A Social-Aware Biased-Randomized Algorithm for Mobile Cloud Computing in Smart Cities

D. Mazza, *Member, IEEE,* A. Pages, *Member, IEEE,* D. Tarchi *Member, IEEE,*
A. Juan, *Member, IEEE,* G.E. Corazza, *Member, IEEE,*

***Abstract*—The Smart City environment is characterized by the presence of a multitude of mobile devices able to connect with remote powerful resources providers in a seamless and ubiquitous way, in order to delegate computing-intensive tasks and massive storage of data. This strategy, commonly called *cyberforaging* or *computation offloading*, aims to shorten execution time, extend battery life, and preserve data storage. It has received a strong impulse by Mobile Cloud Computing (MCC), which enables the mobile users to utilize resources on demand providing a distributed and flexible virtual environment. However, relevant challenges concerning the data-transfer between the mobile devices and the cloud infrastructure have to be taken in consideration, since the transmission needs time and energy for being accomplished. Furthermore, in a typical Smart City ecosystem, where a great affluence of mobile devices aims to perform applications at the same time, a competition for allocating remote resources occurs, becoming a potential cause of delay and energy consumption. Considering these issues, the computation offloading in a Smart City can be seen as a cell association (CA) and resource allocation (RA) problem. In the past, deterministic greedy heuristics have been proposed to solve this problem. In a greedy heuristic, each device tries to maximize its individual utility function by choosing the best network node for the transmission. In this paper, however, we propose a probabilistic algorithm that makes use of biased-randomization techniques in order to enhance the quality of the solution from a *social* or collective point of view. The MCC is exploited for accomplishing applications in a cooperative way, to satisfy the Quality of Service requirements for the maximum number of citizen involved, making effective a vision where Smart City services allow to enhance performance and well-being of the community.**

***Index Terms*—Mobile Cloud Computing, Smart City, Biased-Randomization Algorithm, Heuristics, Computation Offloading.**

I. Introduction

The increasing urbanization has driven the development of technology towards the definition of a Smart City geographic system, characterized by the presence of a multitude of smart devices, sensors and processing nodes aiming to distribute intelligence into the city. An extraordinary phenomenon concerning the information and communication technology (ICT) is happening: smart mobile devices are becoming an essential part of human life and the most effective and convenient communication tools, not bounded in time and place. According to the *Visual Networking Index*[1], the number of mobile-connected devices has already overtaken the number of people in the world, and by 2018 it will be over 10 billion, including machine-to-machine (M2M) modules. Overall mobile data traffic is expected to have nearly an 11-fold increase in the next five years. Moreover, the pervasiveness of wireless technologies has led to the presence of heterogeneous networks where multiple types of access nodes operate simultaneously in the same city area. One of the main challenges in this context is to provide solutions able to optimize jointly the activities of data transfer, exploiting wireless heterogeneous networks (HetNets), and of data processing, delegating computation-intensive tasks to the cloud in a framework of Mobile Cloud Computing (MCC). In fact this *computation offloading* operation allows to tackle with the limited battery power and computation capacity of smart mobile devices (SMDs), and plays a key role in a smart environment where wireless communication is of utmost importance, in determining areas such as mobility and traffic control.

...

In this paper we present a novel algorithm that makes use of biased-randomization techniques [1] to improve the results obtained with a previously developed heuristic [2]. As many other constructive heuristics, in order to generate a 'good' solution, the base heuristic makes use of a 'greedy' behavior which consist of selecting the 'best next step' from a list of potential constructive movements. This selection is based on a certain logic that tries to take advantage of the specific characteristics of the optimization problem being considered. In our base heuristic, each agent tries to make the choice that maximizes his/her individual utility function. However, this might lead to sub-optimal solutions from a collective or social point of view, since individual decisions do not take into account a global perspective. In this context, the main idea behind our approach is to introduce a slight modification in the greedy constructive behavior, in such a way that the constructive process is still based on the heuristic logic but, at the same time, some degree of randomness is introduced. This random effect is introduced throughout the use of a skewed probability distribution: at each step of the constructive process, a different probability of being selected is assigned to each potential movement, being this probability higher as more 'promising' the movement is. As it will be illustrated in the computational experiments section, using this biased-randomized approach allows to easily enhance the quality of the solutions generated by the original heuristic in different dimensions when considering social or collective performance. Also, it is important to notice that if a uniform probability distribution would be used instead of a skewed one, this

