced using this technique. In another research, Jaikar et al. [15] propose a matrix based selection technique in order to reduce the cost as well as to increase the performance. In this research, two matrices including the distance versus cost matrix and also the performance versus availability matrix are used in the approach for selecting appropriate data centers. It is worth mentioning that unlike previous works, this paper is in the context of federated clouds with a centralized management system. Nandwani et al. [22] also present a weight-based approach for the selection of data centers. This research which is another enhanced version of the proximity based data center selection, focuses on the performance and also the cost as two important quality attributes.

As mentioned in the previous section, all of these papers focus on a single or at most two quality attributes for solving the data center selection problem. However, cloud customers are concerned about many quality attributes with different priorities. Hence, optimizing the assignments for a single quality attribute does not suit every customer in the cloud environment. Moreover, none of the previous works had a focus on federated clouds except Jaikar et al. which concentrated on a centralized management system. However, in this paper we focus on the aggregated architecture for the federated clouds and in the next section, the benefits provided using this architecture are also discussed.

3. Federated Clouds Architecture

As described in previous sections, a single cloud can run out of resources when the users' demands exceed the maximum capacity of the cloud provider. In this situation the provider is unable to give further services with previous quality. Hence, the SLAs between the cloud provider and the users may even be violated. Federated environment gives the cloud providers the opportunity to provide services with a higher quality. In this environment, having few resources will not be a major obstacle to giving services and hence, small cloud

providers can also join the federation.

Several architectures are introduced for federated clouds in recent years. Moreno-Vozmediano et al. have classified [29] these architectures based on the coupling levels of cloud instances. Based on this classification, four types of architectures can be considered for cloud federation, including Bursting Architecture, Broker Architecture, Aggregated Architecture, and also Multitier Architecture. In the first category, cloud customers should decide themselves which provider they should use. However, in the second category of architectures, a broker is responsible for the assignment of each customer to an appropriate cloud provider. In the aggregated architecture on the other hand, cloud customers are aware of just a single primary cloud provider while the provider itself has some contractions with other cloud providers in the federation. Therefore, a huge amount of virtual resources will be offered to the customers. Furthermore, the Multitier Architecture is a special architecture for organizations that have multiple cloud infrastructures distributed geographically.

According to the categories described by Moreno-Vozmediano et al., broker architecture and also the aggregated architecture are the most widely used federated architectures. However, there are numerous concerns about the broker architecture such as the overhead of the single broker and also the risk of failure of this component. It should be mentioned that for cloud providers also the broker architecture is not favorable, since the broker in this architecture does not discriminate between different cloud providers. However, each cloud provider prefers to be selected more in the federation when it has enough resources. Therefore, aggregated cloud federation can provide numerous benefits for both cloud customers and also cloud providers. Hence, in this paper we focus on the aggregated architecture for the federated clouds.

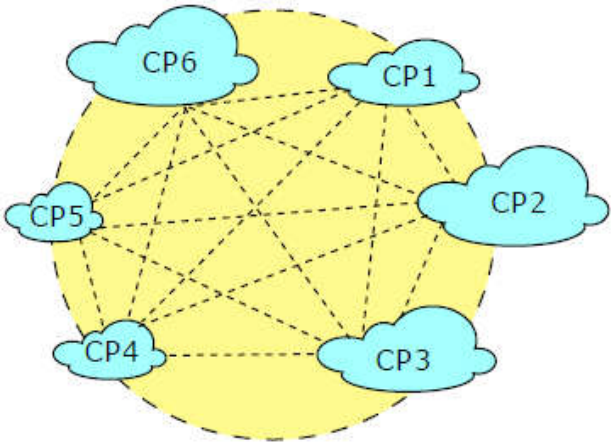

Figure 1: Aggregated Architecture for Federated Clouds

Fig. 1 shows the aggregated architecture for federated clouds. In this architecture, cloud providers $CP_1, \dots, CP_n$ have peer to peer contacts with each other and unlike broker architecture, no central component is responsible for choosing cloud providers. In aggregated architecture, each cloud provider can forward its customers' requests to other cloud providers in the federation in order to provide services with higher quality. In this architecture also each cloud provider is responsible for some part of the federation. Consequently, all components responsible for satisfying federation needs in different cloud providers, make the whole federation infrastructure. Therefore, in Fig. 1 the dashed circle represents the federation infrastructure.

4. Data Center Selection

According to the architecture described in previous section, cloud providers can send their customers' requests to any of the other cloud providers in the federation and of course based on their negotiations. However, the question then arises that how and to which providers should cloud customers' requests be redirected. Therefore, in the following subsections we try to solve this question by defining the Data Center Selection problem based on customers' preferences of quality attributes and then we propose different solutions for the mentioned problem.

4.1. Problem Definition

In this section we define the data center selection problem with the goal mentioned before. The solution of this problem leads to higher satisfaction of the overall service received by all cloud providers' customers. Therefore, we first model our problem according to the following integer linear programming (ILP).

$$\max \sum_{i=1}^{s}\sum_{j=1}^{n} u_{ij}x_{ij}$$

$$s.t. \sum_{i=1}^{s} r_{il}x_{ij} \le C_{jl}, \forall j \in \{1, \dots, n\}, \forall l \in \{1, \dots, d\} \quad (1)$$

$$\sum_{j=1}^{n} x_{ij} \le 1, \forall i \in \{1, \dots, s\}$$

$$x_{ij} \in \{0,1\}, \forall j \in \{1, \dots, n\}, \forall i \in \{1, \dots, s\}$$

According to this ILP, we have s cloud customers as well as n data centers. The goal is to gain maximum overall utility for the assignments of the customers to the data centers available. Each data center in this model has d type of resources (e.g. CPU and memory) and also a specific capacity for each type of resource. Hence, C_{jl} indicates the capacity of resource type l in the jth data center.

Cloud customers on the other hand have also a specific requirement for each type of resource. This requirement is shown by r_{ij} which indicates requirement of ith customer for the lth resource type. It is worth mentioning that sum of the requirements of the customers assigned to each data center should never exceed its total capacity for each type of resource. This important issue is formalized in the second line of the ILP. Furthermore, there is another assumption in the problem definition which is presented in the third line. By this line, we assume that each cloud customer will be assigned to at most one data center.

As mentioned before, in the proposed model of the data center selection problem we aim to maximise customers' satisfaction which is denoted as u_{ij}. This parameter is also defined in the equation below. Based on this equation, t quality attributes can be considered for the assignments of cloud customers to data centers. In each of these

assignments, the utility that the i th customer receives, depends on the customer's priority of different quality attributes and also the quality delivered by the corresponding data center. Therefore, w_{ik} in the equation denotes customer i's priority of the kth quality attribute. Moreover, qijk in this equation shows the quality delivered by jth data center to the i th cloud customer for the kth quality attribute. It is important to note that sum of the cloud customer's priorities for different quality attributes should be 1, as shown in second equation below.

$$u_{ij} = \sum_{k=1}^{t} w_{ik} q_{jik} \quad (2)$$

$$\sum_{k=1}^{t} w_{ik} = 1, \forall i \in \{1, \dots, s\} \quad (3)$$

According to the utility definition described, each cloud providers' data center can provide different qualities to different customers based on several parameters. For example, one parameter that affects response time delivered to the customer is the distance between the customer and the cloud provider's data center. Hence, different cloud customers from different parts of the world receive various qualities from each cloud provider.

