Reducing Service Migrations in Fog Infrastructures by Optimizing Node Location

Ioanna Stypsanelli, Samir Medjiah, Balakrishna Prabhu

Reducing Service Migrations in Fog Infrastructures by Optimizing Node Location

Ioanna-Vasiliki Stypsanelli, Samir Medjiah, Balakrishna J. Prabhu
LAAS-CNRS, Université de Toulouse, CNRS, UPS, Toulouse, France
Email: {firstname.lastname}@laas.fr

Abstract

In order to ensure service continuity of connected cars moving inside a Fog Computing infrastructure under a Service Level Agreement, a service needs to migrate from a fog node to another. An approach is to always keep migrating the service towards the fog node that is the closest to the current position. However, frequent service migrations have a migration and network cost. Intuitively, the more migrations are triggered, the bigger this cost is. In this work we look into ways to reduce this cost by studying how to minimize the number of VM migrations triggered.

We introduce a general case in which we minimize a linear combination of the infrastructure cost and the number of service migrations given statistics on the routes taken by the vehicles. This problem can be represented as a bipartite graph where the minimization problem is an instance of the *Weighted Set Cover* problem.

For a special case of pair-wise mobility model in which the origin and destination of vehicles are in the coverage range of adjacent base stations, we first present a static offline ILP formulation of the migration minimization problem. For this simple case, we then propose two heuristics inspired by the greedy algorithm for the weighted set cover problem as polynomial approximations.

Keywords — fog computing, service migration, integer linear programming, weighted set cover, greedy heuristic

1 Introduction

Fog Computing is a new promising paradigm where computational resources are distributed closer to user premises and near the points of presence. Fog nodes are sometimes referred to as *cloudlets, micro data centers, microclouds* or they can as well reside in eNodeB base stations nodes in architectures we call by *FogRAN* or *Fog enabled cellular networks.*

This new distributed paradigm brings a number of advantages: low application latency, network traffic distribution, since processing takes place at the network edge in different locations by the Fog nodes, network traffic distribution increasing availability, reduced bandwidth cost etc.[14]

However, there are challenges that Fog computing needs to deal with. One challenge is service continuity for a user who moves in a Fog computing infrastructure. This user can be a connected car, drones, flying taxis etc.

In order to deal with service continuity, a service needs to migrate among these Fog nodes so that the user can stay closer to the Fog node from which it can be served better. However there is cost implied by the migrations.

First, we have a *migration cost* which is incurred when the service is moved from the previously serving node to the one serving the new user location [7, 11, 14]. Factors that affect the migration cost are the service size, the cost to initiate and release a service and the bandwidth consumed to migrate the service.

Second, the *transmission cost* which increases, when the user is moving far away from the original server which hosts the service, no service migration is triggered, and so the service is accessed from the original service node via the backhaul network, instead of one-hop link.

In our paper, we are focusing on the migration cost. Intuitively, the more migrations are triggered, the bigger the migration cost is. In this work we look into ways to reduce this cost by studying how to minimize the number of VM migrations triggered.

1.1 Contributions

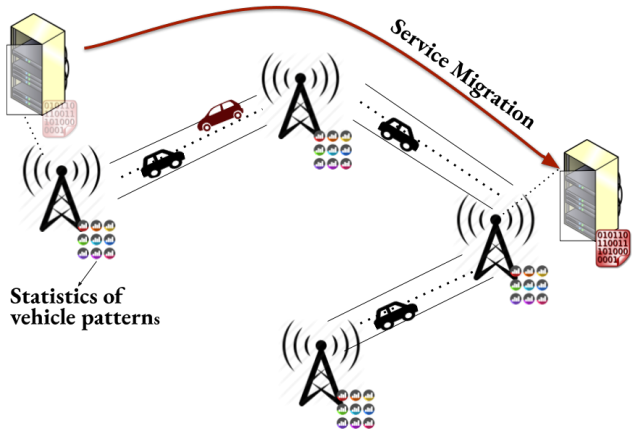

Figure 1: Service Migration based on statistics gathered on base stations

We take a user-centric approach where we focus on data objects (user sessions and related service data) that need to be migrated between Fog nodes. We do not concern ourselves with the technical details implementing the data objects, or whether the service is implemented using VMs, containers or similar.

