

A SIMHEURISTIC ALGORITHM FOR SERVICE PLACEMENT IN COMMUNITY NETWORKS

Javier Panadero
Laura Calvet
Christopher Bayliss
Joan Manuel Marquès

Mennan Selimi

IN3 – Computer Science Dept.
Universitat Oberta de Catalunya
Av. Carl Friedrich Gauss 5
Castelldefels, 08860, SPAIN

Max van der Stoel Institute
South East European University
Tetovo 1200, NORTH MACEDONIA

Felix Freitag

Department of Computer Architecture
Universitat Politècnica de Catalunya
Barcelona, 08028, SPAIN

ABSTRACT

The growing demand for network connectivity has boosted the number of community networks (CNs). CNs are decentralized and self-organized communication networks owned and managed at the edge by volunteers. Due to the heterogeneity of edge node characteristics, high software and hardware diversity, irregular topology and unreliable behavior of the network, the performance of its services varies depending on where they are hosted. These characteristics of CNs and edge platforms running on them require advanced simulation-optimization methods to place services. In this context, we propose a simheuristic algorithm to address this stochastic problem. The core of this approach relies on a multi-start metaheuristic with a multi-objective optimization method. Our approach combines Monte Carlo simulation and the multi-criteria optimal placement heuristic. The method is tested using real traces of *Guifi.net* CN, which is considered to be the largest CN worldwide.

1 INTRODUCTION

Community networks (CNs) started to attract attention in the early 2000s, due to the need of connectivity in remote areas. CNs are owned and managed at the edge by volunteers, who share their resources. Typically, these networks tend to be large-scale, heterogeneous and distributed. *Guifi.net* (https://guifi.net/en/what_is_guifinet) is an example of a well-known CN. The project was originally created to solve the broadband Internet access difficulties in rural areas in Catalonia (Spain) and today it has more than 36,000 operational nodes (Cerdà-Alabern et al. 2020).

The consolidation of today's cloud technologies offer CNs the possibility to collectively build CN micro-clouds (Lertsinsruttavee et al. 2018), building upon user-provided networks, and extending towards an ecosystem of cloud services. In CN micro-clouds, services are hosted at edge nodes with communication,

computation, and storage capabilities. The micro-clouds can include micro-servers, routers, IoT gateways, mobile devices, etc., and they are connected to the wireless mesh network (WMN) and provide low-latency service to users that are within a suitable communication distance. Services can be distributed storage, live-video streaming or other IoT service.

Unfortunately, the provisioning of these type of services is not simple. Given the fact that volunteers in CNs are free to connect and disconnect their resources whenever they want, CNs may suffer from a lack of reliability, which can cause significant delays or a loss of data. Further, given the characteristics of a communication over a wireless channel, unreliable network and user devices at non-optimal locations, the physical topology of the CN where the micro-clouds are deployed is in a constant state of flux. Therefore, the question arises whether the performance (*i.e.*, service performance) of a CN can be improved by carefully choosing the micro-clouds that are going to host a particular service. Key factors to take into account when answering this question are the bandwidth (*i.e.*, maximize the overall available bandwidth between nodes), availability of the nodes, resource usage (*e.g.*, CPU, memory), etc. Many authors in the literature have modeled the service placement as a combinatorial optimization problem (COP) ignoring the *stochasticity* of the bandwidth between nodes (Selimi et al. 2019), that is, simplifying the problem.

In this context, we present a simheuristic approach (Juan et al. 2015) for service placement in CN micro-clouds. Simheuristics are algorithms that integrate simulation into a metaheuristic framework to address a stochastic COP. The simheuristic designed integrates Monte Carlo simulation (MCS) (Taimre et al. 2019) into a multi-start metaheuristic (Martí et al. 2013): while the metaheuristic component searches for promising solutions, the simulation component assesses each of these solutions in a number of scenarios and computes an average objective function value. The metaheuristic is an iterative approach relying on a multi-objective optimization method named *multi-criteria optimal placement (MCOP)*. Hence, the main contribution of this work is to propose a simheuristic to find a service configuration that maximizes the overall expected bandwidth among the service instances under an stochastic environment. To the best of our knowledge, previous work in the literature has only considered the deterministic version of the problem. Incorporating uncertainty is particularly important as it allows the optimization to account for variations in the bandwidth of the network, due to the unreliable behavior of this kind of networks. A set of computational experiments are carried out using real traces from *GuifiSants*, which is a subset of *Guifi.net*. These experiments enable us to validate our approach and compare it against an approach not taking into account the stochasticity.

