# Supporting Mobile Cloud Computing in Smart Cities via Randomized Algorithms

Daniela Mazza, Adela Pagès-Bernaus, Daniele Tarchi, *Senior Member, IEEE,* Angel Alejandro Juan, Giovanni Emanuele Corazza, *Senior Member, IEEE*

***Abstract*—Smart cities represent rich and dynamic environments in which the cities of the future can be designed by taking advantage of recent advances in information and communication technologies. In these cities, a multitude of smart mobile devices (SMDs) interact among them by sharing data, telecommunication systems, and computing resources. These SMDs require from fast access to data and online services, but at the same time they offer limited computing capabilities, battery lifetime, and data storage capacities. To cope with these constraints, SMDs make frequent use of *computation offloading*, i.e., delegating computing-intensive tasks to the cloud instead of performing them locally. However, in such a large-scale and dynamic environment, there might be thousands of SMDs simultaneously executing processes and, therefore, competing for the allotment of remote resources. This arises the need for a smart allocation of resources, such in a way that greedy behaviors are substituted by more efficient approaches. Accordingly, this paper proposes a probabilistic optimization algorithm that relies on the use of biased (oriented) randomization techniques to support efficient and fast allocation of resources. According to our experimental results, the proposed algorithm is able to provide 'real-time' near-optimal solutions that outperform solutions obtained through existing greedy heuristics. Furthermore, it overcomes the responsiveness limitations of exact optimization methods, which cannot be used in real-life practice since they require excessive computing times.**

***Index Terms*—Smart City, Mobile Cloud Computing, HetNets, Computation Offloading, Biased-randomized Algorithms.**

## I. Introduction

According to the flagship publication of the United Nations *World Urbanization Prospect* [1], more than one half of the world population is currently living in urban areas, and it is expected that about 70% of global population will be city inhabitants by 2050. At the same time, the *Visual Networking Index* [2] outlines that mobile-connected devices have already outnumbered the amount of people in the world. Including machine-to-machine (M2M) modules, there will be over 10 billion of such devices by 2018. Furthermore, it is expected an 11-fold increase in the overall mobile data traffic over the next five years.

Innovative smart city projects have been adopted in the political agenda of many governments as a key program to enable a vision where municipalities can use technology to meet sustainability goals, boost local economies, and improve urban services. These projects focus on strengthening network infrastructures, optimizing traffic and transportation flows, decreasing energy consumption, and supplying innovative services [3], [4]. In addition, a *green* concept of technology, intended to mitigate and reverse the effects of human activity on the environment cannot be neglected in a sustainable city management [5].

The increasing trends of both urbanization and mobile connected people have driven the society towards the definition of a geographic information system for smart cities that is characterized by the presence of a multitude of smart devices, sensors and processing nodes aiming to improve urban services.

Smart mobile devices (SMDs) are becoming an essential part of social life and the most effective and convenient communication tool, which is not bounded in time nor place. In recent years, the increasing request of ubiquitous services has also led to a sharp increase in the so-called cloud computing, where remote servers can be exploited for remote storage and processing services. This is even more important in smart city scenarios: due to their complexity related to the number of users, heterogeneous services, and specific requirements, smart cities can significantly benefit from a cloud computing infrastructure as it can be used to improve system performance and the battery life of user devices through decreased work loads. Indeed, it is through information and communication technologies (ICTs) that smart cities are truly turning *smart* [6], in particular by exploiting SMDs, which together with cloud computing constitute the mobile cloud computing (MCC) framework [7]. A third element is the wireless network, allowing to link resource-constrained devices to centralized large data centers located inside the cloud.

As suggested by Michael Batty, *to understand cities we must view them not simply as places in space but as systems of networks and flows* [8]. Thus, we can consider the MCC as a framework that is the technological nervous system allowing the networks and flows of the city to achieve a better urban way of life. 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 [9], [10].

In this context, one of the main challenges is to provide solutions able to jointly optimize the activities of data transfer (by exploiting wireless heterogeneous networks or HetNets), and data processing (by delegating computing-intensive tasks to the cloud in the MCC framework) [11], [12]. This strategy, commonly called *cyberforaging* or *computation offloading*, allows to overcome SMD limitations concerning limited battery

This work has been partially supported by the Spanish Ministry of Economy and Competitiveness (grant TRA2013-48180-C3-P) and FEDER.