It is not our concern whether the VM or container must be entirely migrated, or whether a VM or container is running for one or many users.

Base stations are connected to a number of Fog nodes which process the service requests. We do not consider base stations being Fog processing nodes, even though they may also provide services relevant to maintain the Fog running, such as working with handover events to trigger service migrations. Base stations also use their networking data to decide which Fog node to route their traffic to (or, more appropriately, decide where a data object should be spawned or migrated depending on its SLA). On the other hand, Fog nodes can be heterogeneous and have a different OS, computing capacity, number of servers. VMs, hardware etc.

In our model a car is connected using the cellular network to a number of base stations. Base stations maintain a list of candidates Fog nodes with allowed thresholds for SLA. Whenever a car moves to another base station, the data object may be migrated to another Fog node satisfying the SLA and minimizing the overall cost of service (Fog migrations, bandwidth, etc). Instead of deciding of the appropriate Fog node to migrate to when a handover happens, the Fog controller computes the optimal set in advance when the itinerary is shared.

Given a set of candidate locations for fog clouds, we study the problem of obtaining the optimal set of the fog nodes. The optimality criterion is a linear combination of the cost of infrastructure and the number of migrations. The cost of infrastructure increases with the number fog nodes. On the other hand, one would expect the number of migrations to decrease with more fog nodes. Therefore, there is trade-off between these two quantities which need to be balanced to get the optimal solution.

We shall assume that the statistics on the number of cars going from the coverage range of one base station to another is given. These statistics will be used to compute the total number of migrations.

While we present the problem for a scenario in which cars can travel along any path in the network, we shall present results for special case in which cars travel only one hop between coverage range of one base station to another. The traffic statistics can be represented by a traffic matrix whose (i, j)th entry is the traffic going from the coverage range of base station i to that of base station j.

1.2 Organization

This rest of the paper is organized as follows. In Section 2, we describe the general model, give some details on the well-known weighted set cover problem, and present an interger linear program (ILP) formulation of the problem. In Section 3, we present a heuristic to compute a feasible solution to this ILP. This heuristic is based on the greedy algorithm for the weighted set cover problem. In Section 4, we evaluate the performance of the heuristic and compare it with the standard greedy algorithms for the set cover and weighted set cover problems. Finally, we end the paper with conclusions and future work in Section 6

2 System Model

We are given as input a set $\mathcal{F}$ of potential fog nodes as well as a set $\mathcal{B}$ of base stations. Base station $i \in \mathcal{B}$ can route the traffic from a user to a subset of candidates fog nodes $F_i \subset \mathcal{F}$ which can host the service with respect to a allowed SLA. A fog node $j \in F_i$ is said to cover base station i when fog node j respects the SLA requirements of the user. In this, we say there is a link between fog node j and base station i. Here, we are assuming that all the users have the same SLA requirement.

The base stations and fog nodes can be represented by a bipartite graph $G = (\mathcal{B} \cup \mathcal{D}, \mathcal{E})$, where an edge between base station i and fog node j is in $\mathcal{E}$ if and only if j can guarantee the SLA requirement of users in the coverage range of i.

When the user changes position and moves out of the coverage range of base station i and in to the coverage range of base station k where the F_k and F_i do not have any fog nodes in common, a service migration is triggered. That means the user session is migrated from a fog node in F_i to a node in F_k with $F_i \cap F_k = \emptyset$. In another words, if $\{i, k\} \notin F_i$ a service migration is triggered.

Let us define a path in the road network by a set of ordered base stations. We shall assume we are given as input the traffic statistics of how many users move along each path.

As an example, consider the a network with four base stations $\{A, B, C, D\}$ and three fog nodes $1, 2, 3$. Here $F_A = \{1\}$, $F_B = \{1, 2\}$, $F_C = \{2, 3\}$ and $F_D = \{3\}$. There are only two subsets of $\mathcal{F}$ that can cover $\mathcal{B}$: $\{1, 3\}$ and $\{1, 2, 3\}$.