2 RELATED WORK

2.1 Deterministic Version

Placement in decentralized clouds. A smart resource placement is key in distributed cloud systems, which are geographically distributed over a large number of locations. Pasteris et al. (2019) proposes an approximation algorithm for solving the service placement problem in MEC systems with heterogeneous service/node sizes and rewards. Their algorithm has a constant approximation ratio. Similarly, Farhadi et al. (2019) proposes a two-time-scale solution for joint service placement and request scheduling in edge clouds under communication, computation, and storage constraints. They prove the NP-hardness of the service placement problem in the general case and characterize its complexity in all special cases. Zhang and Tang (2014) presents two algorithms for the distributed cloud selection and VM partition problems. They are based on a novel method, which minimizes the maximum inter-distance between clouds considering the distance between clouds and networking information such as topology and density. Given the increasing popularity of online social networks (OSN), there are many works on the assignment of resources in OSN that run over large-scale distributed clouds. Jiao et al. (2016) studies the problem of optimizing the monetary cost spent on cloud resources when deploying an OSN service over multiple geo-distributed clouds. The authors model the cost of data placement, quantify the QoS, and address data availability by ensuring a minimum number of replicas for each user. Relying on these models, they propose *cosplay*, an algorithm

that minimizes the cost while ensuring the QoS and the data availability. Similarly, Xia et al. (2017) analyzes the problem of user data placements of OSN in a distributed cloud minimizing the operational cost of a cloud service provider. This work proposes a fast yet scalable algorithm, which builds on the use of the community concept by grouping users of a social network into different communities and placing the master replicas of user data of the users in the same community into a data center. In addition, it replicates their slave replicas of the user data into nearby data centers. Khalajzadeh et al. (2016) presents a genetic algorithm-based approach to optimize social media data placement and replication in distributed clouds. It attempts to find a near-optimal number of replicas for every user's data and a near-optimal placement of replicas to minimize monetary cost while satisfying latency requirements for all users.

Placement in wireless environment. Wireless networks have been gaining popularity during the last few decades, which has been accompanied by an increment of the number of related works. Selimi et al. (2019) puts forward a service placement algorithm called BASP, to place micro-cloud services in CNs. The algorithm uses K-Means for clustering and a lightweight bandwidth computation/estimation heuristic. Another study about placement in CN is proposed by Selimi et al. (2016), which optimizes the communication cost in CN by minimizing: (i) the service overlay diameter; and (ii) the coordination cost along the network. Coimbra et al. (2018) proposes a novel service placement approach based on community finding using a scalable graph label propagation technique and decentralized election procedure.

There are some works that consider a different type of wireless networks. For example, Herrmann (2010) proposes a fully decentralized, dynamic, and adaptive service placement algorithm for ambient intelligence (Ducatel et al. 2001) environments like the ad-hoc service grid infrastructures (Herrmann et al. 2005). This algorithm achieves a coordinated global placement pattern that minimizes the communication costs without any central controller. It does not even require additional communication among the replicas. Tarneberg et al. (2017) presents an algorithm that holistically returns the globally optimal placement of static and mobile applications in mobile clouds networks. Its goal is to minimize the global system cost as a means to manage the compute and network resources.

2.2 Stochastic Version

Despite its practical relevance, to the best of our knowledge, previous works of service placement in community networks have not considered stochastic version of the problem. Nevertheless, there are some closely works which have focused on placement methods with stochastic components in distributed systems. For instance, Panadero et al. (2017) presents a mathematical formulation and a solving approach based on a metaheuristic for the resource allocation problem large-scale distributed systems. The approach is designed to deal with data-intensive applications, which must guarantee the availability of the data at all times. The authors also propose a simheuristic to deal with the stochasticity of resources' quality. Shabtai et al. (2018) proposes a stochastic based placement algorithm to find a solution that minimizes the expected total cost of ownership in case of two data centers. Badri et al. (2020) addresses the energy-aware application placement problem in mobile edge computing systems. They formulate the problem as a multi-stage stochastic program and design a parallel greedy SAA-based algorithm for solving the application placement problem. For investigating its performance they are using real-world trace data. Yan et al. (2019) proposes a stochastic VM placement algorithm that takes into account the uncertainty of resource requirements while minimizing the total energy consumption in a cyber-physical cloud system. Different from existing approaches, the proposed stochastic VM placement algorithm models the uncertainty of resource requirements as random variables, and further formulates the uncertainty-affected VM placement problem as a stochastic optimization model.

3 PROBLEM DESCRIPTION

A CN is modeled as an incomplete directed graph $G = (N, E)$, where N is the set of nodes and E is the set of links connecting pairs of nodes. Each node has a particular $QoS_i, \forall i \in N$, derived from the real

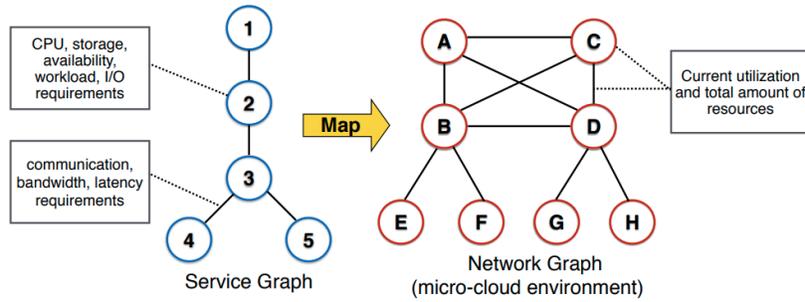


Figure 1: Service placement problem.

measurements in the CN. Each link is characterized by a given bandwidth B_{ij} , $\forall(i, j) \in E$, and latency L_{ij} , $\forall(i, j) \in E$. Given a micro-cloud to be deployed, at most R_{\max} replicas can be placed. A micro-cloud can be deployed in a node only if this node has a QoS greater than a minimum threshold (QoS_{min}). Similarly, a given link will be used only if its bandwidth is higher or equal to a given threshold (B_e). Each node not deploying any micro-cloud has to be assigned to at least one node deploying one. However, a node with a micro-cloud deployed is not assigned to any node but is required to have at least a different node assigned. The path from node i to node j is defined as the path with minimum length among all the possible paths. If two paths have the same length, the path with maximum bandwidth is selected. The bandwidth of a given path is described as the minimum bandwidth among its links. The bandwidth of a given micro-cloud is the mean bandwidth of the paths associated.