It is worth mentioning that each cloud provider in a federated environment with aggregated architecture has information about the capacities of other cloud providers' data centers. Furthermore, for finding the qualities delivered to customers, several papers in previous years have investigated quality of services prediction in cloud environments. Zheng et al. [37], [36] use past service experiences of cloud customers to predict the qualities delivered to similar customers. They introduce two approaches for ranking cloud services based on the qualities delivered to a customer. Ding el al. [4] also propose a similar prediction for solving cloud service evaluation problems. Using the techniques proposed in these papers, each cloud provider can predict the quality provided by each data center in the federation before solving the data center selection problem.

4.2. Problem Analysis

In this subsection, we aim to analyze the proposed definition of data center selection problem in (1). This definition is an extension of a variation of Knapsack problem which is called Multiple Multi-dimensional Knapsack. Therefore, in this subsection we first introduce the definitions of both Multiple Knapsack and Multi-dimensional Knapsack problems in IV.1 and IV.2.

Definition IV.1. The 0/1 multiple knapsack problem [1] consists of B knapsack and S items. Each item has a profit p_i and also a size s_i. On the other hand, each knapsack has a specific capacity c_j . We aim to maximize our profit by selecting items in such a way that no capacity restriction is violated.

Definition IV.2. The 0/1 multi-dimensional knapsack problem [13] consists of n items and also one knapsack with m constraints. Each knapsack constraint has a capacity b_i and each item has a profit c_j and also weights A_i^j associated with each knapsack constraint. Like previous definition, we aim to maximize our profit while keeping capacity restrictions unviolated.

Both of these problems are NP-hard [1], [13] and hence we can conclude that our problem which is an extension to a combination of these problems is also NP-hard. Furthermore, it has been proved that multiple knapsack problem does not have any FPTAS solution [1] which consequently means that our data center selection problem also does not have any FPTAS solution.

A few researchers have focused on solving the multiple multi-dimensional knapsack problem. Romaine [25] introduced a Tabu Search algorithm for solving aircraft load-scheduling problem which was modeled as a multiple multidimensional knapsack with packing constraints. In this problem, weight and volume are the only dimensions of the knapsacks. Song et al. [28] also used multiple multi-dimensional knapsack to model centralized spectrum allocation problem in radio cognitive networks. In this paper, first an exact solution based on branch-and-bound technique is introduced for the problem. Since the exact method is very time consuming in the sense of the algorithm's time complexity, a heuristic algorithm is also presented.

For analyzing the data center selection problem defined in (1), we have reduced some constraints to analyze the effects on its time complexity. These variations of the problem are as follows:

1) **Assumptions:** Only one data center with one type of resource exists.

Solution: Since the original problem turns into the knapsack problem which is NP-hard [30], all the approximation solutions for the knapsack problem can also be applied to the proposed data center selection problem with this assumption.

2) **Assumptions:** Only one type of resource with the same capacity for all data centers exists. The utility for all customers is also one.

Solution: The original problem turns into the Binpacking problem which is NP-hard and all approximations for the mentioned problem can also be applied to this version of the data center selection problem as well.

3) **Assumptions:** For each cloud customer, the requirements for all types of resources are equal.

Solution: Based on (1), each customer is assigned to maximum one data center. Therefore, the requirements of a cloud customer is supposed to be fulfilled with maximum one data center. As a result, minimum capacity of a data center will be its bottleneck and it can be considered as the data center's capacity. Consequently, the original problem in this case turns into the Generalized Assignment Problem [1] which is APX-hard. Hence, every approximation for the so called problem can be applied to this version of the data center selection problem. It is worth mentioning that the best approximation for the Generalized Assignment Problem is 2 [1].

4) **Assumptions:** The requirements for all cloud customers and all types of resources are equal.

Solution: Like previous solution, minimum capacity of different resource types for each data center can be considered as the data center's capacity C'_j . Since the requirements are assumed equal, we can divide C'_j by this

equal requirement $C"_j = {C'_j}/{r}$.

Then we can create a weighted bipartite graph for the problem. For each cloud customer a vertex is considered in the first part of the graph. In the second part of the graph, for each data center some nodes are considered. The number of nodes for each data center in the graph are based on their calculated capacity $C"_j$. Moreover, weights in this graph are the utilities of each cloud customer in the assignment to each data center.

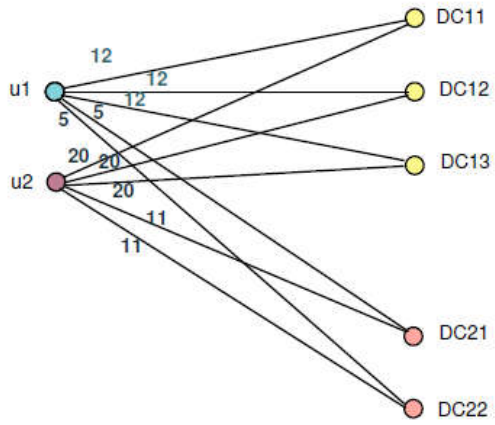

Figure 2: Example of the weighted bipartite graph created for solving data center selection problem with assumption 4

Fig. 2 is an example of the created graph with this assumption. In this figure, there are two cloud customers and two data centers. The first data center has $C"_1 = 3$ and hence three nodes are created for this data center. The second one has $C"_2 = 2$ and two nodes are considered for this data center in the graph. Furthermore, the weights in the example graph are the utilities of each assignment.

Having this graph, a maximum weighted matching can be found as a solution for this version of data center selection problem. While maximum weighted matching problem has a polynomial time algorithm, data center selection problem with this assumption can also have a polynomial time algorithm [23].

Based on the analysis presented in this subsection, several simplifications of the data center selection problem (1) are also NP-hard. Therefore, in the next subsection we present some heuristics for solving the problem.

5. Proposed Solutions

In the previous section, it was shown that the proposed data center selection problem (1) is NP-hard. Therefore, in this section five algorithms are presented for finding near optimal solutions for the mentioned problem. These algorithms are a simple greedy algorithm, an Ant Colony System algorithm, a Tabu Search algorithm, a Genetic algorithm, and also a Simulated Annealing algorithm. The last four algorithms are in the category of meta-heuristics and are presented in detail in the following subsections.

5.1. Simple Greedy Algorithm

In the data center selection problem, data centers' capacities along with the utilities are crucial in the assignments of cloud customers to data centers. To be more precise, in a heavy loaded data center, customers with low requirements are preferable to those with high requirements. Moreover, customers with low requirements from important resource types (the ones with lower capacity) are also preferable. As a results, in the proposed simple greedy algorithm, cloud customers' usage of the data centers' capacities are considered as well as the utility of each assignment.

Algorithm 1 presents the proposed simple greedy algorithm. In this algorithm, all possible assignments are sorted based on the equation 4. According to this equation, we want to have not only a high utility, but also a high remaining capacity of important resource types after each assignment. After sorting B_{ij}s computed in accordance to the specified equation, assignments are then selected in turn. Since choosing assignments with high B_{ij} s may violate the maximum capacity of data centers, some assignments must be ignored if they violate the capacity of the corresponding data center.

$$B_{ij} = u_{ij}^2 + \min\{C_{jl} - r_{il}\}, \forall l \in \{1, \dots, k\} \quad (4)$$

As a result of running this algorithm, after one traversal on the sorted list, some cloud customers may remain unassigned. Hence, the sorted list of B_{ij}s of these customers will be merged with the B_{ij}s computed for the second batch of the customers' requests in the next call of the algorithm. Consequently, these customers will have another chance for being assigned to data centers.