D. Mazza, D. Tarchi and G. E. Corazza are with Dept. of Electrical, Electronic, and Information Eng., University of Bologna, Italy

A. Pagès-Bernaus and A. A. Juan are with Computer Science Dept., Open University of Catalonia - IN3, Spain

power and computational capacities. Moreover, it plays a key role in a smart environment where wireless communication is of utmost importance, particularly in mobility and traffic control domains [13].

Although computation offloading can significantly increase data processing capabilities of mobile users, it is challenging to achieve an efficient coordination among the entire set of requesting devices. In fact, and in order to make the operation possible, the computation offloading entails data transmission among different devices and the cloud. If a large number of citizens use the wireless resource to delegate computation to the cloud, the efficiency of the access node can be severely affected: its channel capacity has to be shared among all the devices, leading to a reduced throughput for each user. This can lead to a crucial increase in energy-usage and data transmission times, representing an inefficient use of offloading operations [14].

In this paper we present a probabilistic algorithm based on biased-randomization techniques [15] to solve the link selection problem. In this problem setting, the most promising node allowing the potential increase in system efficiency has to be selected. The biased-randomization techniques work by introducing a biased or oriented random effect on the possible solutions of a problem, allowing to choose the best solution from a set of possible alternatives that are close to the global optimal. Hence, a set of alternative solutions is generated in short computational times.

We will compare the solution provided by our probabilistic method with the optimal solution provided by exact methods. However, these methods cannot be used in practice since they require excessive computation times. Furthermore, we compare our approach to a greedy algorithm [16] that implements a selfish behavior in allocating resources to users. Such three techniques –the proposed probabilistic algorithm, the greedy heuristic, and the exact method– are based on the definition of a proper objective function taking into account different requirements and characteristics of the considered scenario. In particular, this paper will focus on energy consumption and time delays as key performance indicators (KPIs) for comparing the performance of the three approaches.

The proposed randomized algorithm allows to achieve similar solutions to those provided by the optimal method, although in much less computing time. Moreover, the proposed randomized algorithm clearly outperforms the greedy heuristic while employing similar computing times once it is parallelized, given that the modifications added do not increase the computational complexity of the greedy heuristic. These characteristics make it an efficient alternative for real-life applications.

All in all, in this paper we propose a model establishing the relationship among all the devices (the cloud infrastructure, HetNet nodes, and Smart Mobile Devices), both in terms of computing and communication. From this model a cost function is derived. The goal is to minimize the overall energy consumption by selecting the best communication link for each SMD. We show that this cost function cannot be optimally-solved promptly, given that we have to apply it in real time. Therefore, we present a biased-randomized algorithm that allows to find a near-optimal solution in short computing times. Thus, our proposed approach allows to reach near-optimal solutions in 'real time'. Moreover, we show that it outclasses a greedy heuristic solution, used as a benchmark, presented by the same authors in [16] and summarized in this paper for completeness.

The remainder of the paper is organized as follows. We first analyze the problem-related works in Section II, and describe the problem and the optimization model in Section III. In Section IV the biased randomization techniques are described. Then, in Section V, we present the numerical results. Finally, in Section VI, the main findings and conclusions are highlighted.

## II. Related Works

The cyberforaging mechanism for achieving efficient computation offloading for mobile cloud computing has been discussed in the literature. In [14] a game theoretic approach is proposed, showing that the game always admits a Nash equilibrium. In [10] the authors propose a scheme for optimizing the QoS-aware energy efficiency of the Base Stations composing the environment (macro and small cells), and in [17] the authors investigate an energy-efficient offloading policy for transcoding as a service (TaaS) in a generic mobile cloud system. The cloud system consists of a dispatcher at the front end, and a set of service engines at the back end. In [18] the authors exploit opportunistic communications to facilitate information dissemination in the emerging mobile social networks (MoSoNets). They propose three algorithms for selecting the target set with only $k$ users, for minimizing the mobile data traffic over cellular networks.

In the literature there are several works focused on the offloading operation from the point of view of a single mobile user, e.g., in [19], [20]. In addition, the focus of several works concerning computation offloading is related to the development of application models aiming to extract offloading friendly parts of codes from existing applications [21]–[23]. In [24] the authors present a dynamic offloading algorithm based on Lyapunov optimization for selecting the software components to be remotely executed given an available wireless network connectivity.