CNs require using the available bandwidth resources to reach their full potential. For this reason the objective is to design a deployment that maximizes the mean value of the micro-clouds' bandwidths. By optimizing this measure, we aim to avoid or reduce the risk of bottlenecks. A solution is described by three types of binary variables: (i) x_i identifies the nodes where the replicas are placed; (ii) y_{ij} reveals whether node i is assigned to node j ; and (iii) z_{ijkl} shows the path from the nodes where i is assigned to itself, thus z_{ijkl} is equal to 1 if arc (k, l) is traversed when serving node i from micro-cloud j .

We can abstract the service placement problem as a graph (*i.e.*, node) placement problem as illustrated in the Figure 1. Service placement can be seen as an application of facility location problem (FLP) to ad hoc networking (*i.e.*, wireless CNs). Based on this, the service placement problem is the problem of mapping the service graph into an actual network graph, subject to some network or node constraints (*e.g.*, bandwidth, latency, availability, etc).

4 THE METAHEURISTIC APPROACH

This section describes the metaheuristic approach, which builds on the MCOP heuristic. This heuristic is an efficient multi-objective optimization method to select the most suitable nodes to place services in large-scale CNs. It aims to maximize the overall available bandwidth among the micro-cloud nodes, while minimizing the number of replicas of a microcloud node instance to avoid inefficient resource usage.

4.1 Description of the Heuristic

The first stage of the method is based on a multi-objective optimization method inspired on the lexicographic ordering (LO) approach (Marler and Arora 2004) to generate an initial solution. By relying on this approach, we avoid having to add up quantitative parameters using weight factors. Instead, the priorities of the individual planning criteria (*i.e.*, the objective functions) are explicitly incorporated in the optimization. In particular, the objective functions are ranked in terms of importance and are minimized (or maximized) sequentially.

The inputs of the method are: a complete list of nodes, a set of criteria parameters (which are ranked by importance), the minimum (or maximum) acceptable values of these parameters, and a list of constraints. Three different criteria parameters are considered (ranked by importance): node quality,

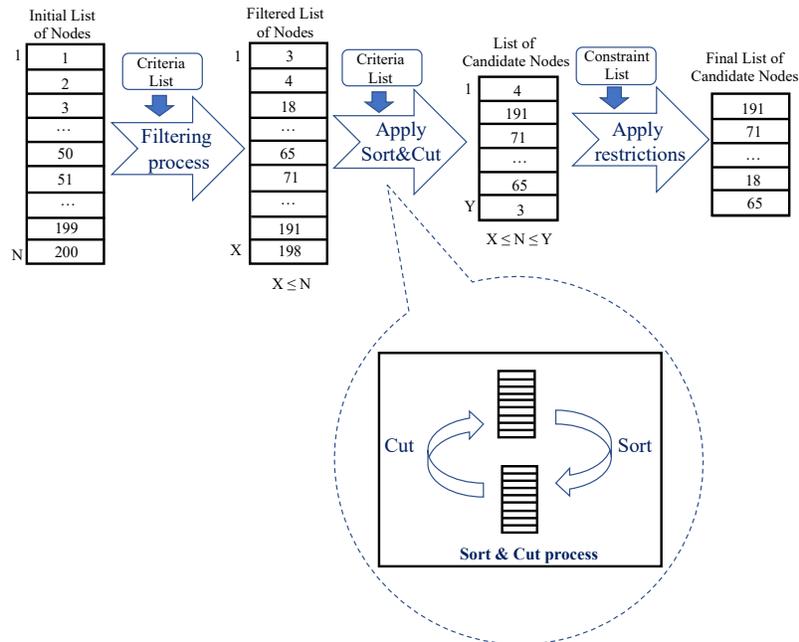


Figure 2: Overview of the MCOP heuristic.

number of connections (which indicates for a given node the number of adjacent nodes), and latency (which indicates the largest latency among all the links starting at a given node and is expressed in MB/s). There are two constraints: the maximum number of replicas and the minimum number of hops between the nodes where the replicas are placed. This last restriction aims to select nodes distributed along the network, minimizing the distance between the replica nodes and their assigned nodes.

Fig 2 illustrates an overview of the first stage. Initially, the list of active nodes is filtered by the criteria parameters; thus, the nodes that do not meet the minimum/maximum acceptable values are removed. This makes the procedure more efficient, since it discards at the beginning nodes without possibility of being selected. Then, the iterative procedure *sort&cut* is applied taking into account the priority of the criteria parameters. First, the list is sorted by the first criterion (the most important) in an ascendant order if the corresponding value has to be maximized or in descendent order otherwise. Then, a percentage of nodes is selected (these are elite nodes). After this first iteration, the procedure continues reducing the size of the list considering a different criterion at each iteration. By default, we set $\%elite_i \forall i \in \{1, 2, 3\}$ to 10%. The loop stops when all the criteria have been considered. From the returned list of best ranked nodes, we select the final subset of nodes taking into account a final list of restrictions. Having selected the nodes in which services will be placed (so called candidate nodes), the other nodes have to be assigned so that they have access to a service deployed. This assignation relies on the Breadth First Search (BFS) algorithm (Even 2011), which is a well-known algorithm for traversing graphs. The output of this algorithm is a set of sub-graphs that contain a candidate node as a root node and the nodes with a depth lower or equal to the maximum number of hops allowed. Notice that, since the BFS algorithm relies only on distance, a node can be served by more than one candidate node. Then, the average bandwidth of each sub-graph is computed to obtain the system's bandwidth.