Algorithm 1 Simple Greedy Algorithm for Data Center Selection

function GreedyAlg(Data Centers, Cloud Customers, Previous Assignments)
  $finalSolution \leftarrow \emptyset$
  Compute B_{ij} for all the assignments of customers to data centers
  Sort B_{ij}s
  Merge the sorted list with the previous assignments
  for all B_{ij} **do**
    if customer i is unassigned **then**
      if Assignment of the customer i to data center j is possible **then**
        Add this assignment to the final solution
  return Final solution as well as B_{ij} s of unassigned customers

5.2. Ant Colony System Algorithm

Ant colony algorithm is a nature-inspired meta-heuristic algorithm which simulates real world ants' behaviors for finding their food. In this algorithm [7], a group of ants seek to find the solution iteratively and the search is based on the pheromones left on previous solutions. The strength of these pheromones informs the ants about the possibility of getting near to the optimal solution.

A crucial decision for designing an effective ant colony

algorithm is the definition of the problem's solution components. In our data center selection problem, each assignment of cloud customers to cloud providers' data centers is considered as a problem component. Therefore, ants iteratively build components until they reach an acceptable solution. They also leave pheromones on the solution components as feedback of the obtained solution. This feedback is used for the next generation of solutions.

As mentioned before, building solution is based on the pheromone left on previous solutions' components. More precisely, selection of a component in the path for making the final solution is based on a probabilistic approach. This probability is according to the benefit earned in using the respective component which is a function of the optimization problem's objective and also the pheromone left from previous traversals.

After reaching the solutions, the pheromones of the solutions' components are updated. Ant colony algorithms have several categories that each of them has a different approach for these updates. In the proposed algorithm for the data center selection problem, ant colony system is considered for updating the pheromones.

What makes ant colony system different from other categories of ant colony algorithms [8] is the local and also global updates of the pheromones. In the local update, the pheromone is reduced in order to make diversification possible and hence, push the ants towards the global solution. The global update is also based on the quality of the solution which is presented in equation 5. In this equation, $\Delta\tau_{ij}$ is calculated based on equation 6.

$$\tau_{ij} = \alpha\tau_{ij} + (1-\alpha)\Delta\tau_{ij} \quad (5)$$

$$\Delta\tau_{ij} = \begin{cases} U_{GBS} & if\ (i,j) \in GBS \\ 0 & otherwise \end{cases} \quad (6)$$

In the above equations, α is a number between zero and one which defines how the pheromone is reduced. According to the equation 6, if a component is part of the global best solution, then the pheromone is increased based on the utility of the associated solution.

Algorithm 2 is inspired from the ant colony algorithm given by Dorigo et al. [5] for the Traveling Salesman problem. As mentioned before, we consider each assignment of customers to cloud providers' data centers as a solution component for the data center selection problem. This assignment is based on the equation 7.

$$S = \begin{cases} argmax_{i \in FC_k(j)}\left\{\tau_{ij} * \eta_{ij}^{\beta}\right\} & if\ q \le q_0 \\ P & otherwise \end{cases} \quad (7)$$

$FC_k(j)$ in the above equation, is the set of candidate customers which can be assigned to the data center j according to its remaining capacity in the kth ant's path for finding the solution. Furthermore, q_0 is a number between zero and one which is an input for the algorithm showing exploitation versus exploration. Lower amounts of q_0 enables more exploration in the solution space. In the first case of the equation 7, the best user is selected from the candidate list based on the pheromone left on the component and also the utility provided for the assignment. However, in the second case of the equation, which is the exploration phase, user selection is based on a probability presented in equation 8.

$$p_k(i,j) = \begin{cases} \frac{\tau_{ij} \times \eta_{ij}^{\beta}}{\sum_{i \in FC_k(j)} \tau_{ij} \times \eta_{ij}^{\beta}} & if\ i \in FC_k(j) \\ 0 & otherwise \end{cases} \quad (8)$$

It is worth mentioning that η_{ij} in all the equations, shows how desirable is the assignment as a component in the whole solution. We have defined this in the equation 9 based on the utility of the assignment in the data center selection problem's objective function. Moreover, β in all the equations is an input for the algorithm that shows importance of the utility gained from selecting a component versus the pheromone traces on that component.

$$\eta_{ij} = u_{ij} \quad (9)$$

Algorithm 2 Ant Colony Algorithm for Data Center Selection Problem

1: **function** AntColonyAlg(Data Centers, Cloud Customers, Number of Ants, Maximum Number of Iterations, β, α, q_0)
2: Set an initial pheromone τ_0 for every possible assignment
3: $iterationNumber \leftarrow 0$
4: $BestSolution \leftarrow \emptyset$
5: **while** $iterationNumber \neq maxIterations$ **do**
6: **for all** $ants \in antsSet$ **do**
7: $antSolution_i \leftarrow \emptyset$
8: $antCustomer_i \leftarrow 1$
9: $iterationNumber \leftarrow iterationNumber + 1$
10: $antsList \leftarrow antsSet$
11: **while** $antsList \neq \emptyset$ **do**
12: Select an ant in $antsList$ randomly
13: **while** $FC_k(ant_j) = \emptyset$ **do**
14: $antCustomer_j \leftarrow antCustomer_j + 1$
15: **if** $antCustomer_j > customersLength$ **then**
16: Remove ant_j from $antsList$
17: Select a customer from $FC_k(ant_j)$ based on equation 7
18: Update ant_j customers list
19: **for all** $ants \in antsSet$ **do**
20: **if** Utility of the ant is greater than the best utility found so far **then**
21: Update best solution with ant's solution
22: $iterationNumber \leftarrow 0$
23: Compute $\Delta\tau$
24: **for all** Data centers **do**
25: **for all** Customers **do**
26: Update the component's pheromone based on equation 5
27: **return** Best solution found

5.3. Genetic Algorithm

Genetic algorithm is a meta-heuristic based on the process of natural selection [3]. In this algorithm, an initial population of chromosomes are created as a set of possible solutions. Then based on two crossover and mutation functions and their probabilities, new generations are constructed. In each

generation of the new population, fitness of the chromosomes in current population are first calculated. Finding this fitness helps to choose the best chromosomes as parents of the next generation. In this algorithm, a solution with a higher fitness is selected with more probability as a parent for the newborns.

Algorithm 3 is the proposed genetic algorithm for the data center selection problem. What is important in this algorithm is the way solution formulated as a chromosome in each population. In the proposed genetic algorithm, we use permutation encoding for the chromosome presentations. In this method, each chromosome is a string of numbers with a length equal to the number of cloud customers. Each number in this string shows the data center that the corresponding customer is assigned to. Moreover, the population size in the proposed algorithm is equal to the number of cloud customers.

