Proceedings of the 2014 Winter Simulation Conference
A. Tolk, S. Y. Diallo, I. O. Ryzhov, L. Yilmaz, S. Buckley, and J. A. Miller, eds.

COMBINING BIASED RANDOM SAMPLING WITH METAHEURISTICS FOR THE FACILITY LOCATION PROBLEM IN DISTRIBUTED COMPUTER SYSTEMS

Guillem Cabrera
Sergio Gonzalez-Martin
Angel A. Juan
Joan M. Marquès

Computer Science Department
IN3 - Universitat Oberta de Catalunya
08018 Barcelona, SPAIN

Scott E. Grasman

Industrial and Systems Engineering Dept.
Rochester Institute of Technology
Rochester, NY 14623-5603, USA

ABSTRACT

This paper introduces a probabilistic algorithm for solving the well-known Facility Location Problem (FLP), an optimization problem frequently encountered in practical applications in fields such as Logistics or Telecommunications. Our algorithm is based on the combination of biased random sampling –using a skewed probability distribution– with a metaheuristic framework. The use of random variates from a skewed distribution allows to guide the local search process inside the metaheuristic framework which, being a stochastic procedure, is likely to produce slightly different results each time it is run. Our approach is validated against some classical benchmarks from the FLP literature and it is also used to analyze the deployment of service replicas in a realistic Internet-distributed system.

1 INTRODUCTION

The Telecommunication and Computing sectors are one of the key role players in current economy. The evolution experienced, with considerable improvements in network infrastructures and capabilities, made both sectors to grow very fast in the last two decades. This growth can be explained both by the advances in telecommunication technologies, the market privatization (Lam and Shiu 2010) and the expansion of the computational infrastructures and applications. All this has brought with it new optimization problems in several related fields.

The Facility Location Problem (FLP) involves locating an undetermined number of facilities to minimize the sum of the setup costs and the costs of serving customers from these facilities. The problem assumes that the alternative sites where the facilities can be located are predetermined and the demand required by each customer is known beforehand. Facility location decisions are costly and difficult to reverse as, once the facility has been installed, the associated opening cost is actually incurred. This problem is useful to model problems in very disparate areas like transportation and logistics, inventory planning, telecommunication network or computational infrastructures planning. Clear examples of its application on Information technologies are the placement of web-servers in a distributed network which have to provide some service to a given set of customers; or the placement of cabinets in optical fiber networks to server all the customer with a minimal network deployment cost.

There is a research trend that promotes the aggregation of non-dedicated computing resources to form multi-purpose computing platforms similar to *clouds*, known as volunteer computing (Anderson 2004), community clouds (Marinos and Briscoe 2009) or contributory communities (Lázaro 2011). In such environments, users freely donate their computing resources to host collective services or store data. The allocation process in the platform has a critical impact on overall system performance and resource usage.

We present in this article a probabilistic algorithm that combines biased randomization —random sampling from skewed probability distributions— with a metaheuristic in order to obtain good-quality solutions for the FLP. We prove the goodness of our methodology in one of the well-established benchmarks of this problem. Once the approach has been validated, we use it to solve a realistic Internet computing scenario. In this scenario, we model the data or application allocation in a community cloud as a FLP on which we are interested on minimizing the network hops from any client to one of the deployed replicas. Our methodology is applied to this scenario to determine where in a network to place services or applications.

The rest of this paper is structured as follows: we briefly present the Facility Location Problem and its related literature in Section 2. The basics of our proposed methodology are introduced in Section 3. Benchmark-based evaluation of our algorithm is provided in Section 4. In Section 5 we apply our methodology to a network use case on which we simulate different service allocation strategies. Finally, Section 6 highlights the main contributions of this work.

2 THE FACILITY LOCATION PROBLEM

The (uncapacitated) FLP involves locating an undetermined number of facilities to minimize the sum of setup and serving costs. It was first described in Balinski (1964) and Stollsteimer (1961), and is considered as the *simple* facility location problem (Verter 2011), where both the alternative facility locations and the customer positions are considered to be discrete points. The problem is defined over an undirected strongly connected graph $G = (V,E)$ where V is composed of a subset of customers $C \subseteq V$ and a subset of facilities $F \subseteq V$, and E is a set of edges connecting the nodes in V. Each edge $e \subseteq E$ has an associated cost of using it $c_{ij} \geq 0$, and for all $i \in F$ we are given a facility opening cost $f_i \geq 0$. Furthermore, for every facility i and customer j we have an associated cost of connecting the customer to the facility $c_{ij} \geq 0, \forall i \in F, j \in C$. Under these circumstances, the objective of the problem is to open a subset of the facilities in F and connect each customer with an open facility, so that the total cost is minimized:

$$Minimize \sum_{j \in C} c_{ij} + \sum_{i \in F} f_i \tag{1}$$

Next, we review some solutions proposed to the problem and its applications to the computing field. For a more extensive literature review on this topic, we refer the reader to (Drezner 1995), (Snyder 2006) and (Fotakis 2011).