The second stage applies an heuristic that, given an initial solution, reduces the number of replicas used while considering a minimum loss overall available bandwidth (less than 1%). By reducing the replication of redundant services in the network, this stage helps to alleviate the resources' load. The main idea behind this heuristic is to merge the sub-graphs, generated in the previous step, with a low number of nodes, generating larger sub-graphs and thus avoiding the use of resources to serve few nodes.

The heuristic receives as inputs the set of candidate nodes, the complete graph, and the initial solution. Then, all the shared nodes in different sub-graphs are identified and stored in a list (*sharedNodes*). Subsequently, the heuristic starts an iterative procedure, where in each iteration a shared node is considered. In each iteration, the sub-graphs containing this shared node are sorted by size (from bigger to smaller). Then, the algorithm considers merging the sub-graph with the lowest number of nodes (*baseSubGraph*) with each of the other sub-graphs, establishing the shared node as candidate node for the resulting new sub-graph (*newSubGraph*). For each potential merging, the heuristic checks if the resulting sub-graph is promising. In particular, a sub-graph is labelled as promising if its bandwidth is at most a 1% lower than the average bandwidth of the two sub-graphs considered. If a merge is promising, the heuristic assesses whether the bandwidth of *newSubGraph* improves the bandwidth of the previous merges. Once all the potential merges have been analyzed, the heuristic merges *baseSubGraph* with the sub-graph with the highest bandwidth. Once this process finishes, the current solution (*currentSol*) is accordingly updated and its bandwidth is computed.

4.2 Biased Randomization Techniques

The MCOP heuristic is a constructive approach that applies a deterministic procedure to generate a solution. Hence, if the method is executed several times, the same solution is obtained. Although constructive methods may provide good solutions, they could fall in a local minima. In order to avoid it and improve the quality solution, the search has to be diversified. An efficient way to achieve this is to rely on biased randomization techniques (Juan et al. 2010). These techniques enable transforming the deterministic behavior of constructive methods into a probabilistic one by using skewed probability distributions to induce an ‘oriented’ (non-uniform) random behaviour. This process turns a deterministic heuristic into a randomized one whilst preserving the logic behind the original heuristic. Thus, biased randomization techniques allow for generating a series of solutions.

The first stage of the MCOP heuristic is enriched by integrating biased randomization techniques. The oriented random behaviour is introduced by means of a Geometric probability distribution, which has a parameter β ($0 < \beta < 1$). This distribution assigns a higher probability to the most promising elements, giving them more opportunities to be selected. Hence, the method introduces randomness in the node selection process, assigning a probability of (approximately) β to the best ranked nodes of the candidate list (promising nodes), and probabilities with an exponential diminishing pattern to the rest of the nodes of the candidate list. Since the algorithm does not always select the most promising node, the procedure allows diversification and load balancing, keeping the logic behind the sort.

We have wrapped our biased-randomized heuristic into a multi-start framework, which is a sequential and iterative approach. Multi-start methods are composed of two phases: a first one in which a new solution is generated using the biased-randomized heuristic, and a second one in which the algorithm compares the newly generated solution with the best solution obtained so far –updating the latter whenever appropriate.

5 OUR SIMHEURISTIC APPROACH

This section introduces our simheuristic approach, which builds on the MCOP heuristic, to address the stochastic version of the problem. This version models the nodes’ bandwidth as random variables. Our approach is a multi-start algorithm that combines a biased-randomized version of the MCOP heuristic with MCS. The use of a multi-start algorithm enables the diversification of the search in a fast and efficient way, restarting the process from a new solution once a region has been extensively explore. The multi-start algorithms usually have two sequential phases: the generation of the initial solution and an iterative process to improve it. The integration of MCS allows the algorithm to deal with the stochastic nature of the problem.

Our approach, composed of three stages, is summarized in Algorithm 1 and described next. First, a feasible initial solution (*initSol*) is generated using the deterministic MCOP heuristic. Then, *initSol* is submitted to an evaluation process using stochastic data. In particular, the algorithm relies on MCS to

simulate a relatively low number of scenarios (where each scenario is described by a value for each random variable) and compute the average objective function value (*i.e.*, the average bandwidth) across all the scenarios. To keep the computational time low, only 1,000 scenarios are simulated (*fastSimulation*). At this point, *initSol* is stored as the best solution found so far (*bestSol*). Then, during the second stage (lines 5 to 16), an iterative process relying on the biased-randomized MCOP heuristic is initiated to explore the search space. In each iteration, a new solution (*newSol*) is generated. If *newSol* outperforms *bestSol* in the deterministic scenario (where no stochasticity is considered), it is sent through a fast MCS process to estimate its stochastic average bandwidth (which is computed as the average bandwidth among the scenarios). If the stochastic average bandwidth of *newSol* improves that of *bestSol*, the latter is updated and added to the pool of ‘elite’ solutions. This process is repeated until the algorithm reaches the maximum computational time. From this stage, a reduced set of elite solutions is obtained. In the third stage (lines 17 to 23), a refinement procedure using a larger number of MCS runs is applied to these elite solutions in order to obtain more accurate estimates of the expected bandwidth. Afterwards, the algorithm returns the best stochastic solution obtained. Since the number of generated solutions during the search can be large and the simulation process is time-consuming, we limit the number of MCS runs to 50,000 scenarios.