Another important decision in a genetic algorithm is how to select chromosomes as parents of the newborns for the next generation. There are several methods for selecting chromosomes among which Reulette Wheel Selection and Rank Selection are two well-known methods. In the Reulette Wheel method, when there is a huge difference between utilities of different chromosomes in a population, chromosomes with relatively low utility have a very small chance to be selected. However, in the ranking selection, all chromosomes are first sorted based on the utility function and then each chromosome is selected based on its ranking among other chromosomes. As a result, ranking selection has a slower convergence according to its selection process.

Algorithm 3 Genetic Algorithm for Data Center Selection Problem

```
1: function GeneticAlg(Data Centers, Cloud Customers,
Population Size, Maximum Number of Iterations,
Crossover Probability, Mutation Probability)
2:    Generate first population
3:    iterationNumber ← 0
4:    while iterationNumber ≠ maximumIteration do
5:       Calculate utilities of chromosomes in the population
6:       while New population is not complete do
7:          Select two parents based on the Ranking Method
8:          Generate a newborn with the best genes of his
parents (based on the crossover probability)
9:          Change some assignments in the newborn
chromosome (based on the mutation probability)
10:         Add the newborn in the new population
11:      currentPopulation ← newPopulation
12:      iterationNumber ← iterationNumber + 1
13:   return Best solution in the current population
```

Mutation and crossover methods in a genetic algorithm have also a high impact on the effectiveness of the algorithm. In the proposed genetic algorithm for the data center selection problem, after selecting the parents from the current population, the best genes are chosen from them according to the utility of each corresponding assignment. Mutations in this algorithm are also based on a mutation probability which helps escaping local optimums. In this phase, some genes from the newborn are mutated based on the mutation probability. This mutation changes the assignment of a cloud customer to another feasible data center.

Crossover probability and mutation probability are two of the inputs of the proposed algorithm. If the crossover probability is equal to zero, the newborn will be the exact copy of its parent and in fact, there will be no crossover. As mentioned before, the goal in the crossover phase is to keep desirable genes in the newborn. On the other hand, if the mutation probability is zero, the newborn will not be mutated at all, while mutation probability of one mutates the whole chromosome of the newborn. It should be noted that high mutation probabilities also turn the algorithm into a random search. Hence, we will not use high mutation probabilities for the proposed genetic algorithm.

5.4. Simulated Annealing Algorithm

Simulated annealing algorithm [14] mimics the process of heating and cooling of metals for changing their physical structures. Therefore, high temperatures help the algorithm to navigate different areas of solution space in the exploration phase and consequently to avoid local optimums. However, low temperatures help the algorithm to converge to the best solution in a local area.

Algorithm 4 presents the proposed simulated annealing algorithm for the data center selection problem. In each iteration of this algorithm, first a neighbor of the current solution is chosen. Neighbor of the current solution is a solution with the same assignments of cloud customers to cloud providers' data centers except that one assignment is changed. If the chosen neighbor has a better utility, then the current solution will be replaced by this neighbor. Else, the replacement will be based on the current temperature of the system and also the difference between the current solution's utility and the neighbor's utility. As mentioned earlier, higher temperatures permits the algorithm to explore more in the solution space. Therefore, selection of a neighbor with low utility will be based on the probability presented in equation 10.

$$\exp\left(\frac{-(solutionEnergy - neighborEnergy)}{temperature}\right) \quad (10)$$

At each iteration of the algorithm, the temperature is also reduced based on the equation 11. The algorithm also stops searching when the temperature is low enough.

$$temperature = (1 - 0.003) \times temperature \quad (11)$$

5.5. Tabu Search Algorithm

Tabu search algorithm [10-11] is another meta-heuristic which avoids local optimums and tries to navigate different areas of the solution space. This algorithm consists of two phases called Intensification and Diversification. In the intensification phase, movements in the solution space are with small distances from the current solution and the goal is to get a solution with higher utility. However, in the diversification phase, the goal is to search different areas of the solution space without considering the utility function.

Algorithm 4 Simulated Annealing Algorithm for Data Center Selection Problem

```
1: function SimulatedAnnealingAlg(Data Centers, Cloud Customers, Temperature)
2:     Generate an initial solution randomly
3:     while temerature > 1 do
4:         Change one assignment in the current solution randomly to find a neighbor
5:         if neighbor's utility is greater than current utility then
6:             Update the current solution with the neighbor
7:         else
8:             Accept the neighbor based on equation 10
9:         Update the temperature based on equation 11
10:        iterationNumber ← iterationNumber + 1
11:    return Best solution found
```

In the tabu search algorithm, previous visited solutions are kept so that the algorithm does not consider them repetitively in its process. Our proposed tabu search algorithm is inspired by the team selection algorithm for prediction tasks [6] given by Fazli et al. This algorithm uses Probabilistic Move Selection Rule for choosing neighbors with relatively lower utilities. Using this method, the algorithm's final solution will converge to the global optimum solution.

Algorithm 5 presents the proposed tabu search algorithm for the data center selection problem. In this algorithm, if no neighbor with a better utility is found for the current solution, then the current solution is replaced by the best neighbor with the probability of P which is an input of the algorithm. This probability is in fact the probability of escaping the local maximum in the solution space and based on the research done by Wu et al., we use 0.1 for this input [31]. It should be noted that neighbors in this algorithm have the same definition with the neighbors in the presented simulated annealing algorithm.

6. Evaluation

In this section, the evaluation of the proposed solutions for the data center selection with the aim of increasing cloud customers' satisfaction is presented. First, a brief description of the program setup and the dataset used for evaluation is given. Then, the results are presented along with their comparison with the optimal solution and also previous works in data center selection.

6.1. Experiment Setup and Data

For evaluation of the presented algorithms in previous section, we have used a dataset created from a distributed data center from BitBrains [27]. This dataset is created in Delft University and contains information from 1250 virtual machines of a BitBrains's data center. This information include a time stamp showing the time when the customer's request is arrived, the customer's requirement and usage of the CPU, the customer's requirement and usage of the memory, throughput of the disk read, throughput of the disk write, network received throughput and the network transmitted throughput.

Algorithm 5 Tabu Search Algorithm for Data Center Selection Problem

```
1: function TabuSearchAlg(Data Centers, Cloud Customers, Maximum Number of Iterations, Selection Probability)
2:     Generate a feasible solution randomly
3:     iterationNumber ← 0
4:     tabuList ← ∅
5:     bestSolution ← ∅
6:     while iterationNumber ≠ maximumIteration do
7:         Generate neighbors of the current solution which are not in the tabu list
8:         Find the best neighbor based on the total utility
9:         if Best neighbor's utility is higher than the current utility then
10:            Update the current solution with the best neighbor
11:        else
12:            Accept best neighbor based on the selection probablity
13:        Update tabu list
14:        if Current solution is better than the best solution found so far then
15:            Update the best solution
16:            iterationNumber ← 0
17:        else
18:            iterationNumber ← iterationNumber + 1
19:    return Best solution found
```

For the presented data center selection problem, we need to know the amount of customers' requirements in order to find a near optimal assignment. For generating near realistic requirement values as the inputs for the algorithms, first a statistical analysis is performed on the BitBrains data about customers' requirements. Therefore, distributions of customers' requirements for the CPU and also the memory are investigated for the analysis. According to this analysis, both CPU requirements and memory requirements have a uniform distribution. Furthermore, the correlation between these two requirements is also calculated. Based on this calculation, customers' CPU requirements and their memory requirements have a correlation of 0.6830478.