However, none of the above explicitly considers the interaction among users. This paper addresses this problem by considering entire populations instead of each user individually, aiming at reducing overall 'community' or social costs. In contrast, in [16] the same authors proposed a utility function model derived from economics, in which the greedy heuristic summarized in Section IV-B was used for minimizing the collective cost for a community in mobile clouds. Our approach in this paper is focused on a global perspective, corresponding to a centralized vision that is more effective for a community. Hereby, the same application (viewed as an entity made out of data and operation) is performed by many users.

With respect to biased randomization techniques, probabilistic algorithms –similar to the one presented here– have been widely used to solve many combinatorial optimization problems such as sequencing and scheduling problems [25], vehicle

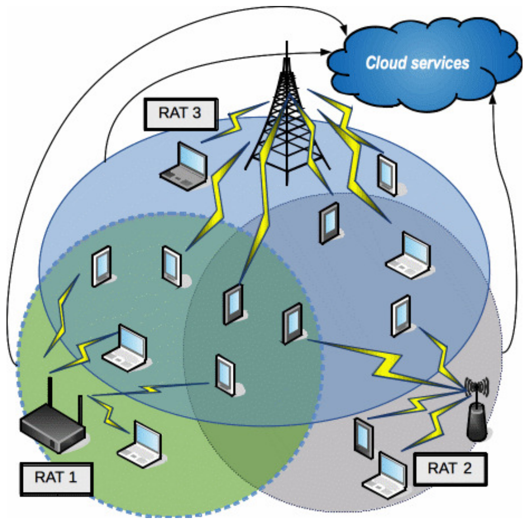



Fig. 1. The reference scenario with multiple SMDs in a multi-RAT environment.

routing problems [26], quadratic assignment problems [27], location and layout problems [28], covering, clustering, packing, and partitions problems [29].

In particular, as described in [30], greedy randomised adaptive search procedure (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 very large scale integration (VLSI) design.

Regarding the use of biased or skewed randomization, the SR-GCWS-CS (SimoRouting Generalized Clarke and Wrights Savings with Cache and Splitting) algorithm is proposed in [15] 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 [31] 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 [32], the authors propose the Iterated Local Search Efficient, Simple, and Parameter-Free (ILS-ESP) algorithm for solving the permutation flow-Shop problem. The ILS-ESP uses an iterated local search framework and combines the Nawaz, Enscore, and Ham (NEH) heuristic [33] with a biased randomization process guided by a descending triangular distribution.

## III. Problem Description and Optimization Model

In this Section, the optimization problem is introduced. The focus is on an urban area with a pervasive wireless coverage, where several SMDs require cloud services. Each SMD can connect to the cloud after selecting one of the available HetNet radio access technologies (RATs), as shown in Fig. 1.

The SMDs aim at delegating storage and computing tasks to overcome their resource constraints and to decrease computational times. Beyond data storage, that is one of the most common activities transferred to remote cloud infrastructures, it is now possible to distribute the whole computational load, or just a fraction, to a remote unit.

Multiple radio access technologies, such as IEEE 802.11 WLANs, mobile WiMAX, HSPA+ and LTE, are being integrated to create a heterogeneous wireless network. Different strategies can be carried out in order to increase the network capacity, such as deployment of relays, distributed antennas, and small cellular base stations (e.g., microcells, picocells, femtocells). These deployments have occurred both indoors (e.g., in residential homes and offices) and outdoors (e.g., in amusement parks or traffic intersections).

Such network deployment is typically composed 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 the users, overall network capacities can be significantly boosted through a better reuse of the spatial resources.

Inspired by the attractive features and the potential advantages of the HetNets, their development has gained much momentum in the wireless industry and research communities in recent years.

The distributed smart city scenario is also an environment where the SMDs can choose and exploit different available RATs for transmitting the application data and for offloading towards the cloud. However, the application can be computed locally by the requesting SMD if this operation is not convenient. The proposed model focuses on direct connections between a RAT and multiple SMDs, considering the availability of direct interaction in a pervasive environment (note that multi-hops links are not taken into account in this paper). Furthermore, we consider energy savings from the user-perspective, differently from other application-focused works, which include the energy consumed by the cloud when sending the resultant data to the SMDs [34]. In the following, the entities involved in this scenario are introduced.