Algorithm 1 Simheuristic Approach

```

1: procedure SIMHEURISTIC (graph, maxNReplicas, maxCompTime,  $\beta$ )
2:   initSol  $\leftarrow$  MCOP(graph, maxNReplicas)
3:   fastSimulation(initSol)
4:   bestSol  $\leftarrow$  initSol
5:   compTime  $\leftarrow$  0
6:   while (compTime < maxCompTime) do
7:     newSol  $\leftarrow$  MCOP(graph, maxNReplicas,  $\beta$ )
8:     if (detBW(newSol) > detBW(bestSol)) then ▷ detBW: deterministic bandwidth
9:       fastSimulation(newSol)
10:      if (stochBW(newSol) > stochBW(bestSol)) then ▷ stochBW: stochastic bandwidth
11:        bestSol  $\leftarrow$  newSol
12:        insert(poolBestSols, bestSol)
13:      end if
14:    end if
15:    updateTime(compTime)
16:  end while
17:  for (sol  $\in$  poolBestSols) do
18:    deepSimulation(sol)
19:    if (stochBW(sol) > stochBW(bestSol)) then
20:      bestSol  $\leftarrow$  sol
21:    end if
22:  end for
23:  return bestSol
24: end procedure

```

6 COMPUTATIONAL RESULTS

6.1 Implementation Details and Description of the Experiments

The proposed simheuristic has been implemented using Java SE 8.0 and tested on a workstation with a multi-core processor Intel Xeon E5-2650 v4 with 32GB of RAM. Real traces from *GuifiSants*, which is a subset of *Guifi.net*, have been employed for the experiments. *GuifiSants* is composed of around 80 nodes (with more than 200 users) and two gateways (*i.e.*, proxies) that connect the network to the rest of *Guifi.net* and

the Internet (Dimogerontakis et al. 2017). The data has been obtained from <http://dsg.ac.upc.edu/qmpsu/>, which is a live monitoring page where the data is publicly available. It provides information for each hour of the day. We have logs for 31 successive days (1 month) and 74 nodes. 10 different days (instances) of the system have been randomly selected to test our approach. We have extended the logs to incorporate stochastic bandwidths. To extend the aforementioned logs, we have assumed that the bandwidth between two nodes i and j , $B_{(i,j)}$, follows a Log-Normal probability distribution. To choose this distribution, we have carried out an analysis of the *Guifi.net* traces, and we show that the log-normal distribution fit properly to model the bandwidth. As discussed in Alasmar et al. (2019), this distribution is a natural choice for modeling networks. The Log-Normal distribution has two parameters: the location parameter, μ , and the scale parameter, σ , which relate to the expected value $E[B_{i,j}]$ and the variance $Var[B_{i,j}]$ as follows:

$$\mu_{i,j} = \ln(B_{i,j}) - \frac{1}{2} \ln \left(1 + \frac{Var[B_{i,j}]}{E[B_{i,j}]^2} \right) \quad (1)$$

$$\sigma_{i,j} = \left| \sqrt{\ln \left(1 + \frac{Var[B_{i,j}]}{E[B_{i,j}]^2} \right)} \right| \quad (2)$$

We have set the variability of the bandwidth with reference to the deterministic equivalents such that $Var[B_{i,j}] = c \cdot B_{i,j}$ and $c \geq 0$. The parameter c is a design parameter, which allows us to experiment with different levels of uncertainty. It is expected that as c converges to zero, the results from the stochastic version converge to those obtained in the deterministic scenario. Three different levels of uncertainty are considered: low ($c = 0.1$), medium ($c = 0.5$), and large ($c = 1.0$).

6.2 Analysis of the Results

The results for the different days are presented in Table 1. The first column identifies the instant, while the rest of columns are split into two parts. In the first part (columns 2-6), our approach is validated using the original logs, *i.e.*, without considering stochasticity. The *OBS – D* column shows the objective function value of our best deterministic solution, while the third column displays the computational time (in seconds) required to reach it. The next two columns correspond to the results obtained using the BASP method (Selimi et al. 2019), which reports best known solutions (BKS) in the literature. Subsequently, the next column shows the corresponding percentage gap between our approach and the BASP method. The second part of the table reports the results obtained considering stochastic bandwidths between nodes. The first three columns (*OBS – S – L*, *OBS – S – M*, and *OBS – S – H*) show the expected bandwidth obtained using our simheuristic considering a low, medium, and high level of uncertainty, respectively. The following three columns (*OBS – D – L*, *OBS – D – M*, and *OBS – D – H*) show the expected bandwidth obtained when the best deterministic solution is evaluated under a stochastic scenario with low, medium and high levels of uncertainty, respectively. Finally, the last columns report the percentage gaps between the stochastic solutions and the deterministic solution evaluated considering different levels of uncertainty.