2.1 Solutions to the FLP

The FLP has been studied from the perspectives of worst case analysis, probabilistic analysis, polyhedral combinatorial and empirical heuristics (Cornuejols et al. 1990). Despite its NP-hard nature, exact algorithms were also considered on its study.

One of the first works on the FLP was a branch-and-bound algorithm developed by Efroymson and Ray (1966). They used a compact formulation and its linear programming relaxation to solve it by inspection. However, this relaxation does not provide tight lower bounds and is known as a weak formulation. Another of the earliest approaches is the direct search or implicit enumeration method proposed by Spielberg (1969). Schrage (1975) presented a tight linear programming formulation for the location problem different from the one defined by Efroymson and Ray (1966). He applied a specialized linear programming algorithm for variable upper bound constraints. Erlenkotter (1978) presented a dual-based procedure which begins with this tight linear programming formulation but differed from previous approaches by considering a dual objective function. Körkel (1989) presented an improved version of the original Erlenkotter algorithm.

Approximation algorithms are very valuable for a theoretical analysis of the problem, but they are outperformed in practice by more straightforward heuristic. Constructive algorithms and local search methods for this problem have been used for decades, although more sophisticated metaheuristics have been applied. Kuehn and Hamburger (1963) presented a heuristic procedure with a constructive phase followed by an improvement phase. Alves and Almeida (1992) proposed a Simulated Annealing algorithm; Kratica et al. (2001) presented a Genetic Algorithm outperforming previous works; Ghosh (2003) discussed a Neighborhood Search heuristic for the problem; and Michel and Van Hentenryck (2004) defined a simple but robust, efficient and competitive Tabu Search algorithm that used a linear neighborhood and flipped a single facility at each iteration. Resende and Werneck (2006) proposed an algorithm based on the Greedy Randomized Adaptive Search Procedure (GRASP) metaheuristic that combined a greedy construction phase with a local search procedure. It obtained results very close to the best known solution for a wide range of different instance sets.

2.2 FLP applications to telecommunication networks

Recently, the FLP problem found several new applications in digital network design problems. One example is the equipment allocation in Video on Demand (VoD) network deployments (Thouin and Coates 2008). VoD services are complex and resource demanding, so deployments involve careful design of many mechanisms where content attributes and usage should be taken into account. The high bandwidth requirements motivate distributed architectures with replication of content. An important and complicated task part of the network planning phase of these distributed architectures is resource allocation. The growth of peer-to-peer networks and the use of mobile devices for accessing the contents have made the problem even more complex.

Another example of application can be found in (Lee and Murray 2010), where the authors introduce an approach for survivable network design of citywide wireless broadband based on the FLP model. They address how to locate the Wi-Fi equipment to maximally cover the given demand and how to connect Wi-Fi equipment to ensure survivable networking. The Online Facility Location Problem (OFLP) (Meyerson 2001) can model a network design problem in which several servers need to be purchased and each client has to be connected to one of the servers. Once the network has been constructed, additional clients may need to be added to the network. In this case, additional costs will appear into the problem such as the connection cost of connecting a new customer to the cluster and, if additional capacity is required to accommodate the increase of demand, an additional server should be purchased (which means opening an additional facility).

3 METHODOLOGY OVERVIEW

Our probabilistic algorithm combines an Iterated Local Search (ILS) framework (Lourenço et al. 2010) with biased randomization. Random variates generated from skewed probability distributions are used here to introduce a certain degree of randomness inside the local search process. In this hybrid approach, we define a metaheuristic that obtains results not far from optima by itself and then introduce a biased-randomization process to refine the final result of our methodology as explained in Section 3.2. With this randomized process, alternative feasible solutions of a similar quality are quickly generated at each iteration of the algorithm (which can be seen as a simulation process) and, at the end, the best generated outcome is saved —notice, however, that other statistics, such as the sample mean, sample variance or confidence intervals could also be obtained in a similar way as in any other simulation process.