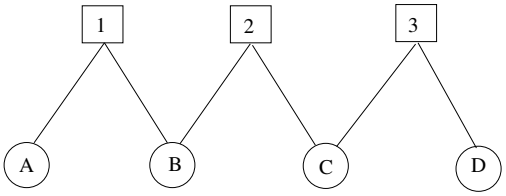

Figure 2: Example bipartite graph

Suppose there is only one path $\{A, B, C, D\}$ in the network with 100 users taking this path. Then, the number of migrations for $\{1, 3\}$ is 100 (one for each user migrating from fog node 1 to 3 when the user moves from base station B to base station C) whereas for $\{1, 2, 3\}$ this number is again 100 (assuming 2 is not used) or 200 (if we assume that A is connected to 1, B and C are connected to 2 and D is connected to

3). Assuming the infrastructure cost is linear in the number of fog nodes, then it is optimal in this case to operate fog nodes 1 and 3 only.

Now, suppose that in addition to the previous users there are 500 users that take the path $\{B, C\}$ only. Now, the number of migrations for $\{1, 3\}$ is 600 whereas for $\{1, 2, 3\}$ it is $100 \times 1 + 500 \times 0 = 100$ (this is because the users on the path $\{B, C\}$ will be connected to fog node 2 and hence will not require any migrations). It might be more profitable in this case to operate the set $\{1, 2, 3\}$. This will depend upon the cost of infrastructure.

The decision that we want to make is to find a set $F \in \mathcal{F}$ which minimizes a linear combinations of the number of migrations and the cost of infrastructure with the constraint that all the base stations in $\mathcal{B}$ are covered. This decision problem can be seen an instance of the Weighted Set Cover Problem [16], which we describe next.

2.1 Weighted Set Cover Problem

The Weighted Set Cover Problem is a known problem in the area of combinatorial optimisation which is NP-complete. It is a generalization of the Set Cover problem which is specified by a set of elements $\{1, 2, ..., n\}$ called the *universe* and a collection S of m sets whose union equals the universe. The Set Cover problem is to identify the smallest sub-collection of S whose union equals the universe. This problem has many practical applications, e.g airline scheduling, vehicle routing, facility location. No polynomial time exact algorithm exists, so a greedy approach can be used as an approximation algorithm. In the greedy strategy, we will iteratively pick the biggest subset that contains the most number of uncovered elements.

In the weighted version of the Set Cover problem a cost function is associated to each subset. The objective is to find the subset that has the least total cost.

Problem 2.1 (Weighted Set Cover) *Given a universe U of n elements, a collection of subsets of U, $S = S_1, \ldots, S_k$, and a cost function $c : \mathcal{S} \to \mathbb{R}_+$, find a minimum cost sub-collection of $\mathcal{S}$ that covers all elements of U. [16]*

In the greedy strategy for the weighted version, instead of picking the subset that adds the subset that has the most number of new elements, we iteratively pick the subset with the smallest $\frac{c(S)}{|S \setminus C|}$.

The greedy weighted set covering heuristic does the following [16]:

Algorithm 1: Greedy Weighted Set Cover algorithm

```
C ← ∅
while C ≠ U do
    Pick S with the smallest c(S)/|S\C|
    C ← C ∪ S
Output the picked sets.
```

2.2 Pair-wise vehicle mobility

In this section, we formulate the minimum number of VM migration problem for a pair-wise mobility model into an ILP problem. By pair-wise mobility model, we mean that the origin and the destination base stations are neighbors and the paths are direct routes from the origin to the destination. In the example in Figure 2, the pair-wise mobility model will have traffic going between only pairs of base stations, that is routes will be $\{A, B\}$, $\{A, C\}$, $\{A, D\}$, $\{B, A\}$, $\{B, C\}$, $\{B, D\}$ and so on. As mentioned in the Introduction, in this paper we will only focus on the pair-wise mobility model and will not be investigating the general mobility model.

Assume we are a given a traffic mobility pattern between various base stations. This pattern is summarized in a matrix, W, whose (i, j)th element, $W_{i,j}$ represents the number of vehicles that move from base station i to base station j per unit time. We shall call W the mobility matrix.