The results show that our approach provides high-quality solutions for the deterministic instances. Notice that our algorithm outperforms the solutions provided by the BASP method in terms of bandwidth (average gain of 25.85%). Regarding the computational time, on average, the MCOP method needs 42.01 seconds to find the best solution, while the BASP method needs 14.47 seconds. Although the MCOP method invests more computational time, it is compensated by the solutions' quality. Regarding the stochastic scenario (which represents the main contribution of this paper), Fig. 3 shows boxplots of the percentage gap of both the stochastic solutions and deterministic solutions for the different levels of uncertainty, with respect to the best deterministic solution. The results show that the solutions provided by our approach for the stochastic version (*OBD – S – x*) clearly outperform the solutions for the deterministic version

when these are simulated ($OBD - D - x$). In other words, near-optimal solutions for the deterministic version of the problem might be sub-optimal solutions for the stochastic version. Hence the importance of integrating simulation methods during the searching process when dealing with stochastic optimization problems. Notice also that the $OBS - D$ value can be seen as a reference upper bound value in a scenario with perfect information (*i.e.*, without uncertainty) for the expected bandwidth under stochastic conditions, while ($OBS - D - H$) can be seen as a lower bound for the expected bandwidth.

7 CONCLUSIONS

Community networks (CNs) are increasingly attracting more attention. They constitute scalable, practical and low cost platforms, which are managed at the edge by volunteers. As their complexity increases, it becomes more relevant to maximize their performance, which depends on the placement of services. It may be a challenging task considering several variables, restrictions, and objectives functions. Typically, the literature simplifies the problem by means of ignoring the stochasticity of the bandwidths among nodes. However, the validity of this simplifications has not been analyzed. In this work, we have presented a simheuristic approach to place services considering the stochasticity. The core of this approach relies on a multi-start metaheuristic. The simheuristic extends this approach by integrating Monte Carlo simulation techniques to generate plenty of scenarios where promising solutions are tested. Computational experiments have been carried out using real traces from *GuifiSants*, a subset of *Guifi.net* CN. These experiments help us to validate our approach and assess the importance of taking into account the stochasticity in CNs.

Several lines of future research stem from this work. First, different CNs could be studied to determine their stochasticity. In addition, different metaheuristics could be applied to compare the quality of the solutions provided. Finally, it would be interesting to consider dependencies among bandwidths and assess scenarios where a crisis that made most resources not available took place.

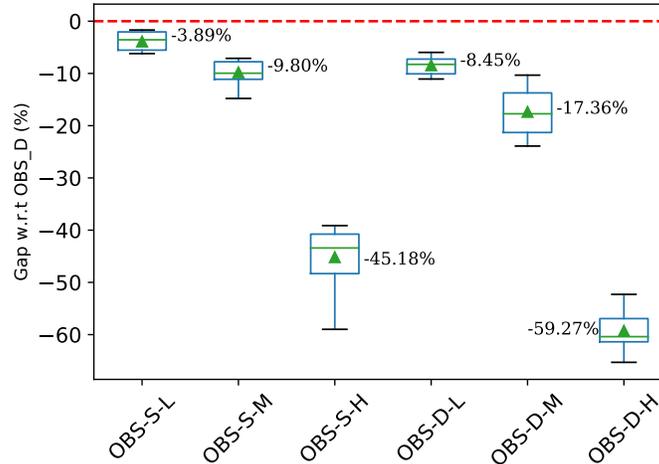


Figure 3: Gaps between approaches.

ACKNOWLEDGMENTS

This work has been partially supported by the Spanish Ministry of Science, Innovation and Universities under contract PGC2018-097599-B-100 and by the Spanish State Research Agency (AEI) under contracts PCI2019-111850-2 and PCI2019-111851-2.

Table 1: Table of results comparing both deterministic and stochastic scenarios.

Instant	Deterministic					Stochastic								Gaps (%)		
	OBD-S [1]	Time (s)	BASP [2]	Time (s)	Gap (%) [1-2]	OBS-S-L [3]	OBS-S-M [4]	OBS-S-H [5]	OBS-D-L [6]	OBS-D-M [7]	OBS-D-H [8]	[3-6]	[4-7]	[5-8]		
11	47.47	27.75	36.22	15.31	31.06	46.47	42.47	19.47	43.41	36.47	16.47	7.05	16.45	18.21		
12	50.21	65.64	43.03	14.84	16.69	49.21	45.21	30.21	47.21	38.21	19.21	4.24	18.32	57.26		
13	54.2	61.6	47.23	14.74	14.76	51.2	50.2	27.2	48.2	44.2	24.2	6.22	13.57	12.40		
14	48.4	43.13	42.5	14.64	13.88	45.4	44.4	26.4	44.4	43.4	18.43	2.25	2.30	43.24		
15	59.28	24.15	41.19	14.22	43.92	58.28	53.28	34.28	53.28	51.28	28.28	9.38	3.90	21.22		
16	50.13	25.33	43.48	14.71	15.29	49.13	46.13	25.13	47.13	43.13	20.13	4.24	6.96	24.84		
17	59.79	22.79	50.21	11.00	19.08	56.79	52.79	31.79	53.79	49.79	26.79	5.58	6.03	18.66		
18	56.22	16.6	36.71	14.70	53.15	54.22	52.22	34.22	52.22	50.22	22.22	3.83	3.98	54.01		
19	54.09	37.54	35.38	15.32	52.88	51.09	46.09	32.09	50.09	43.29	21.09	2.00	6.47	52.16		
110	56.3	25.04	42.59	14.28	32.19	54.3	49.3	33.3	50.3	46.32	22.3	7.95	6.43	49.33		
111	52.98	59.58	44.54	15.37	18.95	49.98	48.98	29.98	48.82	40.98	21.98	2.38	19.52	36.40		
Avg:	52.98	42.1	42.09	14.47	25.87	51.46	48.28	29.46	48.99	44.30	21.92	5.05	8.98	34.41		