Our approach offers some advantages over other state-of-the-art algorithms, which usually require more cumbersome and time-costly parameter set up. In practice, these approaches are usually harder to implement and often neither reproducible nor applicable to real-life problems. In order to avoid any fine-tuning process and allow our methodology to be used in very different scenarios, we designed a probabilistic algorithm with a single parameter, the one of the Geometric distribution used to introduce biased randomness into the metaheuristic.

3.1 Our randomized algorithm

The main operation of the proposed heuristic method is detailed in Algorithm 1 and works as follows:

1. Generate an initial random solution.
2. Perform an overcoming local-search on the solution and compute the solution total cost.
3. Start an iterative improvement process until the stopping criteria is reached.
   (a) Perturbate the best-known solution by randomly replacing several open facilities by closed ones.
   (b) Apply the local-search procedure and compute the total cost.
   (c) If the new solution improves the best known solution, keep it as best solution.
4. Return the best solution found so far.

Algorithm 1 Pseudocode of the ILS proposal

```
Require: facilities, customers, beta, maxIter
baseSol ← genInitRandSol(facilities, customers, beta)
baseSol ← localSearch(baseSol)
bestSol ← baseSol
nIter ← 0
credit ← 0
while nIter ≤ maxIter do
   newSol ← perturbate(baseSol, beta) {destruction-construction}
   newSol ← localSearch(baseSol)
   delta ← cost(newSol) − cost(baseSol)
   if delta < 0 then
      credit ← −delta
      baseSol ← newSol
      if cost(newSol) < cost(bestSol) then
         bestSol ← newSol
      end if
   else if delta > 0 and credit ≥ delta then
      {acceptation criterion}
      credit ← 0
      baseSol ← newSol
   end if
   nIter ← nIter + 1
end while
return bestSol
```

The initial solution generator chooses a random number p_i between $\frac{|F|}{2}$ and $|F|$ and picks randomly p_i facilities to open. Closing a facility on the solution is computationally cheaper than opening a new facility due to the way FLP costs are calculated. Therefore, starting with a solution with a higher number of open facilities should lead to a less computationally expensive process.

The local search procedure is based only in closing movements. It starts from the current solution and closes one by one each of the open facilities. The facilities to close are selected in a random order among all the open ones. If the new obtained solution has a lower total cost by closing the selected facility, it is actually removed from the solution; otherwise the facility is kept in.

The perturbation operator destructs some part of the solution by removing open facilities and then reconstructs it by opening others. The actual facilities to be closed and to be opened are selected randomly as well. This operator always opens more facilities than the amount of closed ones, so its outputs could be refined by the same local search procedure used on the initial solution, which benefits from the fact that the closing movement is less computationally expensive.

In our case, the stopping criteria for the ILS procedure is a time limit, so we can control the computing time for the algorithm.

3.2 Use of skewed probability distributions to induce biased randomization

Randomization on the selection of facilities in different stages of the procedure allows the algorithm to widely explore the solution space (Juan et al. 2014). We already stated our method uses randomization in the initial solution construction, in the local-search stage and in the perturbation operator. As discussed in Juan et al. (2009), Juan et al. (2011) and González et al. (2012), biased-random sampling has proven to be extremely useful for solving similar problems. While the uniform randomization used in GRASP limits the search space and performs a blind search, biased randomization drives more effciently our ILS metaheuristic to better solutions by exploring widely the solution space. Furthermore, these latter techniques fostered the apperance of new state of the art metaheuristics for several optimization problems (Juan et al. 2011, Juan et al. 2014). To the best of our knowledge, this is the first proposal that uses biased-randomization techniques to efficiently solve the FLP.

As explained above, randomization on the facility selection has to be done without introducing too many parameters in the algorithm in such a way we avoid non-trivial, time-consuming and often instance-dependent fine-tuning processes for input parameters. When compared to GRASP, our approach is not restricting the list of candidates to be evaluated, so we are leaving out one parameter. Furthermore, the randomization is biased, since all the candidate facilities are not considered with the same probability (Figure 1). One of the probability distribution functions which have demonstrated to obtain competitive results when combined with metaheuristics is the single-parameter geometric distribution.

Figure 1: Uniform randomization and biased randomization.

Our algorithm would then obtain new solutions to the FLP problem by introducing changes in the list of open facilities. These changes will be determined by the result of sampling random variates from a geometric distribution, which will add a single input parameter for the algorithm: the beta parameter for the probability distribution. This process is therefore run using oriented random sampling.