[1] http://www.cisco.com/c/en/us/solutions/service-provider/visual-networking-index-vni/index.html

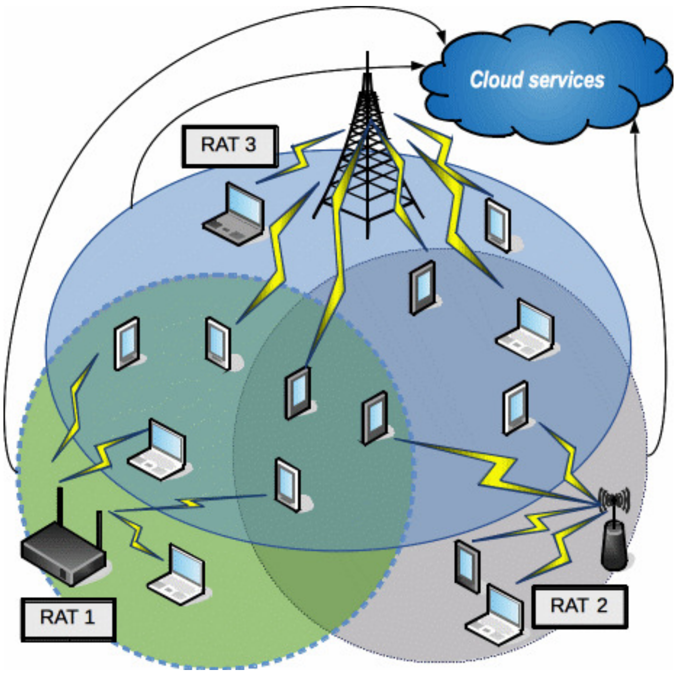

Fig. 1. The reference scenario with different node types in HetNet used by SMDs for offloading applications to MCC

improvement would very rarely occur since the logic behind the constructive heuristic would be destroyed and, accordingly, the process would be random but not correctly oriented.

II. Problem Description

We are focusing on an urban area with a pervasive wireless coverage, where several mobile devices are interacting with a traditional centralized cloud infrastructure and request for services from a remote data center. In order to connect the SMDs to the cloud, we consider the presence of various types of Radio Access Technologies (RATs) that compose the basic elements of the HetNet. The SMDs can connect to the cloud, as shown in fig.1, using the reachable RATs, depending on the availability due to the position and other features we describe below. The presence of different RATs has advantaged cloud computing systems leading to the concept of MCC, where the cloud works as a powerful complement to resource-constrained mobile devices, allowing to delegate storage and computing functions towards the cloud, choosing between different technologies the one that best suits to the contingency of the moment. If the storage is one of the most common and legacy activities that can be transferred to a remote cloud infrastructure, recently, thanks to modern programming paradigms, it is feasible also to allot the entire computation load, or even a part of it, to a remote unit. This allows users to optimize the system performance by offloading just a fraction of the application to be computed, distributing also various parts of the application using different RATs. Multiple radio access technologies, such as IEEE 802.11 WLANs, mobile WiMAX, HSPA+, LTE and WiFi, are being integrated to form a heterogeneous wireless network. For enhancing the network capacity, generally there has been an increasing interest in deploying relays, distributed antennas and small cellular base stations (e.g. macrocells, picocells, femtocells), e.g., indoors in residential homes and offices as well as outdoors in amusement parks and busy intersections. These new network deployments, comprised of a mix of low-power nodes underlying the conventional homogeneous macrocell network, by deploying additional small cells within the local-area range and bringing the network closer to users, can significantly boost the overall network capacity through a better spatial resource reuse. Inspired by the attractive features and the potential advantages of HetNets, their development have gained much momentum in the wireless industry and research communities during the past few years. Examples of heterogeneous elements include microcells, picocells, femtocells, and distributed antenna systems (remote radio heads), which are distinguished by their transmit powers/coverage areas, physical size, backhaul, and propagation characteristics.