Another input for the data center selection problem is the qualities delivered to the customers. For generating near realistic values for this type of input, we have used WSDREAM dataset [38], [35]. This dataset contains near 2 million invocation records of real world web services. This dataset provides information about qualities delivered to 339 users from 5825 web services. Qualities delivered to the customers in this dataset include response time and the throughput. Therefore, two matrices are provided in this dataset indicating the quality delivered to each user from each web service. We have also analyzed the distribution of these two qualities and also the correlation between them. According to this analysis, both of these quality metrics have

an exponential distribution. However, calculations show no correlation between these two qualities delivered to the customers.

After analyzing requirements and the quality attributes in the BitBrains and WSDREAM dataset, near realistic data were generated as the inputs of the algorithms. These inputs include cloud customers with different requirements for the CPU and memory and also different priorities for the response time and the throughput. Cloud providers' data centers are the other inputs for the problem which include information about the qualities they provide for the customers and also their capacities for the CPU and the memory.

Based on the analysis done on the CPU and the memory requirements, these two parameters have a correlation of 0.68. Therefore, for generating near realistic values for these two requirements for each cloud customer, we first generate a covariance matrix for the CPU and the memory data of the BitBrains dataset. For creating this matrix, first the initial matrix x containing these data is transformed to a matrix of deviation scores according to the equation 12. In this equation, n is the number of requirements' vectors of the customers in the BitBrains dataset.

$$d = x - I'xI(I'I)^{-1} = x - I'xI(\frac{1}{n}) \quad (12)$$

After creating the d matrix, using the equation x'x, a square matrix from the deviation sums of squares and cross products of the initial matrix is created. Then by dividing each element by n, the covariance matrix is generated. This matrix is used for the generation of requirements values for each cloud customer. Furthermore, each cloud customer has priorities for the quality attributes which were created randomly based on a uniform distribution.

For the other type of input for the data center selection problem, three data centers with the capacities of 3511199, 5851999, and 1170399 MHz for the CPU and also 20132659200, 33554432000, and 6710886400 KB for the memory were created. Moreover, the qualities delivered to the cloud customers by each data center were also created randomly based on the distribution discussed before.

The algorithms presented in previous section are coded in Java 8 and were run on an Intel(R) Core(TM) i5-2467M 1.6GHz CPU with the inputs described in this section. The results of these algorithms and their comparison with previous works are given in the following subsection.

6.2. Results

For comparing the algorithms, first the inputs of each algorithm are initialized. For example, the Tabu Search algorithm is run with the selection probability of 0.1 and the maximum iteration numbers of 20. The Genetic algorithm also has the maximum number of iterations of 120 and the crossover rate of 90 and also the mutation rate of 0.1. Moreover, the Simulated Annealing algorithm also has the initial temperature of 1700. The inputs of the Ant Colony System algorithm are also a maximum iteration number, beta, alpha, q0, and the initial pheromone which are set to 3, 2, 0.1, 0.4, and 1, respectively.

Fig. 3 shows the execution time of all the proposed algorithms along with the optimal algorithm in seconds. Since the optimal algorithm takes a huge amount of time to be executed for more number of cloud customers, Fig. 4 shows the execution time of all algorithms except the optimal algorithm for more number of cloud customers in seconds. It is worth mentioning that the execution time in this figure is the average execution time of running each algorithm four times.

It should be noted that the proposed algorithms are only used by cloud providers in a federated environment when there is a shortage in the resources. Moreover, based on our analysis on the number of requests in the BitBrains dataset, in each second there are only 16 customer requests. Therefore, the running times of the proposed algorithms shown in Fig. 4 are reasonable.

Fig. 5 also shows the error of each algorithm in finding the optimal utility compared to the optimal algorithm's utility. As mentioned before, since the optimal algorithm is very slow for larger numbers of cloud customers, this figure is shown only for few numbers of customers. However, Fig. 6 shows the utility obtained by each algorithm for larger numbers of cloud customers. Like Fig. 4, this figure also shows the average utility of four times running each algorithm.

Based on presented evaluations, the proposed Tabu Search algorithm, the Genetic algorithm, and also the Ant Colony System algorithm provide the highest amounts of utilities for the customers. However, the proposed Ant Colony System algorithm provides these high amounts of utilities in a much less amount of time compared to the Tabu Search and Genetic algorithms. Therefore, the Ant Colony System algorithm is the best algorithm in comparison with the other proposed algorithms from both the time and utility perspectives.

Next, we have compared the utility obtained by the proposed algorithms with the utility obtained in the weight based approach [22] for the data center selection and also the matrix based approach [15] presented in two recent and previous works. In the evaluation section of both of these approaches, the earth is divided into different continents and the data centers and the cloud customers are all placed in different continents of the earth. Then, the geographical region of the data centers and customers are used in the calculations for finding optimal assignments. Therefore, we also use the same strategy in order to compare our algorithms with theirs. Hence, each cloud customer is placed randomly in a region. The first two data centers are also placed in the US and the third one is placed in the Europe.

Fig. 7 shows the average utility obtained in four times running the weight based approach and the proposed Ant Colony System algorithm. According to this figure, although two quality attributes are considered for finding optimal assignments, the utilities obtained by the proposed algorithm in this paper are at least 1.232 times higher. Because, an important factor which shows the utility is the customers' priorities of the quality attributes that indicate their overall satisfaction.

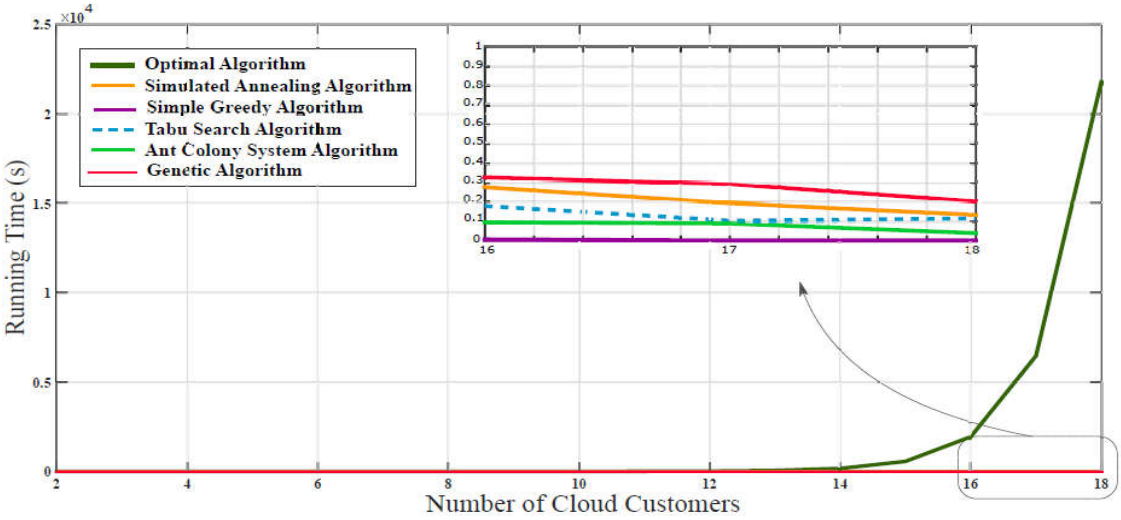

Figure 3: Running Time of All Proposed Algorithms Along with the Optimal Algorithm for Different Number of Cloud Customers