We are also given a connectivity matrix C whose (i,j)th element, $c_{i,j}$, is 1 if data center j meets the SLA for traffic from base station i, and 0 otherwise.

We wish to determine the set of fog nodes to which each base station should send its data traffic to. Implicitly, this determines which fog node sites within the set $\mathcal{F}$ should be operational. The choice of operating or not a fog node is influenced by two conflicting costs. First, there is a cost of operating a fog node which includes the infrastructure as well as the maintenance costs. Thus, larger the number of fog nodes selected by one of more base stations, larger is this cost. The other cost is for service migration. When a vehicles moves out of the range of base station i and into the range of base station j, in order to maintain the continuity of its services, its data has to be migrated from a data center selected by i to one selected by j. This migration cost can be avoided if both i and j have a data center in common. There is a clear conflict between the migration and the operating costs: to reduce migrations it is better to open more data centers which increases the operating costs.

Let $x_k \in \{0,1\}$ be a binary variable that indicates if fog node k is operational or not. It can be shown that the number of service migrations per unit time due to vehicles moving between i and j is given by

$$m_{i,j}(\mathbf{x}) = w_{i,j} z_{i,j} \tag{1}$$

where

$$y_{i,k} = c_{i,k} x_k, \tag{2}$$

$$z_{i,j} = \begin{cases} 1 & \text{if } \mathbf{y}_i \cdot \mathbf{y}_j > 0, \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

Here $z_{i,j}$ indicates whether i and j have a node in common in the set of operational fog nodes indicated by $\mathbf{x}$. Note that $m_{i,j}$ does not depend upon how many fog nodes are in common between i and j as long as there is at least 1.

The optimization problem is formulated as the following Integer Program:

$$\underset{\mathbf{x}}{\text{minimize}}\ g(\mathbf{x}) + \beta \sum_{i,j} m_{i,j}(\mathbf{x}) \tag{4}$$

$$\text{s.t.}\quad \mathbf{x} \in \{0,1\}^{|\mathcal{D}|},$$

$$y_{i,k} = c_{i,k} x_k,\ \forall k \in \mathcal{D}, \tag{5}$$

$$\sum_k y_{i,k} \geq 1,\ \forall i. \tag{6}$$

The function g is the operational cost and could potentially depend upon which fog nodes are open. In the simplest non-trivial case, $g(\mathbf{x}) = \sum_k x_k$, i.e. a linear cost of operating a fog node.

3 Heuristic Algorithm

In our problem, the subsets in the weighted set cover problem are the individual fog nodes. It is not easy to assign a weight or a cost function to each node since this will depend upon the other nodes in the solution. We modify the weighted set cover by using two heuristics cost functions: *(i)* an inverse cost function, and *(ii)* a cost function that depends only on the additional migrations.

3.1 Weights / Cost function

We have calculated the weights through a weight function that counts that sum of incoming traffic to each of base stations in a subset. Given W the mobility matrix, for a subset S, $\sum_{j \in S} \sum_{i \to j} W_i$. Our model wants to

Table 1: Variables used in the ILP formulation

Notation	Description
W	mobility matrix
C	connectivity matrix
x_k	A binary variable indicating whether tfog node k is operational or not
z_{ij}	A binary variable indicating whether i and j have a fog node in common
w_{ij}	variable in the mobility matrix, represents the number of vehicles moving from base station i to base station j per unit time
m_{ij}	Number of service migration per unit time
c_{ij}	binary variable in the connectivity matrix, if 1 it means the j fog node meets the SLA for service coming from i
$g(\mathbf{x})$	operational cost which depends upon which fog nodes are open.

favor higher weights. Since the weighted set cover algorithm minimizes the weight we are taking the inverse of the weights.

3.2 Modified weighted set cover

For $j \in \mathcal{F}$, denote B_d, the set of base stations covered by j. With slight abuse of notation we will replace j by a subset of $\mathcal{F}$ in which case B_A will be the set of base stations covered by the subset A. Also, for $A \subset \mathcal{F}$, A^c will be denote the complement inside $\mathcal{F}$, that is, the set $\mathcal{F} \setminus A$.