The distributed Smart City scenario, indeed, is an environment where the applications requested by the SMDs are partitioned and distributed using the available RATs for the offloading to the cloud. Furthermore, part of the applications can be computed locally by the Requesting SMD (RSMD). In the following part of the paragraph we describe the entities which are involved in this scenario.

1) Cloud: The cloud infrastructure C_{CC} provides the citizens to interact remotely, e.g., for accessing to open data delivered by the public administrations. It refers to the presence of a remote cloud computing infrastructure having a huge amount of storage space and computing power, virtually infinite, offering the major advantage of the elasticity of resource provisioning. The cloud infrastructure is often used for delivering the computing processes to remote clusters, owing a higher computing power, and/or for storing big amount of data. The centralized cloud allows to reduce the computing time by exploiting powerful processing units, but it could suffer from the distribution latency, due to the data transfer from the users to the cloud and *vice versa*, the congestion, due to the multiple users exploitations, and the resiliency, due to the presence of a single performing infrastructure leading to the Single-Point-of-Failure (SPOF) issue. The cloud entity C_{CC} is characterized by its own speed to perform the computation, f_{CC}. The storage availability of C_{CC} can be considered infinite, therefore not constraining in the interaction with the SMDs. Hence, we can write for the centralized cloud:

$$C_{CC} = C_{CC}(f_{CC}) \tag{1}$$

2) HetNet nodes: The wireless HetNet infrastructure is characterized by some RAT nodes, each of them characterized by different features:

- Channel Capacity *BW*: The nominal bandwidth of a certain communication technology that is available for the requesting connecting devices;
- Coverage radius r: the radius that defines the area in which the node can communicate;
- n: The maximum number of devices which is available to connect simultaneously;

- Priority/QoS management: The ability of a certain communication technology to manage different QoS and/or priority levels.

Even if every SMD of the system can simultaneously requests the computation offloading of one or more applications, we are focusing on a scenario where a single application is requested. This simplification does not prevent the generalization of the system, as we can consider the general case as an extension composed by the overlapping of many simplified cases. The requested application *App* is defined through the number of operation to be executed, O, the amount of data to be exchanged, D, and the amount of data to be stored, S.

III. Problem-Related Work

[2]

IV. Our Methodological Approach

A. The Greedy Algorithm

In this sub-section, we present a heuristic algorithm used to resolve the cell association problem following a greedy behavior; the cell association scheme is based on the selection of the *best* node for respecting the requirements of the considered applications; whenever a SMD requests to offload an application, the utility function is evaluated for each access point of the network. The SMD will connect to the node which maximize the utility function.

The selection of a certain access point for establishing the connection could modify the values S_{tr}, E_{part_od} and T_{part_od} for the SMDs already connected with the same access point. Hence, the utility function related to those SMDs is evaluated again, by considering the new incoming SMD. The cell association algorithm is reported in Algorithm 1, where it is possible to note the utility function elaboration and the updating of the utility function for all the SMD already connected to the selected access point.

Algorithm 1 Cell Association Algorithm

```
Greedy Cell Association Algorithm
for all SMD do
    Cell association request by the SMD_j
    for offloading the App_k

    for all RAT_i do
        compute S_tr,ij
        associate SMD_j with RAT_a s.t. S_ajk = max(S_ij) ∀i
        RAT_a.n = RAT_a.n + 1 // update the number of
        SMDs associated to the RAT_a
        for all SMD_h associated to RAT_a do
            compute S_tr,ah
        end for
    end for
end for
```