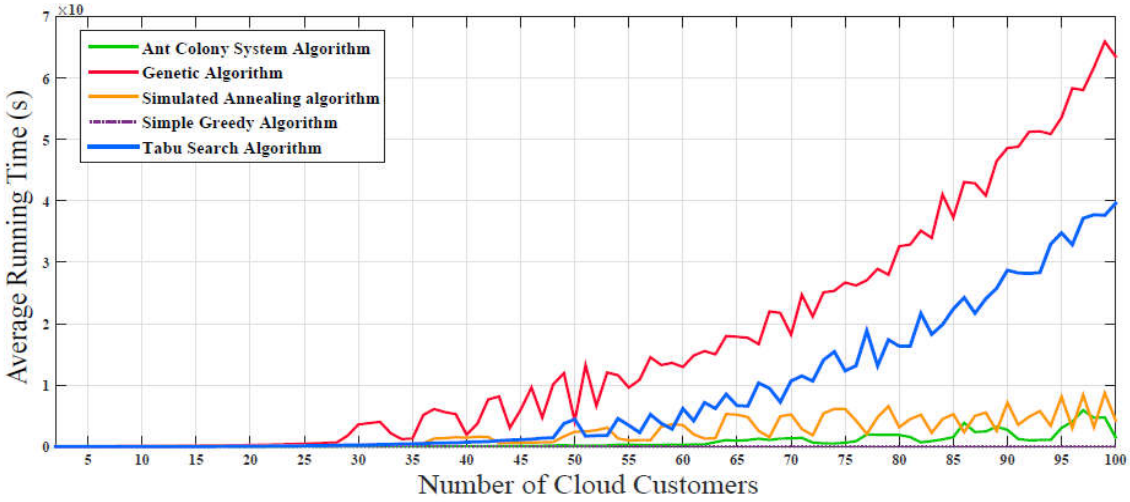

Figure 4: Average Running Time of All Proposed Algorithms for Different Number of Cloud Customers

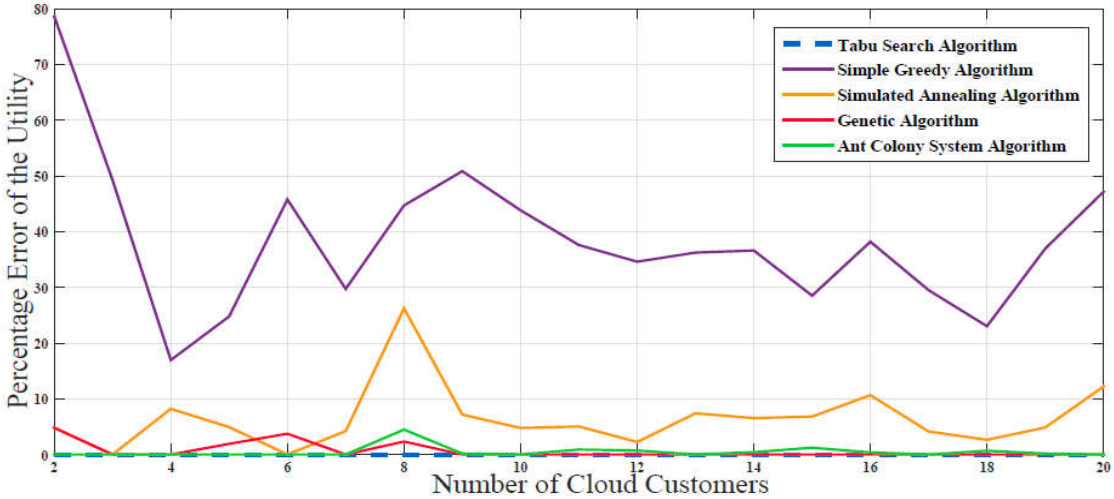

Figure 5: Percentage Error of the Utility of All Proposed Algorithms for Different Number of Cloud Customers

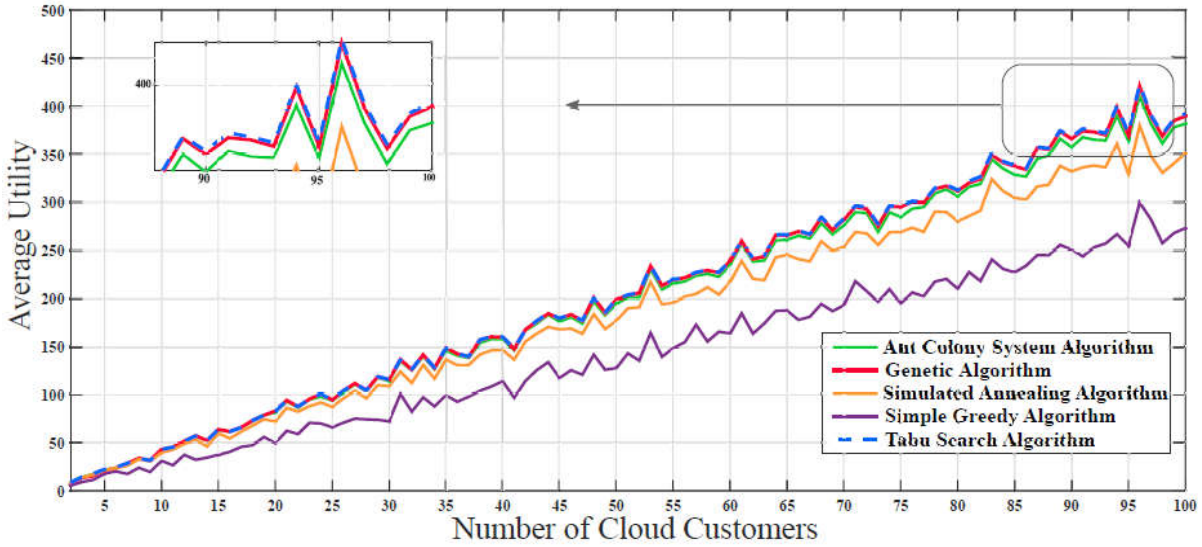

Figure 6: Average Utility of All Proposed Algorithms for Different Number of Cloud Customers

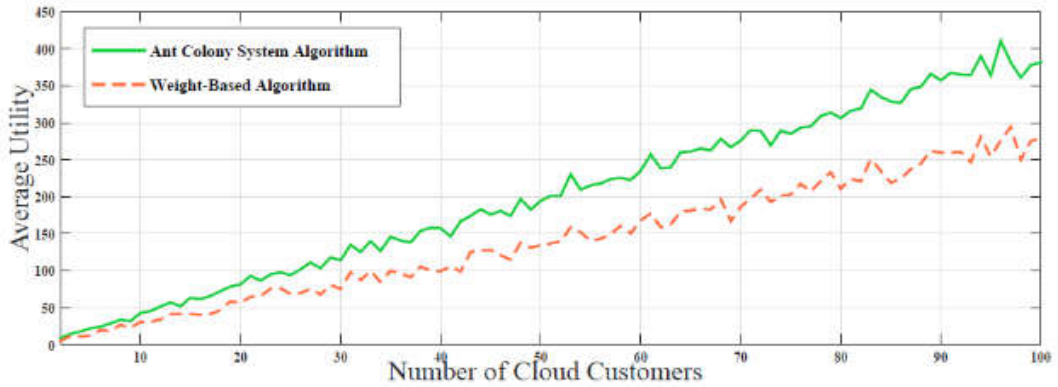

Figure 7: Average Utility of Ant Colony System Algorithm and Weight-based Algorithm for Different Number of Cloud Customers