For a $A \subset D$ define:

$$\gamma_A(k) = \frac{\sum\limits_{(i,j) \notin B_k \setminus B_A} m_{i,j}}{|B_k \setminus B_A|}, \ \forall k \notin A. \tag{7}$$

The ratio $\gamma_A(k)$ can be interpreted as additional migrations per base station that will be covered if k is added to A. A higher value of γ, indicates that this fog node eliminates the need for a larger number of migrations and that it has a lower marginal cost.

Algorithm 2: Modified weighted set cover

```
Output: A
C ← ∅
A ← ∅
while C ≠ B do
    k̂ = arg min_{k∈A^c} γ_{A^c}(k)
    C ← C ∪ B_{k̂}
    A ∪ k̂
```

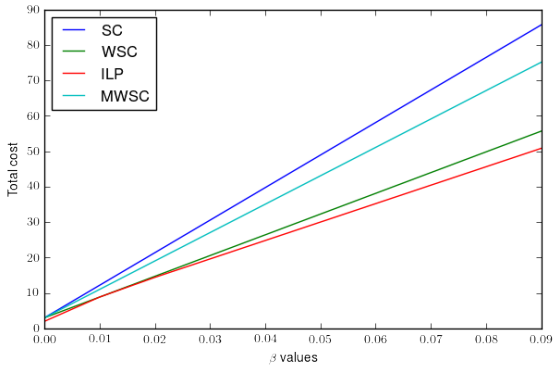

Figure 3: Cost as a function of β for Set Cover (SC), Weighted Set Cover (WSC), Modified Weighted Set Cover (MWSC) and the ILP formulation.

4 Performance Evaluation

In this section we describe the results obtained from three different heuristics: a plain set cover implementation, a weighted set cover implementation where weights represent the inverse of inbound traffic on each fog node and a modified version of weighted set cover which does not start with explicit weights but computes them greedily. We also show the solution to ILP which was computed using Gurobi [6] which is a solver for optimization problems.

In the first example, we show how the solution of the different heuristics varies with β which is the weight if the migrations. For Figure 3 we took $\mathcal{B} = 10$ and $\mathcal{F} = 5$. Thus C is a 10×5 matrix and W is a 10×10 matrix. The number of cars was between 0 and 100 in the mobility matrix W.

When β is close to 0, the ILP is like a set cover problem as the cost of migrations is not taken into account in the optimization. For larger values of β, the migrations become important and the greedy algorithm for set cover becomes worse.

In the second set of experiments we increase the number of base stations and fog nodes. We take different pairs of $(\mathcal{B}, \mathcal{F})$ with maximum $\mathcal{B} = 50$ and maximum $\mathcal{F} = 15$. The traffic matrix has random entries between 0 and 20, and 100 different random matrices were generated. For $\beta = 0.01$, the average total cost is shown in Fig. 4 and the average execution time is shown in Fig. 5. The results show while ILP finds solution with lower cost, it is also the slowest to execute.

For $\beta = 0.001$, the results are shown in Fig. 6 and 7. Again for smaller values of β, the heuristics are close to ILP but with a smaller execution time than ILP.

Finally, in Fig. 8 and 9 we plot the results for maximum $\mathcal{B} = 60$ and maximum $\mathcal{F} = 20$. The traffic matrix has random entries between 0 and 20, and 100 different random matrices were generated, and $\beta = 0.001$

5 State-of-the art

A great amount of state-of-the art literature on mobility induced service migrations is covered by the recent study [11]. The authors present recent advances where service migration has been studied on divers paradigms as cloudlets, Fog computing, cloud-based vehicular networks and multi-access edge computing.

There are different types of service migration discussed in the literature, including the migration of virtual machines (VMs), containers or processes, or of programs using an operating system and hardware architecture agnostic binary format such as WebAssembly [1].