*1) Cloud:* The cloud infrastructure $C_{cc}$ refers to the presence of a remote cloud computing infrastructure with a huge amount of storage space and computing power. It allows the citizens to interact remotely, e.g., for accessing open data delivered by the public administration. The cloud infrastructure is often used for delivering the computing processes to remote clusters, leading to a higher computing power, and/or for storing big amounts of data. The centralized cloud allows to reduce the computational 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 exploitation–, 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 computational speed, $f_{cc}$. The storage availability of $C_{cc}$ can be considered infinite, thus it does not represent a constraint in the interaction with the SMDs. Hence, it is possible to write that the $C_{cc}$ is a function of $f_{cc}$:

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

*2) HetNet nodes:* The wireless HetNet infrastructure is composed of several *RAT* nodes, characterized by different features:

- *BW*: the nominal bandwidth of a certain communication technology that is available for the SMDs;
- $n$: the number of devices connected to the RAT node;
- $pos_{RAT}(c_x, c_y)$: the spatial position of the RAT node, where $c_x$ and $c_y$ represent the coordinates.

Thus, the generic *RAT* is a function depending on the above features:

$$RAT = RAT(BW, n, pos_{RAT}(c_x, c_y)) \tag{2}$$

*3) Applications:* Even if all the SMDs can simultaneously request the computation offloading of one or more applications, we focus on a scenario where each SMD request is restricted to a single application that is characterized by the number of operations to be executed, $O$, and the amount of data to be exchanged, $D$. We neglect other parameters usually considered in more application-focused studies (e.g., Mobibyte [23]) because our study is mainly based on the communication links optimization, aside from the application partitioning. This simplification does not prevent the generalization of the system, since the general case in which a user requests several applications at a given time can be seen as aggregation of several single-application requests. Hence, it is possible to express a generic application *App* as a function of $O$ and $D$:

$$App = App(O, D) \tag{3}$$

*4) SMDs:* Each SMD is characterized by the following features which influence the performance of the computation offloading:

- $P_{tr}$: the power consumed by the SMD to transmit data to the RAT node
- $P_l$: the power consumed by the SMD for local computation
- $P_{id}$: the power consumed by the SMD in waiting mode while the computation is performed in the cloud
- $f_{md}$: the speed to perform the computation locally
- *SNR*: the signal to noise ratio at the receiver side of the link between the SMD and the selected RAT node
- $pos_{md}(c_x, c_y)$: the spatial position of the SMD, where $c_x$ and $c_y$ represent the coordinates.

Thus, for a generic smart mobile device *SMD* we can write:

$$SMD = SMD(P_{tr}, P_l, P_{id}, f_{md}, SNR, pos_{md}(c_x, c_y)) \tag{4}$$

Given these entities, with the respective features, and after having defined as $d_{i,j}$ the distance between the $i$-th RAT and the $j$-th SMD, we can resort to the Shannon formula for evaluating the throughput $S_{i,j}$ of the link between the $i$-th RAT and the $j$-th SMD, where $SNR_j$ is a reference Signal to Noise Ratio (SNR) value at the receiver side and $k$ is an attenuation factor due to the signal propagation:

$$d_{i,j} = |pos_{RAT,i}(c_x, c_y) - pos_{md,j}(c_x, c_y)| \tag{5}$$

$$S_{i,j} = \frac{BW_i}{n_i} \log_2 \left(1 + SNR_j e^{-kd_{i,j}}\right) \tag{6}$$

Among different parameters that can affect the system performance, we will focus on the energy consumption, which is very important when considering a scenario with mobile devices. The energy consumed by the $j$-th *SMD* is composed by the energy spent in transmitting the application data, the energy consumed in idle –while the application is computed on the cloud server–, and the energy consumed for the local computation.

Considering an overall number of $N$ devices and a total number of $M$ RAT nodes, the energy spent by the $j$-th SMD for offloading through the $i$-th *RAT* is:

$$E_{i,j} = \frac{P_{tr\,j}\, D}{S_{i,j}} + \frac{P_{id\,j}\, O}{f_{cc}} \quad i = 1, 2, \ldots, M; j = 1, 2, \ldots, N. \tag{7}$$

In case of a local computation at the $j$-th SMD, its consumed energy is:

$$E_{0,j} = \frac{P_{l\,j}\, O}{f_{md\,j}} \quad j = 1, 2, \ldots, N; \tag{8}$$

where the index 0 stands for the local computation.

In classical approaches, the SMD computing an application *App* has to evaluate the computation offloading benefit and select the most convenient RAT aiming at minimizing the energy consumption of the user requesting the service. This decision is 'selfish' or greedy in the sense that it does not take into account the current and future needs of other users.