B. Biased-randomization Algorithm

In the context of combinatorial optimization problems, constructive heuristics use to employ an iterative process in order to construct a 'good' and feasible solution. Examples of these heuristics are the savings procedure for the Vehicle Routing Problem [3], the NEH procedure for the Flow-Shop Problem [4], or the Path Scanning procedure for the Arc Routing Problem [5]. In all these heuristics, a 'priority list of potential movements is traversed during the iterative process. At each iteration, the next constructive movement is selected from this list, which is sorted according to some criteria. The criteria employed to sort the list depends upon the specific optimization problem being considered. Therefore, a constructive heuristic is nothing more than an iterative greedy procedure, which constructs a feasible 'good' solution to the problem at hand by selecting, at each iteration, the 'best' option from a list, sorted according to some logical criterion. Notice that this is a deterministic process, since once the criterion has been defined, it provides a unique order for the list of potential movements. Of course, if we randomize the order in which the elements of the list are selected, then a different output is likely to occur each time the entire procedure is executed. However, a uniform randomization of that list will basically destroy the logic behind the greedy behavior of the heuristic and, therefore, the output of the randomized algorithm is unlikely to provide a good solution. To avoid losing the logic behind the heuristic, GRASP metaheuristics [6] propose to consider a restricted list of candidates -i.e., a sublist including just some of the most promising movements, that is, the ones at the top of the list-, and then apply a uniform randomization in the order the elements of that restricted list are selected. This way, a deterministic procedure is transformed into a randomized algorithm -which can be encapsulated into a multi-start process-, while most of the logic or common sense behind the original heuristic is still respected. Our biased-randomization approach goes one step further, and instead of restricting the list of candidates, it assigns different probabilities of being selected to each potential movement in the sorted list. In this way, the elements at the top of the list receive more probabilities of being selected than those at the bottom of the list, but potentially all elements could be selected. Notice that by doing so, we are not only avoiding the issue of selecting the proper size of the restricted list, but we also guarantee that the probabilities of being selected are always proportional to the position of each element in the list. As a result, each time the randomized algorithm is executed, a new probabilistic solution is obtained (fig.2). Some of these solutions will improve the original one provided by the base heuristic and, moreover, the proposed approach allows to offer alternative solutions to choose from, each of them with different properties.

V. Methodology-Related Work

In general, probabilistic algorithms –similar to the one presented here– have been widely used to solve many combinatorial optimization problems such as, for example: Sequencing and Scheduling Problems [7], Vehicle Routing Problems [8], Quadratic and Assignment Problems [9], Location and Layout

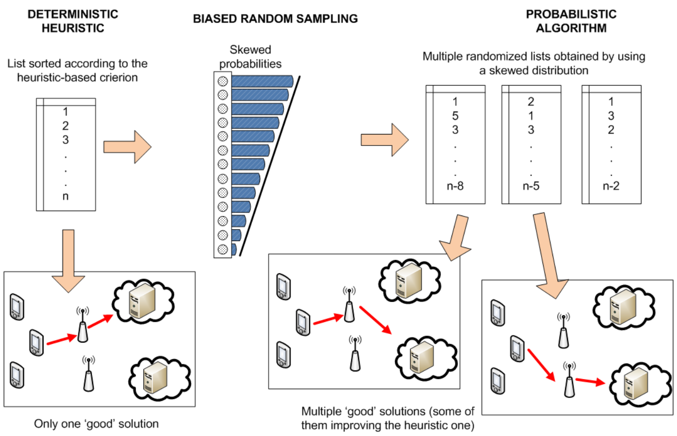

Fig. 2. Scheme of the biased-randomization approach.

Problems [10], Covering, Clustering, Packing and Partitions Problems [11]. In particular, as described in [12], GRASP-like algorithms have been applied to solve a wide set of problems, among others: scheduling, routing, logic, partitioning, location, graph theory, assignment, manufacturing, transportation, telecommunications, biology and related fields, automatic drawing, power systems, and VLSI design. Regarding the use of biased/skewed randomization as proposed in our approach, the SR-GCWS-CS algorithm is proposed in [1] for solving the Capacitated Vehicle Routing Problem. It combines a biased randomization process with a base heuristic. A geometric distribution is used to randomize the constructive process while keeping the logic behind the heuristic. Similarly, in [13] the authors developed the RandSHARP algorithm for solving the Arc Routing Problem. This algorithm combines a savings-based heuristic for the Arc Routing Problem with a biased randomization process also guided by a geometric distribution. Likewise, in [14] the authors propose the ILS-ESP algorithm for solving the Permutation Flow-Shop Problem. The ILS-ESP uses an Iterated Local Search framework and combines the NEH heuristic with a biased randomization process guided by a descending triangular distribution.