We can also mention that [2] discuss Volley, an automatic service placement for geographically distributed DCs based on iterative optimization algorithms. Volley migrates services to new DCs if the capacity of a

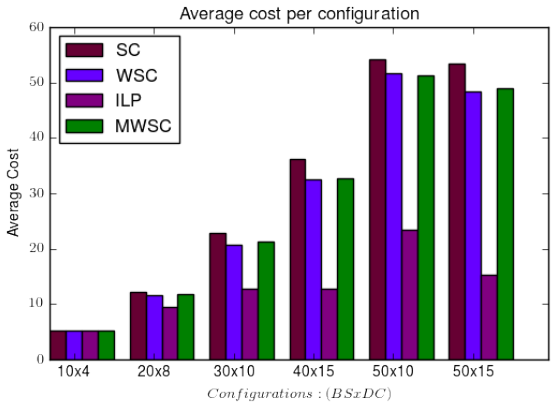

Figure 4: Comparison of the Set Cover (SC), Weighted Set Cover (WSC), Modified Weighted Set Cover (MWSC) and the ILP formulation. Each algorithm was executed 100 times on each configuration

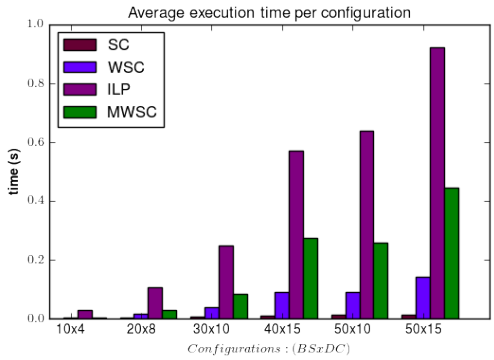

Figure 5: Execution times of the Set Cover (SC), Weighted Set Cover (WSC), Modified Weighted Set Cover (MWSC) and the ILP formulation. Each algorithm was executed 100 times on each configuration

DC changes or the user changes location.

Service migrations in Fog Computing infrastructure borrows ideas and combines techniques used in handovers in cellular networks and live migration in Cloud computing. A reader can refer to survey [17] where the authors compare these two concepts with service migration in MEC as well their different scopes and similarities.

Many publications on service migration refer to the works on "Follow me" approaches [14][12][9][3][13].

In [7] authors propose a quality-of-service aware scheme based on the existing handover procedures to support the real-time vehicular services. A case study based on a realistic vehicle mobility pattern for Luxembourg scenario is carried out, where the proposed scheme, as well as the benchmarks, are compared by analyzing latency and reliability as well as migration cost.

More theoretical results are given in [13], [15]. In the first study, the authors use Markov Decision Processes to capture the tradeoff between migration cost and user experience. In the second study, the authors focus on the similar problem. They model this tradeoff due to service migration's network overhead and latency for the use. Since service migration affect workload scheduling they tackle this decision jointly. Instead of using dynamic programming to solve MDPs, they decouple the MDPs and apply the technique of Lyapunov optimization of control theory.

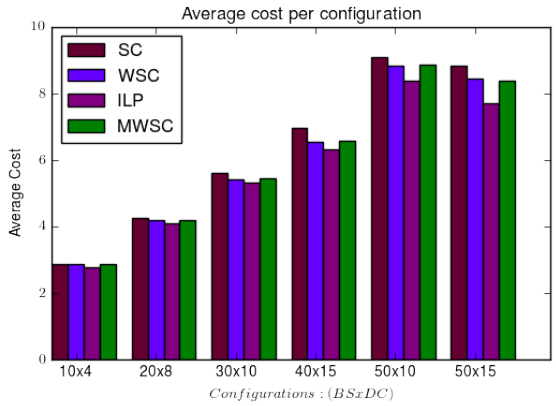

Figure 6: Comparison of the Set Cover (SC), Weighted Set Cover (WSC), Modified Weighted Set Cover (MWSC) and the ILP formulation. Each algorithm was executed 100 times on each configuration

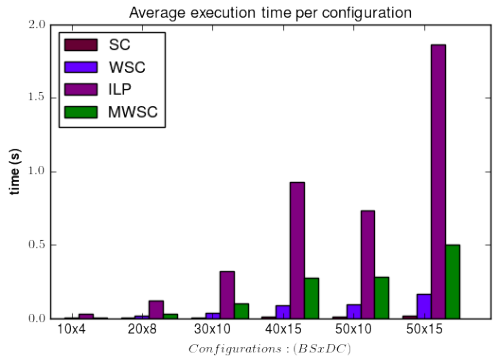

Figure 7: Execution times of the Set Cover (SC), Weighted Set Cover (WSC), Modified Weighted Set Cover (MWSC) and the ILP formulation. Each algorithm was executed 100 times on each configuration

In [18] a contribution on service migration in the area of VANETS is given. The authors present a Mixed Integer Quadratic Programming formulation. As an input they have a graph of RSUs, resources, vehicles and sessions and they try to solve a minimum network cost VM migration problem. They then propose an heuristic algorithm with polynomial time.