In contrast to this, we propose to optimize the global energy consumption, assuming an environment where the devices of the set $\mathcal{N} = \{1, 2, \cdots, j, \cdots, N\}$ are requesting to offload an application using the nodes of set $\mathcal{M} = \{0, 1, 2, \cdots, i, \cdots, M\}$, where the 0-th element represents the fictitious node related to the local computation. By defining $x_{i,j}$, as a binary variable that stands for the presence of a communication link between the $\text{SMD}_j$ and the $\text{RAT}_i$, and $y_j$, as a binary variable accounting for the local computation performed by $\text{SMD}_j$, it is possible to write the energy consumed by $\text{SMD}_j$ when offloading through the $\text{RAT}_i$ in (7), by exploiting (6), as:

$$E_{i,j} = x_{i,j} \left( \frac{P_{tr\,j}\, D}{\frac{BW_i}{n_i} log_2 \left\{1 + SNR_j e^{-kd_{i,j}}\right\}} + \frac{P_{idj}\, O}{f_{cc}} \right) + y_j \left( \frac{P_{l\,j}\, O}{f_{md\,j}} \right) \tag{9}$$

where:

$$x_{i,j} = \begin{cases} 1 & \text{if SMD}_j \text{ is assigned to RAT}_i \\ 0 & \text{otherwise} \end{cases},$$

$$y_j = \begin{cases} 1 & \text{if SMD}_j \text{ computes } App \text{ locally} \\ 0 & \text{otherwise} \end{cases},$$

and $n_i$ is the number of SMDs connected to $\text{RAT}_i$, defined as

$$n_i = \sum_{l \in \mathcal{N}} x_{i,l}.$$

Rearranging the constant terms and the variable terms it is possible to write:

$$E_{i,j} = x_{i,j}\left\{K^{tr}_{i,j}\sum_{l\in\mathcal{N}} x_{i,l} + K^{id}_{j}\right\} + y_j E_{0,j} \tag{10}$$

where the two constants are

$$K^{tr}_{i,j} = \frac{P_{tr\,j}\,D}{BW_i log_2\left\{1 + SNR_j e^{-kd_{ij}}\right\}}$$

and

$$K^{id}_{j} = \frac{P_{id\,j}\,O}{f_{cc}}.$$

Therefore, the optimization problem can be expressed as:

$$\min_{x,y} Z = \sum_{\substack{i\in\mathcal{M}\\ j\in\mathcal{N}}} x_{i,j}\left\{K^{tr}_{i,j}\sum_{l\in\mathcal{N}} x_{i,l} + K^{id}_{j}\right\} + \sum_{j\in\mathcal{N}} y_j E_{0,j} \tag{11a}$$

$$\text{s.t.} \sum_{i\in\mathcal{M}} x_{i,j} + y_j = 1 \qquad j\in\mathcal{N} \tag{11b}$$

$$x_{i,j}, y_j \in \{0,1\} \qquad i\in\mathcal{M} \qquad j\in\mathcal{N} \tag{11c}$$

The objective function (11a) minimizes the overall energy spent, while ensuring that all devices are either assigned to a RAT or leaving the computations to be done locally (11b).

## IV. Optimal and Heuristic Solutions

In this section we present different methods to find an optimal or near-optimal solution to the previously described optimization problem. First, we use an off-the-shelf commercial solver able to provide optimal solutions. However, for real large-size instances it requires unsuitable computation times. Moreover, since we take into account mobile devices whose position can change, the objective function has to be solved repeatedly. Given that the selection of the RAT by the SMDs has to be done in a very short time, we propose to use a fast probabilistic procedure. Our algorithm makes use of biased (oriented) randomization, and it allows us to obtain near-optimal solutions in real-time, which represent a significant improvement over the previously proposed greedy algorithm [16] summarized below.

In order to evaluate the effectiveness of our approach, the performance is compared with the optimal solution. Note that the global optimum can only be found *ex-post*, i.e., once all the SMDs requests have been set. Thus, since the global optimization needs to know the position and the requests of all the SMDs, it cannot be used for online allocation of resources. This approach assumes that all users' requests are processed at the same time –or, alternatively, that the allocation decision is postponed until all requests have been received–, which in practice might require customers to wait a few seconds for their requests to be served.