4 EVALUATION

To evaluate and assess the performance of the proposed algorithm, several computational experiments were performed. The proposed algorithm was implemented as a Java 7SE application. We carried out all the tests on a commodity desktop computer with an Intel Core i5-2400 processor and 4 GBytes of RAM memory running Ubuntu GNU/Linux 13.04 and the Oracle Java Virtual Machine 7-64bits. Even though Java is a programming language executed in a virtual machine (JVM) and we are aware it may show poorer performance than compiled lanuages, the execution on the JVM offers better replicability and repeatability than other languages (Luke 2009).

To test the efficiency of the proposed algorithm, we ran our algorithm with a well-known benchmark set of the FLP proposed by Bilde and Krarup (1977). This benchmark includes 220 instances divided in 22 subsets. These instances were artificially generated by the authors, selecting the assignment costs randomly on the range [0, 1000], and opening costs being always greater than 1000. Thanks to their reduced scale, optimal solutions for each instance of the problem are provided, so any heuristic can be compared with these optima. While the GRASP by Resende and Werneck (2006) obtained a mean error in the whole set of 0.0015% compared against optimal values, our methodology found the optimal solution in all the instances. We compare in Table 1 the running times for Resende's and our proposals.

Table 1: Results on the Bilde and Krarup benchmark. The time is the spent to find the best solution for that instance.

			Resende and Werneck (2006)	**Our methodology**
Instance	**# facilities**	**# clients**	**time [ms]**	**time [ms]**
B	50	100	310	0.2036
C			450	0.0498
D01	30	80	223	0.0179
D02			211	0.0432
D03			199	0.0627
D04			170	0.0804
D05			162	0.1025
D06			186	0.1236
D07			174	0.1519
D08			166	0.0747
D09			175	0.0736
D10			166	0.2100
E01	50	100	476	0.0343
E02			588	0.0672
E03			512	0.1068
E04			464	0.1360
E05			376	0.1815
E06			408	0.2178
E07			416	0.2520
E08			418	0.2840
E09			352	0.2862
E10			353	0.3180
Average			**316**	**0.1399**

As data in Table 1 states, our methodology outperforms the GRASP by Resende and Werneck (2006). While our method finds the optimal result for all instances, the GRASP method was unable to get to the optima in some of the instances. The computing times were also much shorter (several orders of magnitude shorter). Juan et al. (2014) and (Juan et al. 2011) already showed in metaheuristics for different problems that the usage of skeded probability distributions in the randomization process within the heuristic provides state-of-the-art results in terms of quality and execution times.

5 A REAL USE CASE: MINIMIZING NETWORK DISTANCE TO SERVICES

Network distance between servers and clients has a great impact on the quality perception for some Internet applications and the overall bandwidth consumption. For example, video streaming services might be affected if congestion is found in the path from the server to the final client. Distributing services across the network is a good strategy to reduce these phenomena, but it comes at a high cost for operators.

Data-intensive applications can also suffer from degradation and can generate high bandwidth demands if data is located far from the processing spot. In this field of study, Ryden et al. (2013) proposed a scenario in which user-contributed resources could be gathered to support data-intensive applications. Their proposal included a host selection strategy, based on bandwidth probes, in order to reduce overall bandwidth usage. Also targeting systems composed by non-dedicated resources, Lázaro et al. (2012) proposed an availability-aware host selection policy for service deployment in contributory communities. In these systems, services are hosted by user-donated computers and consumed by others in the platform. From the network perspective, their replica selection strategy would represent a random selection.

Studying which locations should be selected in a network to place a content or a service can be modeled as a FLP or a p-median problem (Resende and Werneck 2004). While the p-median problem does not consider any opening costs and restricts the number of facilities to open, the FLP considers the cost incurred when opening a facility and relates the final number of open ones to the instance size and the opening cost values. In this particular case, the number of replicas each service should deploy is not known in advance. Regarding the network distances, the more service replicas in a network, the closer should be any client to any of them. However, more resources would be utilized and therefore less services could be supported in a platform of the same size.

In order to prove the value of our methodology in a real use case, we simulated a community cloud. To do so, we selected a mesh network topology from a Wireless Community Network (Flickenger 2002). This type of networks are constructed, operated, maintained and owned by the users themselves and pose a great opportunity for community cloud success. We selected a network snapshot with 1,134 nodes and 1,161 links. Due to space limitation and the low visibility of such large graph, we cannot depict the actual network topology in this article. Instead, we provide some basic graph statistics in Table 2 to help the reader better understand the type of network we dealt with.