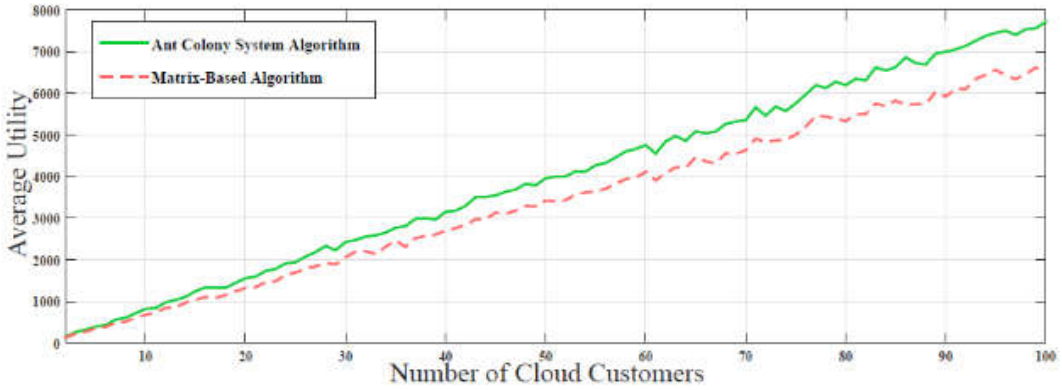

Figure 8: Average Utility of Ant Colony System Algorithm and Matrix-based Algorithm for Different Number of Cloud Customers

Fig. 8 also shows the average utility obtained in four times running the matrix based approach and the proposed Ant Colony System algorithm. Based on this figure, the utilities obtained in the proposed algorithm in this paper are at least 1.323 times higher than the matrix based approach. This utility shows the overall customers' satisfaction of the assignments. As mentioned before, the matrix based approach also does not consider customers' priorities which leads to lower utility for the customers. It should also be mentioned that for comparing our algorithms with the matrix based approach, we have considered three quality attributes including performance, cost, and the availability and their values are set according to the values in the matrix based approach paper. The cloud customers' priorities are also set randomly based on a uniform distribution for these quality attributes.

What is important to note is that both of the weight based approach and also the matrix based approach do not consider data centers' capacities of the resources. Unfortunately, this strategy is a double-edged sword, which can increase the speed of the algorithm, but also can be a serious problem in a federated environment consisting of small cloud providers with limited resources in their data centers. When data centers' capacities are ignored, there may come a situation that these capacities are violated which leads to a decrease in the qualities delivered to the customers and even worse, violation of the service level agreements.

Therefore, we changed the data centers' capacities and compared the algorithms again. So, first the data centers' capacities of the cloud providers are set to 585199, 351119, and 1170399 MHz for the CPU and also 3355443200, 2113265920, and 6710886400 KB for the memory. Using these data centers with the mentioned capacities, the weight based approach offers assignments that violate the capacities. However, the utilities obtained by the proposed algorithm in this paper are still higher than the weight based approach.

Next, the data centers' capacities of the cloud providers are changed to 234079, 231119, and 1170399 MHz for the CPU and also 134217728, 2013265920, and 6710886400 KB for the memory. In this run of the algorithms also the matrix based approach provides assignments that violate the data centers' capacities and still the overall utilities obtained by the proposed Ant Colony System algorithm in this paper are higher than the utility obtained from the matrix based approach.

7. Conclusion

In this paper, we have proposed the data center selection problem for federated clouds with the aggregated architecture. We first introduced the architecture and its benefits for the cloud providers and their customers. Then for increasing cloud customers' satisfaction, the data center selection problem is proposed based on the customers' preferences of quality attributes. It should be mentioned that previous works investigating the data center selection problem have focused on a single or two quality attributes in their assignments of cloud customers to the data centers. However, various cloud customers may have different priorities for the quality attributes. Furthermore, most of these papers also focus on a single cloud provider. Therefore, the integer programming introduced in this paper for the data center selection problem can highly increase the satisfaction of all cloud providers' customers in a federated environment.

The integer programming introduced in this paper is also shown to be NP-hard and different variations of this problem are also analyzed computationally. Moreover, five algorithms for finding near optimal solutions for the proposed problem are also presented. These algorithms include a simple greedy algorithm, an Ant Colony System algorithm, a Tabu Search algorithm, a Simulated Annealing algorithm, and also a Genetic algorithm. In the evaluation, we also show that the proposed Ant Colony System algorithm provides the highest satisfaction for the customers in a reasonable amount of time. This algorithm also provides at least 1.232 times higher overall utility for the customers when compared with previous approaches in this context. It should also be noted that unlike this paper, previous works do not consider data centers' capacities and hence there is a high risk of capacity violation in their provided solutions which may lead to a violation in service level agreements. This is very precarious in a federated environment which may contain small cloud providers with low capacities.

For future work, stochastic programming methods can also be utilized for considering uncertainties in the cloud customers' requirements. Moreover, since the problem is modeled as a variation of the multiple multi-dimensional knapsack problem, any approximation for this problem can help in finding optimal assignments of the cloud customers to the cloud providers' data centers. Furthermore, this problem can also be analyzed in other types of architectures for the federated clouds in the future works.

References

[1] C. Chekuri, and S. Khanna, "A polynomial time approximation scheme for the multiple knapsack problem," *SIAM Journal on Computing*, vol. 35, no. 3, pp. 713–728, 2005.

[2] D. Chudasama, N. Trivedi, and R. Sinha, "Cost effective selection of data center by proximity-based routing policy for service brokering in cloud environment," *International Journal of Computer Technology & Applications*, vol. 3, no. 6, pp. 2057–2059, 2012.

[3] L. Davis, Handbook of genetic algorithms, 1991.

[4] S. Ding, S. Yang, Y. Zhang, C. Liang, and C. Xia, "Combining qos prediction and customer satisfaction estimation to solve cloud service trustworthiness evaluation problems," *Knowledge-Based Systems*, vol. 56, pp. 216–225, 2014.

[5] M. Dorigo, and L. M. Gambardella, "Ant colony system: a cooperative learning approach to the traveling salesman problem," *IEEE Transactions on evolutionary computation*, vol. 1, no. 1, pp. 53–66, 1997.