In contrast, the heuristic method makes use of a 'greedy' behavior, consisting on the selection of the 'most promising' next step from a list of potential constructive movements [16]. This allows to find a good solution in real time while each agent tends to maximize his/her individual utility function. The biased randomized method takes into account a solution from a collective or social point of view by simultaneously considering a set of users' demands altogether instead of one user's demand at a time.

### A. Optimal Solution

The mathematical model (11) can be seen as a particular case of the quadratic semi-assignment problem (QSAP) which is NP-hard [35]. A first approach to find the solution is to use the global optimizer Branch-And-Reduce Optimization Navigator (BARON) [36]. This solver combines constraint propagation, interval analysis, and duality with advanced branch-and-bound optimization concepts.

Note that the QSAP solution not only needs to know the position of all the SMDs requests before assigning them to an antenna, but it also takes longer time than available to quickly serve SMD requests, as shown in Section V. The computational difficulty for solving the QSAP comes from the fact that the objective function is quadratic, which justifies the use of fast heuristic methods. Therefore the optimal solver is only taken as a reference to compare heuristic procedures (which can do the assignment dynamically or following a *wait-and-go* approach) with an optimal solution from the performance point of view.

### B. The Greedy Heuristic

A heuristic algorithm can be used to solve the link selection problem following a greedy behavior [16]. If the offloading operation is advantageous with respect to the local computation, the link selection scheme allows to select the 'most promising' next node from the list of the available ones. As explained in [16], this list is completed for each SMD, which sorts each possible access node based on a self calculated objective function. Since the requests of the SMDs appear in time sequence, the cost function is evaluated on the basis of the current situation, *i.e.*, considering only the previous requests. If the offloading cost is lower than the cost for the local computation, the SMD will connect to the node which minimizes the cost function, otherwise it will compute the application locally.

The selection of the $i$-th node for connecting the $j$-th SMD modifies the values of the throughput $S_{i,k}$ and consequently the energy $E_{i,k}$ for the SMDs already connected with the same node, i.e., for $k = 1, 2, \ldots, j-1$. Thus, this strategy, reported in Algorithm 1, does not take into account any forecast of future connections, leading to a feasible but sub-optimal solution.

### C. Biased-randomized Algorithm

By exploiting biased-randomization techniques [15], the proposed probabilistic algorithm is able to find near-optimal solutions in 'real time' a few seconds or even milliseconds if parallel strategies are employed. Thus, this approach outperforms by far the solution provided by the greedy approach, while approaching the optimal solution in significant lower time than the optimal method. The main idea behind the proposed approach is to introduce a slight modification in

**Algorithm 1** Greedy Link Selection Heuristic

**Inputs:**
$\mathcal{M} = \{0, 1, 2, \cdots, M\}$ // $M$ number of RAT nodes
$\mathcal{N} = \{1, 2, \cdots, N\}$ // $N$ number of SMDs
$RAT_i = RAT(BW_i, n_i, pos_{RAT,i}(c_x, c_y))$ $i \in \mathcal{M} \setminus \{0\}$
$App = App(O, D)$
**Output:**
$X = (x_{i,j})$
$Y = (y_j)$
**Initialization** :
$RAT_i.n \leftarrow 0 \; \forall i \in \mathcal{M} \setminus \{0\}$
$x_{i,j} \leftarrow 0 \; \forall i \in \mathcal{M} \; \forall j \in \mathcal{N}$
$y_j \leftarrow 0 \; \forall j \in \mathcal{N}$
**for** $j = 1$ to $N$ **do**
  $SMD_j = SMD(P_{tr,j}, P_{l,j}, P_{id,j}, f_{md,j}, SNR_j, pos_{md,j}(c_x, c_y))$
  $RAT_i.n \leftarrow RAT_i.n + 1 \; \forall i \in \mathcal{M} \setminus \{0\}$
  $calculate \; E_{i,j} \; \forall i \in \mathcal{M}$
  $choose \; x_{i,j} \;\; y_j \; s.t. \; E_{i,j} = min\{ \, E_{l,j}; \; l \in \mathcal{M}\}$
  $RAT_l.n \leftarrow RAT_l.n - 1 \; \forall l \in \mathcal{M} \; s.t. \; x_{l,j} \neq 1$
  $X \leftarrow \; x_{i,j}$
  $Y \leftarrow \; y_j$
**end for**

the greedy constructive behavior. Hereby the heuristic logic is maintained but a certain degree of randomness is introduced. This random effect is generated by the use of a skewed probability distribution: at each step of the constructive process, a selection probability is assigned to each potential movement, whereby the probability is higher for the more *promising* movements.