Table 2: Traits of the studied network graph.

	avg.	**min.**	**max.**
Network diameter			11
Node degree	2.0476	1	260
Node degree centrality	0.0018	0.0009	0.2295
Node closeness centrality	0.2442	0.1212	0.4258
Node betweenness centrality	0.0028	0.00	0.7912
Path lengths	4.2244	1	11

We considered the 53 nodes in the topology with more than one link to be the nodes that could host a service (the facilities in the FLP terminology) and all of them (including the facilities) to be potential service consumers (the clients in the FLP). Considering facilities to be also consumers is one of the features in the community cloud and contributory communities proposals, as users originally contribute their resources to support the platforms with the only reward of accessing these services.

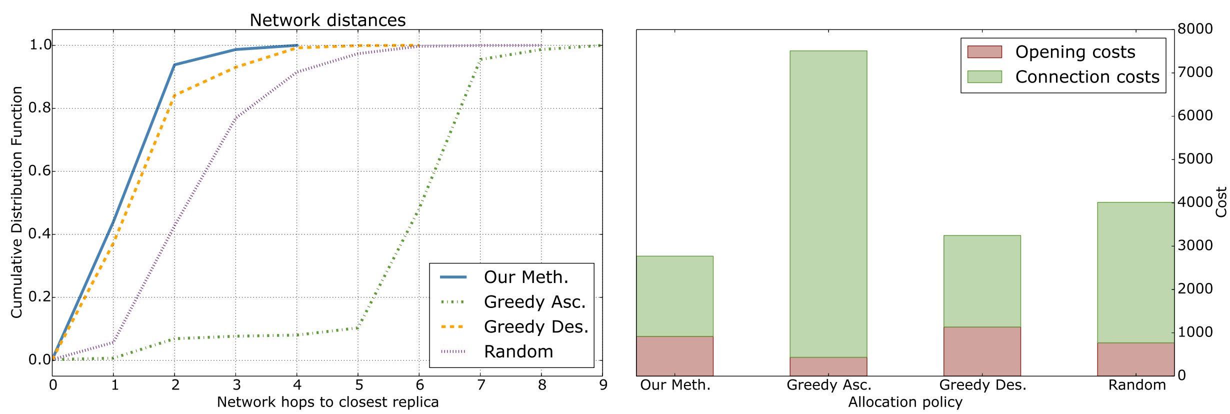
Although in this example we assume complete knowledge about the network topology, probing techniques like *traceroute* could be used in a real system to explore the underlying network connecting the nodes, in a similar way than Ryden et al. (2013) do with available bandwidth used by their allocation strategy.

To transform the network graph into a classical FLP instance, connection costs were established as the number of network hops from one node to another and the opening cost for each facility was defined as $c_i = 1000 \times closeness\_centrality_i$. The closeness centrality is a graph measure on how close is a vertex to all other vertices in a graph. Thus, directly linking the opening cost of a facility at a given spot to this measure intuitively seemed a good strategy to associate higher costs to well connected hosts.

We studied the application of our methodology in the described scenario and compared it to the following allocation strategies:

1. A greedy method that selects the top N nodes ordered by descending opening cost.
2. A greedy method, that selects the top N nodes ordered by ascending opening cost.
3. A random selection of N nodes from all the available facilities.

In all cases, we set N to the same number of nodes selected by our FLP solving method, so the distance and cost comparison is done with the same level of resource usage. For the random selection strategy, we took the average from 100 samples. We gave 1 second of running time to our heuristic, a restricted time that would allow its use in a user-interactive service deployment process. We plot in Figure 2a the distance of all clients in the network to its closest facility when selected with each of the explained methodologies. Figure 2b shows the cost incurred by each of the service allocations.