[6] M. A. Fazli, A. Ghazimatin, J. Habibi, and H. Haghshenas, "Team selection for prediction tasks," *Journal*

of Combinatorial Optimization, vol. 31, no. 2, pp. 743–757, 2016.
[7] S. Fidanova, "Aco algorithm for mkp using various heuristic information," *Proc. International Conference on Numerical Methods, and Applications*, pp. 438–444, 2002.
[8] S. Fidanova. "Ant colony optimization and multiple knapsack problem," *Handbook of research on nature inspired computing for economics and management*, pp. 498–509, 2007.
[9] J.-M. Gil, J. H. Park, and Y.-S. Jeong, "Data center selection based on neuro-fuzzy inference systems in cloud computing environments," *The Journal of Supercomputing*, vol. 66, no. 3, pp. 1194–1214, 2013.
[10] F. Glover, "Tabu search-part i," *ORSA Journal on computing*, vol. 1, no. 3, pp. 190–206, 1989.
[11] F. Glover, "Tabu searchpart ii," *ORSA Journal on computing*, vol. 2, no. 1, pp. 4–32, 1990.
[12] S. Gupta, and I. Kashyap, "Cost effective service broker strategy in selection of data centers in cloud computing," *MR International Journal of Engineering & Technology*, vol. 6, no. 1, 2014.
[13] S. Hanafi, and A. Freville, "An efficient tabu search approach for the 0–1 multidimensional knapsack problem," *European Journal of Operational Research*, vol. 106, no. 2, pp. 659–675, 1998.
[14] C.-R. Hwang, "Simulated annealing: theory and applications," *Acta Applicandae Mathematicae*, vol. 12, no. 1, pp. 108–111, 1988.
[15] A. Jaikar, and S.-Y. Noh, "Cost and performance effective data center selection system for scientific federated cloud," *Peer-to-Peer Networking and Applications*, vol. 8, no. 5, pp. 896–902, 2015.
[16] D. Limbani, and B. Oza, "A proposed service broker policy for data center selection in cloud environment with implementation," *International Journal of Computer Technology and Applications*, vol. 3, no. 3, 2012.
[17] A. S. Ahmed, "Enhanced proximity-based routing policy for service brokering in cloud computing," *International Journal of Engineering Research and Applications*, vol. 2, no. 2, pp. 1453–1455, 2012.
[18] A. Celesti, F. Tusa, M. Villari, and A. Puliafito, "How to enhance cloud architectures to enable cross-federation," *Proc. IEEE 3rd International Conference on Cloud Computing*, pp. 337–345, 2010.
[19] D. Limbani, and B. Oza, "A proposed service broker strategy in cloudanalyst for cost-effective data center selection," *International Journal of Engineering Research and Applications*, vol. 2, no. 1, pp. 793–797, 2012.
[20] P. M. Mell, and T. Grance, "The nist definition of cloud computing," Technical report, Gaithersburg, MD, United States, 2011.
[21] R. K. Mishra, S. Kumar, and B. S. Naik, "Priority based round-robin service broker algorithm for cloud-analyst," *Proc. IEEE International Advance Computing Conference (IACC)*, pp. 878–881, 2014.
[22] S. Nandwani, M. Achhra, R. Shah, A. Tamrakar, K. Joshi, and S. Raksha, "Weight-based data center selection algorithm in cloud computing environment," *Artificial Intelligence and Evolutionary Computations in Engineering Systems*, pp. 515–525, 2016.
[23] C. H. Papadimitriou, and K. Steiglitz, *Combinatorial optimization: algorithms and complexity*, Courier Corporation, 1982.
[24] V. Patel, and N. Shah, "A proposed service broker policy for inter region data center selection in cloud environment," *International Journal of Engineering Research and Applications (LJERA)*, pp. 1699-1702, 2013.
[25] J. M. Romaine, "Solving the multidimensional multiple knapsack problem with packing constraints using tabu search," Technical report, DTIC Document, 1999.
[26] V. Sharma, R. Rathi, and S. K. Bola, "Round-robin data center selection in single region for service proximity service broker in cloudanalyst," *International Journal of Computers & Technology*, vol. 4, no. 2a1, pp. 254–260, 2013.
[27] S. Shen, V. van Beek, and A. Iosup, "Statistical characterization of business-critical workloads hosted in cloud datacenters," *Proc. 15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid)*, pp. 465–474. 2015.
[28] Y. Song, C. Zhang, and Y. Fang, "Multiple multidimensional knapsack problem and its applications in cognitive radio networks," *Proc. MILCOM*, pp. 1–7, 2008.
[29] R. Moreno-Vozmediano, R. S. Montero and I. M. Llorente, "IaaS Cloud Architecture: From Virtualized Datacenters to Federated Cloud Infrastructures,".*Computer*, vol. 45, no. 12, pp. 65-72, 2012.
[30] V. V Vazirani, Approximation algorithms, *Springer Science & Business Media*, 2013.
[31] Q. Wu, and J.-K. Hao, "An adaptive multistart tabu search approach to solve the maximum clique problem," *Journal of Combinatorial Optimization*, vol. 26, no. 1, pp. 86–108, 2013.
[32] H. Xu, and B. Li, "A general and practical datacenter selection framework for cloud services," *Proc. IEEE 5th International Conference on Cloud Computing (CLOUD)*, pp. 9–16. 2012.
[33] S. Yan, X. Wang, M. Razo, M. Tacca, and A. Fumagalli, "Data center selection: A probability based approach," *Proc. 16th International Conference on Transparent Optical Networks (ICTON)*, pp. 1–5, 2014.
[34] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud computing: stateof-the-art and research challenges," *Journal of internet services and applications*, vol. 1, no. 1, pp. 7–18, 2010.
[35] Y. Zhang, Z. Zheng, and M. R Lyu, "Exploring latent features for memory-based qos prediction in cloud computing," *Proc. 30th IEEE Symposium on Reliable Distributed Systems (SRDS)*, pp. 1–10, 2011.
[36] Z. Zheng, H. Ma, M. R Lyu, and I. King, "Collaborative web service qos prediction via neighborhood integrated matrix factorization," *IEEE Transactions on Services Computing*, vol. 6, no. 3, pp. 289–299, 2013.
[37] Z. Zheng, X. Wu, Y. Zhang, M. R Lyu, and J. Wang, "QoS ranking prediction for cloud services," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 6, pp. 1213–1222, 2013.
[38] Z. Zheng, Y. Zhang, and M. R Lyu, "Distributed qos evaluation for real-world web services," *Proc. IEEE International Conference on Web Services (ICWS)*, pp. 83–90, 2010.

Maryam Pourreza received her M.Sc. and B.Sc. degrees in Software Engineering from Sharif University of Technology in 2016 and 2014, respectively. Her current research areas of interest are cloud computing, cloud federation, and big data.
Email: pourreza@ce.sharif.edu

Hamid Haghshenas received his BSc and MSc degrees in software engineering from Sharif University of Technology, in 2010 and 2012 respectively. Currently, he is a PhD candidate in software engineering at Sharif University, focusing on cloud federation architectures. His research interests include cloud computing, cloud federation, software architecture and discrete mathematics.
Email: haghshenas@ce.sharif.edu

Jafar Habibi is a faculty member at the computer engineering department at Sharif University of Technology and the managing director of Electronic Computing Machines Services. He is supervisor of Sharif Robo-Cup Simulation Group. His research interests are mainly in the areas of computer engineering, simulation systems, MIS, DSS and evaluation of computer systems performance.
Email: jhabibi@sharif.edu

Paper Handling Data:

Submitted: 29.04.2018
Received in revised form:10.06.2018
Accepted: 15.06.2018
Corresponding author: Hamid Haghshenas
Computer Engineering Department, Sharif University of Technology, Tehran, Iran