Another theoretical work is [4], where authors as well propose a VM migration approach based on mobility prediction. The authors provide an ILP model for VM placement in Fog computing. The problem they are dealing with is different from ours. Their algorithm defines the set of candidate cloudlets to receive the users VM according to the users future position. In their model they want to maximize the accepted requests while minimizing the user's latency. They execute these two objective functions sequentially.

We finally mention tools such as described in [8], which is an extension of the iFogSim simulator [5] adding virtual machine migration policies for mobile users. Their VM migration policy takes into account user's position, speed and direction. As well they define, the point where the migration can be potentially triggered by specifying the geographical zone based on user's coordinates. *FogNetSim* has been proposed in [10]. *FogNetSim* is an extension of *OMNET++* that is used to simulate a network. The tool simulates among others geographically distributed data centers while providing support for handovers among Fog nodes. The implementation includes a number of mobility models. It is a release of 2018 and as future work authors

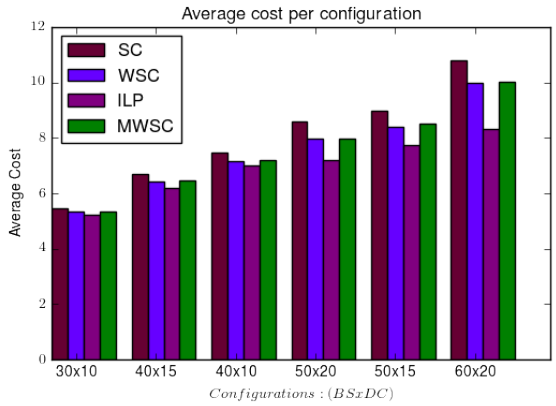

Figure 8: Comparison of the Set Cover (SC), Weighted Set Cover (WSC), Modified Weighted Set Cover (MWSC) and the ILP formulation. Each algorithm was executed 100 times on each configuration

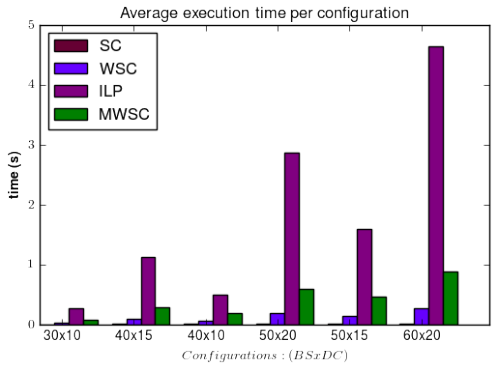

Figure 9: Execution times of the Set Cover (SC), Weighted Set Cover (WSC), Modified Weighted Set Cover (MWSC) and the ILP formulation. Each algorithm was executed 100 times on each configuration

mention the support for VM migration. [11]

6 Conclusions and Future Work

We presented an Integer Linear Program for obtaining the optimal location of fog nodes that minimizes a linear combination of the number of migrations and the cost of infrastructure. This was done for a pairwise mobility model in which the origin and destination base stations are neighbors. We gave two heuristics based on the Weighted Set Cover problem and evaluated the performance of these heuristics with that of the optimal solution.

The research community will benefit from an optimization framework for service migrations taking mobility patterns in consideration.

As future work, we will investigate the general mobility model. For this, other heuristics will have to be proposed and we will study the approximation ratio of these.

Acknowledgment

We gratefully acknowledge the funding received from Continental Digital Services France.

References

[1] Web Assembly. `https://webassembly.org/`.