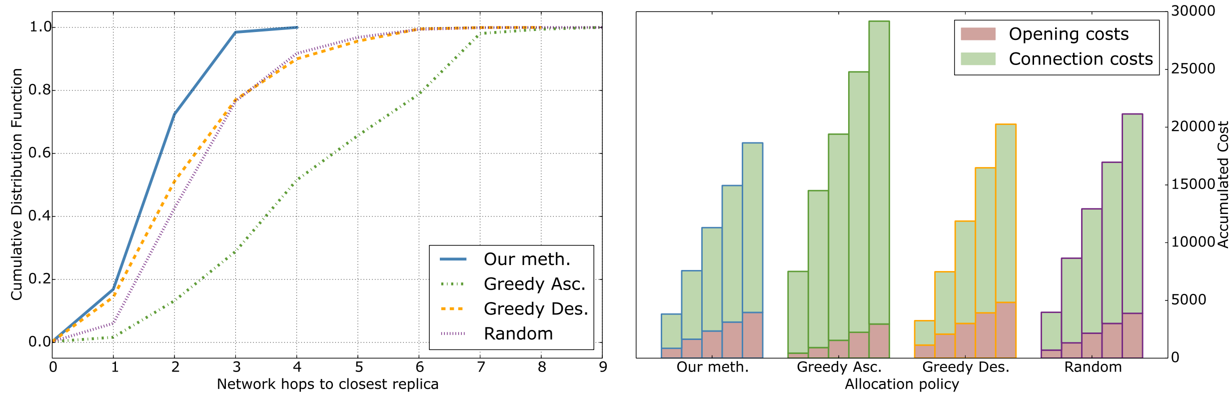
(a) Cumulative Distribution Function of the distance to the closest replica to each client.

(b) Costs comparison of the different deployments obtained by the described allocation methodologies.

Figure 2: Results for the first service allocation.

As can be observed in Figure 2, our simulation-based methodology selects nodes in the network to host a single service that are closer to all other ones while maintaining a low deployment cost. Observing the dispare distances and costs obtained with both greedy methodologies, we can also deduct that (a) selecting those nodes with higher closeness centrality values results in low network distances and restrained total deployment costs; and (b) selecting nodes with lower closeness centrality results in very low opening costs but prohibitively high connecting ones and longer network distances. Both facts are a direct consequence of the cost function used to assign open costs for facilities in the FLP instance, but they help to value our proposal on finding low distance and low cost deployments.

In a real-life scenario, several services or applications should be concurrently supported in the same network. To show the behavior of our methodology in such case, we evaluated the network distances after five consecutive service allocations, considering only one service replica was supported at a time on each host. We show in Figure 3a the distance of each node in the network to the closest facility of each of the five deployed services when replicas are allocated with the explained methodologies. Figure 3b shows the accumulated costs at each of the



(a) CDF of the mean network distances from each node to its closest replica.

(b) Accumulated costs for each of the five deployments with each methodology.

Figure 3: Results for five concurrent service allocations.

Figure 3a shows our methodology consistently allocates replicas closer to all other nodes in the network by using the same number of resources. Specially, if we look at the maximum distance, our proposal is able to allocate replicas at half the distance than other approaches. Thus, we can highlight our methodology does a better resource utilization, getting lower network distances without increasing the number of consumed resources. Moreover, as Figure 3b reflects, the selected service allocations are also cheaper.

6 CONCLUSIONS

In the context of the Facility Location Problem applied to computer networks, this paper discusses the use of random variates generated from skewed probabilty distributions to induce a biased-randomized behavior inside a solving metaheuristic. The use of biased randomization helps the algorithm during the local search, thus providing shorter convergence times than the ones obtained by using standard uniform randomization. The resulting algorithm has been tested against a classical and well-studied benchmark for the FLP. As well, we applied our proposal to a real case scenario of a community platform and showed our methodology regularly selects network locations closer than the ones selected by any of the simple methods compared.

The proposed algorithm while simple, as it is only requiring a single-parameter during the tuning phase, it is obtaining competitive results. These results confirm that simulation-inspired approaches like the one introduced in this proposal can become an efficient tool for solving FLP and other similar optimization problems in the field of distributed computing systems or telecommunication networks. We expect to apply the presented methodology to larger network instances and study its application in real system deployments.

ACKNOWLEDGMENTS

This work has been partially supported by the Spanish Ministry of Science and Innovation (TRA2013-48180-C3-3-P, http://dpcs.uoc.edu), the IN3-UOC Doctoral Grant Program and a RIT International Partnership Development Grant.

REFERENCES

Alves, M., and M. Almeida. 1992. "Simulated annealing algorithm for the simple plant location problem: A computational study". *Revista Investigacao Operacional* 12:1–31.

Anderson, D. P. 2004. "BOINC: A system for public-resource computing and storage". In *Proceedings of the Fifth IEEE/ACM International Workshop on Grid Computing*, 4–10. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Balinski, M. 1964. "On finding integer solutions to linear programs". In *Proceedings of the IBM Scientific Computing Symposium on Combinatorial Problems*, 225–248. Yorktown Heights, New York): IBM.

Bilde, O., and J. Krarup. 1977. "Sharp lower bounds and efficient algorithms for the simple plant location problem". *Annals of Discrete Mathematics* 1:79–97.