The use of biased-randomization significantly improves the quality of the solutions generated by the original heuristic in different dimensions when considering social or collective performance, reaching a solution closer to the optimal without needing to know all SMD positions.

Moreover, it is important to note that if a uniform probability distribution would be used instead of a skewed one, this improvement would rarely occur since the logic behind the constructive heuristic would be destroyed and the process would be random but not correctly oriented.

To avoid losing the logic behind the heuristic, GRASP meta-heuristics [37] propose the consideration of a restricted list of candidates –i.e., a sub-list including just some of the most promising movements, corresponding to those at the top of the list–, and then apply a uniform randomization to the order in which the elements of that restricted list are selected. In 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.

The proposed biased-randomization approach goes one step ahead. Instead of restricting the list of candidates, different selection probabilities are assigned to each potential movement in the sorted list. That is, elements at the top of the list have a higher probability of being selected than those at the bottom, whereas all elements are potentially eligible. This allows not only to avoid the problem of selecting the proper size of the restricted list, but we can also guarantee that the selection-probabilities are proportional to the position of each element in the list.

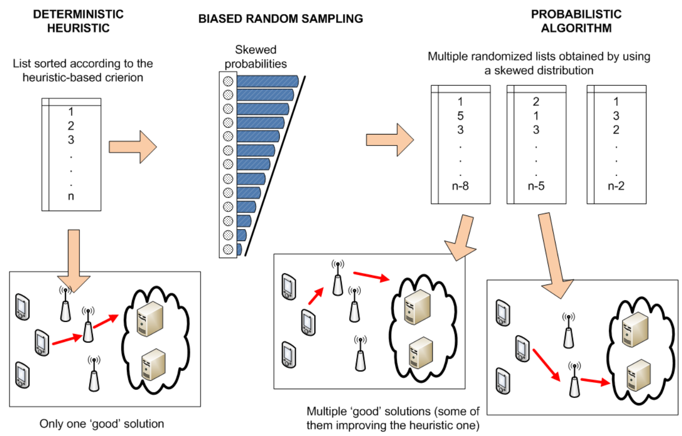



Fig. 2. Scheme of the biased-randomization approach.

Each time the randomized algorithm is executed a new probabilistic solution is obtained (Fig. 2). Some of these solutions allow to improve those obtained through the greedy heuristic. Moreover, the proposed approach allows to achieve different solutions depending on some of the inputs of the model (i.e., throughput, energy, latency, etc.).

Focusing on the selected problem, the biased-randomization algorithm for the link selection consists in a skewed criteria based on the objective function defined in (11), that has been also used for the greedy heuristic.

When there is a new SMD request, the cost function $E_{i,j}$ is evaluated for every possible RAT node (also considering the fictitious node 0 related to the local computation). The obtained values are sorted in such a way that the probabilities of being selected depend not only on the cost function $E_{i,j}$, but also on the distances $d_{i,j}$ of the device from the RAT nodes and on the number of devices $n_i$ that potentially can be connected to each RAT node.

To improve the link selection, the criteria for sorting the choice list takes into account the product $d_{i,j} \cdot E_{i,j}$, with $i \in \mathcal{M}$, instead of $E_{i,j}$. The value of $d_{0,j}$ is adjusted empirically and depends on the overall number of SMDs which are expected to interact in the system. This strategy is reported in Algorithm 2. Note that the modifications included in Algorithm 2 do not increase the computational complexity with respect to Algorithm 1.

## V. Numerical Results

In this Section a realistic test case is evaluated in order to compare the performance of the different algorithms. A deployment area equal to $1000 \times 1000 \; m^2$ is considered, where one LTE eNodeB with a bandwidth of 100 MHz is positioned at point (500,500) and three Wi-Fi access points (AP) with a bandwidth equal to 22 MHz are positioned at point (0,0), (500,999) and (1000,0). An attenuation coefficient $k$ for the propagation equal to $10^{-3}$ has been considered. Furthermore, similarly to [16], the capacity constraints on the antennas is