REFERENCES

- Alasmar, M., G. Parisi, R. Clegg, and N. Zakhleniu. 2019. "On the Distribution of Traffic Volumes in the Internet and its Implications". In *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*. April 29th-May 2nd, Paris, France, 765-774.
- Badri, H., T. Bahreini, D. Grosu, and K. Yang. 2020. "Energy-Aware Application Placement in Mobile Edge Computing: A Stochastic Optimization Approach". *IEEE Transactions on Parallel and Distributed Systems* 31(4):909-922.
- Cerdà-Alabern, L., R. Baig, and L. Navarro. 2020. "On the Guifi.net community network economics". *Computer Networks* 168:1-12.
- Coimbra, M. E., M. Selimi, A. P. Francisco, F. Freitag, and L. Veiga. 2018. "Gelly-scheduling: Distributed Graph Processing for Service Placement in Community Networks". In *Proceedings of the 33rd Annual ACM Symposium on Applied Computing*. April 9th-13th, Pau, France, 151-160.
- Dimogerontakis, E., J. Neto, R. Meseguer, L. Navarro, and L. Veiga. 2017. "Client-side routing-agnostic gateway selection for heterogeneous Wireless Mesh Networks". In *2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*. May 8th-12th, Lisbon, Portugal, 377-385.
- Ducatel, K., M. Bogdanowicz, F. Scapolo, J. Leijten, and J.-C. Burgelman. 2001. "Scenarios for Ambient Intelligence in 2010". Technical Report, IST Programme Advisory Group (ISTAG), Seville, Spain.
- Even, S. 2011. *Graph Algorithms*. 2nd ed. New York, NY, USA: Cambridge University Press.
- Farhadi, V., F. Mehmeti, T. He, T. L. Porta, H. Khamfroush, S. Wang, and K. S. Chan. 2019. "Service Placement and Request Scheduling for Data-intensive Applications in Edge Clouds". In *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*. April 29th-May 2nd, Paris, France, 1279-1287.
- Herrmann, K. 2010. "Self-organized Service Placement in Ambient Intelligence Environments". *ACM Trans. Auton. Adapt. Syst.* 5(2):6:1-6:39.
- Herrmann, K., K. Geihs, and G. Mühl. 2005. "Ad Hoc Service Grid". In *Mobile Information Systems: IFIP TC8 Working Conference on Mobile Information Systems (MOBIS)*. September 15th-17th, Oslo, Norway, 261-274.
- Jiao, L., J. Li, T. Xu, W. Du, and X. Fu. 2016. "Optimizing Cost for Online Social Networks on Geo-Distributed Clouds". *IEEE/ACM Transactions on Networking* 24(1):99-112.
- Juan, A. A., J. Faulin, S. E. Grasman, M. Rabe, and G. Figueira. 2015. "A review of simheuristics: Extending metaheuristics to deal with stochastic combinatorial optimization problems". *Operations Research Perspectives* 2:62 - 72.
- Juan, A. A., J. Faulin, J. Jorba, D. Riera, D. Masip, and B. Barrios. 2010. "On the use of Monte Carlo simulation, cache and splitting techniques to improve the Clarke and Wright savings heuristics". *Journal of the Operational Research Society* 62:1085-1097.
- Khalajzadeh, H., D. Yuan, J. Grundy, and Y. Yang. 2016. "Improving Cloud-Based Online Social Network Data Placement and Replication". In *2016 IEEE 9th International Conference on Cloud Computing (CLOUD)*. June 27th-July 2nd, Alaska, USA, 678-685.
- Lertsinsrubtavee, A., M. Selimi, A. Sathiseelan, L. Cerdà-Alabern, L. Navarro, and J. Crowcroft. 2018. "Information-Centric Multi-Access Edge Computing Platform for Community Mesh Networks". In *Proceedings of the 1st ACM SIGCAS Conference on Computing and Sustainable Societies*. June 20th-22nd, San Francisco, USA, 1-12.
- Marler, R., and J. Arora. 2004. "Survey of multi-objective optimization methods for engineering". *Structural and Multidisciplinary Optimization* 26(6):369-395.
- Martí, R., M. G. Resende, and C. C. Ribeiro. 2013. "Multi-start methods for combinatorial optimization". *European Journal of Operational Research* 226(1):1-8.
- Panadero, J., L. Calvet, J. M. Marquès, and A. A. Juan. 2017. "A simheuristic approach for resource allocation in volunteer computing". In *Proceedings of the 2017 Winter Simulation Conference (WSC)*, edited by W. K. V. Chan, A. D'Ambrogio, G. Zacharewicz, N. Mustafee, G. Wainer, and E. Page, 184-191. Las Vegas, USA: IEEE - Institute of Electrical and Electronics Engineers, Inc.
- Pasteris, S., S. Wang, M. Herbster, and T. He. 2019. "Service Placement with Provable Guarantees in Heterogeneous Edge Computing Systems". In *INFOCOM 2019 - IEEE Conference on Computer Communications*. April 29th-May 2nd, Paris, France, 514-522.
- Selimi, M., L. Cerdà-Alabern, F. Freitag, L. Veiga, A. Sathiseelan, and J. Crowcroft. 2019. "A Lightweight Service Placement Approach for Community Network Micro-Clouds". *Journal of Grid Computing* 17(1):169-189.
- Selimi, M., D. Vega, F. Freitag, and L. Veiga. 2016. "Towards Network-Aware Service Placement in Community Network Micro-Clouds". In *Proceedings of the Euro-Par 2016: Parallel Processing: 22nd International Conference on Parallel and Distributed Computing*, edited by P.-F. Dutot and D. Trystram, 376-388. Grenoble, France: Springer.
- Shabtai, G., D. Raz, and Y. Shavitt. 2018. "Risk Aware Stochastic Placement of Cloud Services: The Case of Two Data Centers". In *Algorithmic Aspects of Cloud Computing*, edited by D. Alistarh, A. Delis, and G. Pallis, 59-88. Cham, Germany: Springer International Publishing.