Cornuejols, G., G. Nemhauser, and L. Wolsey. 1990. "The uncapacitated facility location problem". In *Discrete Location Theory*, edited by P. Mirchandani and R. Francis, 119–171. New York: Wiley-Interscience.

Drezner, Z. 1995. *Facility location: a survey of applications and methods*. New York: Springer-Verlag.

Efroymson, M., and T. Ray. 1966. "A branch-bound algorithm for plant location". *Operations Research* 14 (3): 361–368.

Erlenkotter, D. 1978. "A dual-based procedure for uncapacitated facility location". *Operations Research* 26 (6): 992–1009.

Flickenger, R. 2002. *Building wireless community networks*. Sebastopol, California: O'reilly.

Fotakis, D. 2011. "Online and incremental algorithms for facility location". *ACM SIGACT News* 42 (1): 97–131.

Ghosh, D. 2003. "Neighborhood search heuristics for the uncapacitated facility location problem". *European Journal of Operational Research* 150 (1): 150–162.

González, S., D. Riera, A. A. Juan, M. G. Elizondo, and P. Fonseca. 2012. "SIM-Randsharp: A hybrid algorithm for solving the ARC routing problem with stochastic demands". In *Proceedings of the 2012 Winter Simulation Conference*, edited by C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose, and A. Uhrmacher, 1–11. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Juan, A., J. Faulin, J. Jorba, D. Riera, D. Masip, and B. Barrios. 2011. "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 (6): 1085–1097.

Juan, A., J. Faulin, R. Ruiz, B. Barrios, M. Gilibert, and X. Vilajosana. 2009. "Using oriented random search to provide a set of alternative solutions to the capacitated vehicle routing problem". In *Operations Research and Cyber-Infrastructure*, edited by J. W. Chinneck, M. J. Saltzman, and B. Kristjansson, Volume 47, 331–346. New York: Springer-Verlag.

Juan, A., H. Loureno, M. Mateo, R. Luo, and Q. Castella. 2014. "Using Iterated Local Search for solving the Flow-Shop Problem: parametrization, randomization and parallelization issues". *International Transactions in Operational Research* 61 (1): 103–126.

Körkel, M. 1989. "On the exact solution of large-scale simple plant location problems". *European Journal of Operational Research* 39 (2): 157–173.

Kratica, J., D. Tošic, V. Filipović, and I. Ljubić. 2001. "Solving the simple plant location problem by genetic algorithm". *RAIRO-Operations Research* 35 (01): 127–142.

Kuehn, A. A., and M. J. Hamburger. 1963. "A heuristic program for locating warehouses". *Management science* 9 (4): 643–666.

Lam, P.-L., and A. Shiu. 2010. "Economic growth, telecommunications development and productivity growth of the telecommunications sector: Evidence around the world". *Telecommunications Policy* 34 (4): 185–199.

Lázaro, D. 2011, July. *A Middleware for Service Deployment in Contributory Computing Systems*. Ph. D. thesis, Universitat Oberta de Catalunya, Barcelona, Spain. Available via http://openaccess.uoc.edu/webapps/o2/bitstream/10609/31021/1/thesis_LazaroIglesias.pdf [accessed July 16, 2014].

Lázaro, D., D. Kondo, and J. Marquès. 2012. "Long-term availability prediction for groups of volunteer resources". *Journal of Parallel and Distributed Computing* 72 (2): 281–296.

Lee, G., and A. T. Murray. 2010. "Maximal covering with network survivability requirements in wireless mesh networks". *Computers, Environment and Urban Systems* 34 (1): 49–57.

Lourenço, H. R., O. C. Martin, and T. Stützle. 2010. "Iterated local search: Framework and applications". In *Handbook of Metaheuristics*, edited by M. Gendreau and J.-Y. Potvin, 363–397. New York: Springer-Verlag.

Luke, S. 2009. *Essentials of metaheuristics*, Volume 3. Raleigh, North Carolina: Lulu.

Marinos, A., and G. Briscoe. 2009. "Community Cloud Computing". In *Proceedings of the 1st International Conference on Cloud Computing*, 472–484. Berlin, Heidelberg: Springer-Verlag.

Meyerson, A. 2001. "Online facility location". In *Proceedings of the 42nd IEEE Symposium on Foundations of Computer Science*, 426–431. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Michel, L., and P. Van Hentenryck. 2004. "A simple tabu search for warehouse location". *European Journal of Operational Research* 157 (3): 576–591.