[2] Sharad Agarwal, John Dunagan, Navendu Jain, Stefan Saroiu, Alec Wolman, and Harbinder Bhogan. Volley: Automated data placement for geo-distributed cloud services. In *Proceedings of the 7th USENIX Conference on Networked Systems Design and Implementation*, NSDI'10, pages 2–2, Berkeley, CA, USA, 2010. USENIX Association.

[3] A. Aissioui, A. Ksentini, A. M. Gueroui, and T. Taleb. On enabling 5g automotive systems using follow me edge-cloud concept. *IEEE Transactions on Vehicular Technology*, 67(6):5302–5316, June 2018.

[4] Diogo Gonçalves, Karima Velasquez, Marília Curado, Luiz Fernando Bittencourt, and Edmundo Roberto Mauro Madeira. Proactive virtual machine migration in fog environments. *2018 IEEE Symposium on Computers and Communications (ISCC)*, pages 00742–00745, 2018.

[5] Harshit Gupta, Amir Vahid Dastjerdi, Soumya K. Ghosh, and Rajkumar Buyya. ifogsim: A toolkit for modeling and simulation of resource management techniques in internet of things, edge and fog computing environments. *Softw., Pract. Exper.*, 47:1275–1296, 2016.

[6] LLC Gurobi Optimization. Gurobi optimizer reference manual, 2018.

[7] J. Li, X. Shen, L. Chen, D. P. Van, J. Ou, L. Wosinska, and J. Chen. Service migration in fog computing enabled cellular networks to support real-time vehicular communications. *IEEE Access*, 7:13704–13714, 2019.

[8] Márcio Moraes Lopes, Wilson A. Higashino, Miriam A.M. Capretz, and Luiz Fernando Bittencourt. Myifogsim: A simulator for virtual machine migration in fog computing. In *Companion Proceedings of the10th International Conference on Utility and Cloud Computing*, UCC '17 Companion, pages 47–52, New York, NY, USA, 2017. ACM.

[9] T. Ouyang, Z. Zhou, and X. Chen. Follow me at the edge: Mobility-aware dynamic service placement for mobile edge computing. *IEEE Journal on Selected Areas in Communications*, 36(10):2333–2345, Oct 2018.

[10] T. Qayyum, A. W. Malik, M. A. Khan Khattak, O. Khalid, and S. U. Khan. Fognetsim++: A toolkit for modeling and simulation of distributed fog environment. *IEEE Access*, 6:63570–63583, 2018.

[11] Zeineb Rejiba, Xavier Masip-Bruin, and Eva Marín-Tordera. A survey on mobility-induced service migration in the fog, edge, and related computing paradigms. *ACM Comput. Surv.*, 52(5):90:1–90:33, September 2019.

[12] T. Taleb, P. Hasselmeyer, and F. G. Mir. Follow-me cloud: An openflow-based implementation. In *2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing*, pages 240–245, Aug 2013.

[13] T. Taleb and A. Ksentini. An analytical model for follow me cloud. In *2013 IEEE Global Communications Conference (GLOBECOM)*, pages 1291–1296, Dec 2013.

[14] T. Taleb and A. Ksentini. Follow me cloud: interworking federated clouds and distributed mobile networks. *IEEE Network*, 27(5):12–19, Sep. 2013.

[15] Rahul Urgaonkar, Shiqiang Wang, Ting He, Murtaza Zafer, Kevin Chan, and Kin K. Leung. Dynamic service migration and workload scheduling in edge-clouds. *Perform. Eval.*, 91(C):205–228, September 2015.

[16] Vijay V. Vazirani. *Approximation Algorithms*. Springer-Verlag, Berlin, Heidelberg, 2001.

[17] S. Wang, J. Xu, N. Zhang, and Y. Liu. A survey on service migration in mobile edge computing. *IEEE Access*, 6:23511–23528, 2018.

[18] Hong Yao, Changmin Bai, Deze Zeng, Qingzhong Liang, and Yuanyuan Fan. Migrate or not? exploring virtual machine migration in roadside cloudlet-based vehicular cloud. *Concurr. Comput. : Pract. Exper.*, 27(18):5780–5792, December 2015.