- Taimre, T., D. P. Kroese, and Z. I. Botev. 2019. “Monte Carlo methods”. *Wiley StatsRef: Statistics Reference Online* 10.
- Tarneberg, W., A. Mehta, E. Wadbro, J. Tordsson, J. Eker, M. Kihl, and E. Elmroth. 2017. “Dynamic application placement in the Mobile Cloud Network”. *Future Generation Computer Systems* 70(Supplement C):163 – 177.
- Xia, Q., W. Liang, and Z. Xu. 2017. “The Operational Cost Minimization in Distributed Clouds via Community-aware User Data Placements of Social Networks”. *Comput. Netw.* 112:263–278.
- Yan, S., Y. Zhang, S. Tao, X. Li, and J. Sun. 2019. “A Stochastic Virtual Machine Placement Algorithm for Energy-Efficient Cyber-Physical Cloud Systems”. In *2019 International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*. July 14th-17nd, Atlanta, USA, 587-594.
- Zhang, L., and X. Tang. 2014. “The Client Assignment Problem for Continuous Distributed Interactive Applications: Analysis, Algorithms, and Evaluation”. *IEEE Trans. Parallel Distrib. Syst.* 25(3):785–795.

AUTHOR BIOGRAPHIES

JAVIER PANADERO is an Assistant Professor of Simulation and High Performance Computing in the Computer Science Dept. at the Universitat Oberta de Catalunya. He is also a Lecturer at the Euncet Business School. He holds a Ph.D. and a M.S. in Computer Science. His major research areas are: high performance computing and simheuristics. He has co-authored more than 40 scientific articles. His email address is jpanaderom@uoc.edu.

LAURA CALVET is an Assistant Professor of Statistics in the Computer Science Dept. at the Universitat Oberta de Catalunya and a Lecturer in Economics at the Universitat Internacional de València. Her main lines of research are: i) design of optimization algorithms relying on metaheuristics, machine learning and/or simulation applied to sustainable logistics & computing; ii) applied statistics & economics. Her email address is lcalvetl@uoc.edu.

CHRISTOPHER BAYLISS is a post-doctoral researcher in the ICSO group at the IN3 – Universitat Oberta de Catalunya. His main research interests include metaheuristics, simulation optimization, revenue management, packing problems, airline scheduling, and logistics optimization. His email address is cbayliss@uoc.edu.

JOAN MANUEL MARQUÈS is an Associate Professor at Computer Science, Multimedia & Telecommunication Studies at Universitat Oberta de Catalunya since 1997. He graduated as a Computer Science Engineer from the Facultat d’Informàtica de Barcelona (UPC) and a Ph.D. in Computer Science from UPC. His research interests include the design of scalable and cooperative Internet services and applications. He is member of the Association for Computing Machinery. His email address is cbayliss@uoc.edu.

MENNAN SELIMI is an Associate Professor at South East European University, North Macedonia. He is the head of Distributed Systems and Data Science research group at Max van der Stoel Institute. Previously, he was a Research Associate at University of Cambridge working on H2020 UMOBILE and RIFE projects. He has a Phd in Distributed Computing (with Distinction and Honors) from UPC BarcelonaTech and IST Lisbon. His research interests centre on decentralized cloud infrastructures. He is heavily involved in network and system measurements (e.g., wireless mesh networks), looking at topics ranging from blockchain to network economics. His email address is m.selimi@seeu.edu.mk.

FELIX FREITAG is an associate Professor at the Computer Architecture Department (Departament d’Arquitectura de Computadors) at the Technical University of Catalunya (UPC), which he joined in 1999. He obtained a PhD in Telecommunication Engineering from the UPC in 1998. His research interests include the design of scalable decentralized systems. In the recent years his research focused on community clouds. He is member of the IEEE. His email address is felix@ac.upc.edu.