Resende, M. G., and R. F. Werneck. 2004. "A hybrid heuristic for the p-median problem". *Journal of heuristics* 10 (1): 59–88.

Resende, M. G., and R. F. Werneck. 2006. "A hybrid multistart heuristic for the uncapacitated facility location problem". *European Journal of Operational Research* 174 (1): 54–68.

Ryden, M., A. Chandra, and J. Weissman. 2013. "Nebula: Data Intensive Computing over Widely Distributed Voluntary Resources". Technical Report TR 13-007, University of Minnesota, Minnesota. Available via http://www.cs.umn.edu/tech_reports_upload/tr2013/13-007.pdf [accessed July 16, 2014].

Schrage, L. 1975. "Implicit representation of variable upper bounds in linear programming". In *Computational practice in mathematical programming*, edited by M. L. Balinski and E. Hellerman, 118–132. New York: Springer-Verlag.

Snyder, L. V. 2006. "Facility location under uncertainty: a review". *IIE Transactions* 38 (7): 547–564.

Spielberg, K. 1969. "Algorithms for the simple plant-location problem with some side conditions". *Operations Research* 17 (1): 85–111.

Stollsteimer, J. F. 1961. *The effect of technical change and output expansion on the optimum number, size, and location of pear marketing facilities in a California pear producing region*. Ph. D. thesis, University of California, Berkeley, Berkeley, California.

Thouin, F., and M. Coates. 2008. "Equipment allocation in video-on-demand network deployments". *ACM Transactions on Multimedia Computing, Communications, and Applications* 5 (1): 5.

Verter, V. 2011. "Uncapacitated and capacitated facility location problems". In *Foundations of Location Analysis*, edited by H. Eiselt and V. Marianov, 25–37. New York: Springer-Verlag.

AUTHOR BIOGRAPHIES

GUILLEM CABRERA is a PhD candidate in the Distributed, Parallel and Collaborative Systems (DPCS) research group of the Internet Interdisciplinary Institute (IN3) at the Universitat Oberta de Catalunya (UOC). He holds an MSc in telecommunications from the Universitat Politecnica de Catalunya (UPC). His research interests include scalable and availability-aware distributed mechanisms for computing systems with non-dedicated resources, peer-to-peer systems and optimization of large scale computing infrastructures. His e-mail is gcabreraa@uoc.edu.

SERGIO GONZALEZ-MARTIN is a part time PhD candidate at the Internet Interdisciplinary Institute (IN3) of the Universitat Oberta de Catalunya (UOC) and a software engineer at Roche Diagnostics. He holds an MSc in telecommunications from the Universitat Politecnica de Catalunya (UPC) and a MSc in Free Software from the UOC. His research is focused on Applied Oprimization and Randomized Algorithms for telecommunication infrastructures. His e-mail is sgonzalezmarti@uoc.edu.

ANGEL A. JUAN is an Associate Professor of Applied Optimization, Simulation, and Data Analysis in the Computer Science Department at Universitat Oberta de Catalunya (UOC). He is also a Researcher at the Internet Interdisciplinary Institute. He holds a PhD in Applied Computational Mathematics, a MSc in Information Systems & Technology, and a MSc in Applied Mathematics. His research interests include Applied Optimization and Simulation. Currently, he is the coordinator of the CYTED-IN3-HAROSA@IB Network and editorial board member of both the Int. J. of Data Analysis Techniques and Strategies and the Int. J. of Information Systems & Social Change. He is also a member of the INFORMS society. His website is http://ajuanp.wordpress.com and his e-mail is ajuanp@uoc.edu.

SCOTT E. GRASMAN is a Professor and Department Head of Industrial Systems Engineering at Rochester Institute of Technology. He received his MSc and PhD in Industrial and Operations Engineering from University of Michigan. His primary research interests relate to the application of quantitative models to manufacturing and service systems, focusing on the design and development of supply chains and logistics. His e-mail is scott.grasman@rit.edu.

JOAN M. MARQUÈS is an Associate Professor at Computer Sciences, Multimedia & Telecommunication Studies at Universitat Oberta de Catalunya (UOC) since 1997 and currently is the Director of the Distributed, Parallel and Collaborative Systems research group (DPSC). He graduated as a Computer Science Engineer from the Facultat dInformatica de Barcelona (UPC) in 1991 and received his PhD from UPC in 2003. His research interests include the design of scalable and cooperative Internet services and applications. He is member of the Association for Computing Machinery. His e-mail is jmarquesp@uoc.edu.