VI. Numerical Experiment

We considered a deployment area of $1000 \times 1000\ m^2$, where one LTE eNodeB with channel capacity equal to 100 MHz and three WiFi access point with channel capacities equal to 22 MHz are positioned to cover the entire area. The SMDs are placed in the deployment area according to a randomly uniform distribution. Fig.3 represents the area in case of 500 and 5000 SMD connected, where the access points are positioned at point (0,0), (500,1000) and (0,1000), and the LTE station at (500,500).

The values of S_{md}, P_{id}, P_{tr} and P_l are specific parameters of the mobile devices. We utilized the values of an HP iPAQ PDA with a 400 MHz Intel XScale processor ($S_{md} = 400$) and the following values: $P_l \approx 0.9W$, $P_{id} \approx 0.3W$ and $P_{tr} \approx 1.3W$. As for the cloud server used for the offloading we suppose that S_{cs} = 8000 [15].

The Utility Function utilized in [2] is a weighted sum of three sigmoids functions, that are not independent each other. Thus, we utilized instead, for a comparison of the two algorithms, the throughput function as Utility function. It is a reasonable and feasible way to proceed, because maximizing the throughput is a real system objective (both for the total system and for the single devices). Furthermore, despite the previous study of [2], the capacity constraints on the antennas is considered, putting a limitation on the number of customers (precisely 100 for every RAT). Tab. I shows the comparison among the results of the greedy algorithm and the biased randomization optimization algorithm for increasing numbers of SMDs involved in the system. The biased randomization algorithm solutions are generally better with respect to the original algorithm, in particular for a great number of devices.

VII. Analisys of Results

VIII. Conclusion

Smart Cities...

This paper discusses how biased-randomization techniques can be used to easily improve the performance of already existing or new heuristics aimed at solving combinatorial optimization problems in the field of smart cities and mobile telecommunications. By employing skewed probability distributions, the logic behind the heuristic can be slightly randomized without losing its good properties. This allows to transform the deterministic heuristic procedure into a probabilistic algorithm that can be run several times to obtain several alternative solutions to the original problem, some of them better than the original one provided by the heuristic itself. Some numerical experiments contribute to illustrate the potential of the proposed approach.

Acknowledgment

The authors would like to thank... Also, this work has been partially supported by the Spanish Ministry of Economy and Competitiveness (grant TRA2013-48180-C3-P) and FEDER.

References

[1] A. A. Juan, J. Faulin, J. Jorba, D. Riera, D. Masip, and B. Barrios, "On the use of monte carlo simulation, cache and splitting techniques to improve the clarke and wright savings heuristics." [Online]. Available: http://dx.doi.org/10.1057/jors.2010.29

[2] D. Mazza, D. Tarchi, and G. E. Corazza, "A user-satisfaction based offloading technique for smart city applications," in *Proc. of IEEE Globecom 2014*, Austin, TX, USA, Dec 2014.

[3] G. Clarke and J. W. Wright, "Scheduling of vehicles from a central depot to a number of delivery points," *Operations Research*, vol. 12, no. 4, pp. 568–581, Jul. 1964. [Online]. Available: http://dx.doi.org/10.1287/opre.12.4.568

[4] M. Nawaz, E. Enscore, and I. Ham, "A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem," *Omega*, vol. 11, no. 1, pp. 91–95, 1983. [Online]. Available: http://dx.doi.org/10.1016/0305-0483(83)90088-9

[5] B. Golden, J. Dearmon, and E. Baker, "Computational experiments with algorithms for a class of routing problems," *Computers & Operations Research*, vol. 10, no. 1, pp. 47 – 59, 1983. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0305054883900266

[6] T. A. Feo and M. G. C. Resende, "Greedy randomized adaptive search procedures," *Journal of Global Optimization*, vol. 6, pp. 109–133, 1995. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.48.8667

[7] M. L. Pinedo, *Scheduling: Theory, Algorithms, and Systems*, 3rd ed. Springer Publishing Company, Incorporated, 2008.

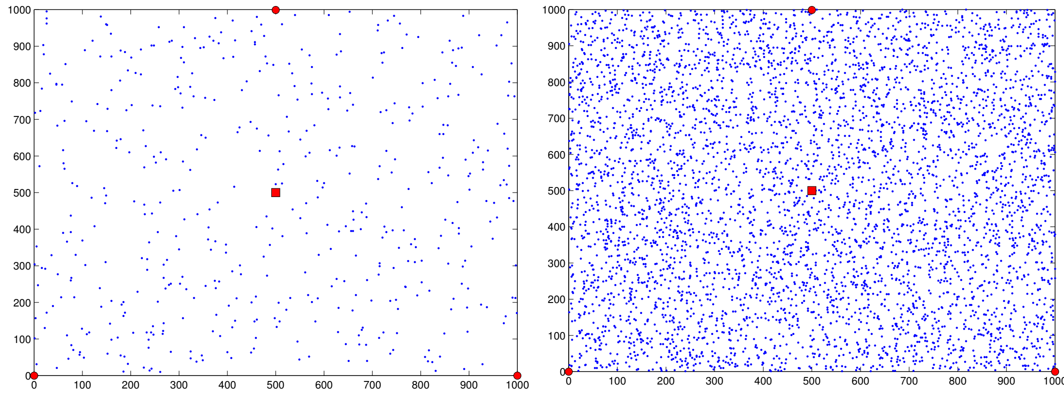

Fig. 3. Area in case of 500 and 5000 SMD connected, where the access points are positioned at point (0,0), (500,1000) and (0,1000), and the LTE station at (500,500)

TABLE I
NUMERICAL RESULTS

SMDs	Throughpt (UF) [Kbps]			Energy Spent [μWs]			Computation Time [s]			Elaboration Time [s]	
N.	GA	BRA	%	GA	BRA	%	GA	BRA	%	GA	BRA
500	15.10	15.88	+5.13	7.34	2.34	-68.16	5.65	1.80	-68.16	0.29	27.32
1000	6.87	8.28	+20.57	3.72	1.65	-55.63	2.86	1.27	-55.62	0.37	51.55
2000	2.18	3.23	+47.75	2.10	0.93	-55.85	1.62	0.71	-55.85	0.56	91.86
5000	1.95	2.36	+21.04	0.69	0.33	-52.41	0.53	0.25	-52.40	0.91	226.74

[8] J. Caceres-Cruz, P. Arias, D. Guimarans, D. Riera, and A. A. Juan, "Rich vehicle routing problem: Survey," *ACM Comput. Surv.*, vol. 47, no. 2, pp. 32:1–32:28, Dec. 2014. [Online]. Available: http://doi.acm.org/10.1145/2666003

[9] E. M. Loiola, N. M. M. de Abreu, P. O. Boaventura-Netto, P. Hahn, and T. Querido, "A survey for the quadratic assignment problem," *European Journal of Operational Research*, vol. 176, no. 2, pp. 657 – 690, 2007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0377221705008337

[10] N. Mladenović, J. Brimberg, P. Hansen, and J. A. Moreno-Pérez, "The p-median problem: A survey of metaheuristic approaches," *European Journal of Operational Research*, vol. 179, no. 3, pp. 927–939, 2007.

[11] A. A. Chaves and L. A. N. Lorena, "Clustering search algorithm for the capacitated centered clustering problem," *Computers & Operations Research*, vol. 37, no. 3, pp. 552–558, 2010.

[12] P. Festa and M. G. Resende, "An annotated bibliography of grasp–part ii: Applications," *International Transactions in Operational Research*, vol. 16, no. 2, pp. 131–172, 2009.

[13] S. González-Martín, A. A. Juan, D. Riera, Q. Castellà, R. Muñoz, and A. Pérez, "Development and assessment of the sharp and randsharp algorithms for the arc routing problem," *AI Commun.*, vol. 25, no. 2, pp. 173–189, Apr. 2012. [Online]. Available: http://dl.acm.org/citation.cfm?id=2350156.2350158

[14] A. A. Juan, H. R. Loureno, M. Mateo, R. Luo, and Q. Castella, "Using iterated local search for solving the flow-shop problem: Parallelization, parametrization, and randomization issues," *International Transactions in Operational Research*, vol. 21, no. 1, pp. 103–126, 2014. [Online]. Available: http://dx.doi.org/10.1111/itor.12028

[15] K. Kumar and Y.-H. Lu, "Cloud computing for mobile users: Can offloading computation save energy?" *IEEE Computer*, vol. 43, no. 4, pp. 51–56, Apr. 2010.