**Algorithm 2** Biased-randomization Algorithm

**Inputs:**
$\mathcal{M} = \{0, 1, 2, \cdots, M\}$ // $M$ number of RAT nodes
$\mathcal{N} = \{1, 2, \cdots, N\}$ // $N$ number of SMDs
$RAT_i = RAT(BW_i, n_i, pos_{RAT,i}(c_x, c_y))\ i \in \mathcal{M} \setminus \{0\}$
$App = App(O, D)$
**Output:**
$X = (x_{i,j})$
$Y = (y_j)$
**Initialization** :
$d_{0,j} \leftarrow d_0$
$RAT_i.n \leftarrow n_i\ \forall i \in \mathcal{M} \setminus \{0\}$
$x_{i,j} \leftarrow 0\ \forall i \in \mathcal{M}\ \forall j \in \mathcal{N}$
$y_j \leftarrow 0\ \forall j \in \mathcal{N}$
**for** $j = 1$ to $N$ **do**
  $SMD_j = SMD(P_{tr,j}, P_{l,j}, P_{id,j}, f_{md,j}, SNRj, pos_{md,j}(c_x, c_y))$
  $calculate\ d_{i,j}\ ; E_{i,j}\ \forall i \in \mathcal{M}$
  sort $d_{i,j} * E_{i,j}\ ascending\ \forall i \in \mathcal{M}$
  choose $x_{i,j}\ y_j\ s.t.\ E_{a,j} = geoinv(\ E_{i,j};\ i \in \mathcal{M})$
  $X \leftarrow\ x_{i,j}$
  $Y \leftarrow\ y_j$
**end for**

not considered, assuming there is no limitation on the number of SMD per each eNodeB/AP.

The location coordinates of each SMD in the deployment area are sampled from a uniform distribution. In particular, we have chosen a controlled random number generation of the Mersenne Twister type and a seed equal to 1 (this pseudo-random number generator is implemented in Matlab). Fig. 3 represents the region assuming 500 and 5000 SMDs, respectively.

The parameters $f_{md}$, $P_{id}$, $P_{tr}$, $P_l$, and *SNR* depend on the selected mobile device. We considered a HP iPAQ PDA with a 400 MHz Intel XScale processor ($f_{md}$ equal to 400 MHz) and the following values: $P_l$ equal to 0.9 W, $P_{id}$ equal to 0.3, $P_{tr}$ equal to 1.3 W, and SNR equal to 30 dB. For the cloud server we suppose a computation speed $f_{cs}$ equal to $10^6$ MHz [20]. We have chosen an application which is accomplished through a number of operation $O$ equal to $10^7$ and, if offloaded, needs a data transfer $D$ equal to $10^4$ bits.

The algorithms are compared in terms of throughput, energy and time needed for accomplishing the application. The no-connection configuration (i.e., all the SMDs compute locally) and the nearest-node configuration (i.e., each SMD is connected to the closest RAT independently from the RATs and SMDs characteristics) are also taken into account for comparison purposes. However, we choose to compare all the values with those obtained with the greedy algorithm as reference, due to its strict relationship with the proposed biased-randomized approach. The throughput, the energy consumed by the SMDs, and the time employed for accomplishing the application are evaluated and observed for each configuration. Also, different scenarios are considered, each of them involving an increasing number of SMDs: 500, 1000, 2000, and 5000.

Tabs. I, II, and III show, respectively, the results related to the throughput, the energy, and the time. The ex-post solution has been computed with Baron under the Neos server (http://www.neos-server.org). The model was implemented with the mathematical modeling language GAMS. The other methods were implemented in Matlab.

The results show that the biased-randomized algorithm clearly outperforms the link selection configuration obtained with the greedy heuristic. Improvement of 2.24% in throughput, and reductions of 14.30% in energy consumption, and 0.42% in computational time can be observed. Moreover, the average energy and time values provided by the randomized algorithm are similar to the optimal / near-optimal solutions provided by Baron, which has been stopped after 1000 seconds of computational time. This demonstrates that our probabilistic algorithm is able to provide near-optimal solutions –similar to the ones provided by the exact solver– in much lower computation times. The higher improvement in terms of energy is expected, since the objective function in (11a) is designed for minimizing the energy consumption.

Tab. IV shows indeed the time needed by the algorithms for achieving a stable solution of the link selection problem; such time is not directly related to the time needed for the application computation in Tab. III, while it is more related to the ability of the algorithms to solve the problem in a fast way and, hence, its exploitation in a real-life dynamic scenario. On the one hand, it has to be noted that the exact solver requires computation times that cannot be employed in real-life practice. On the other hand, the computing time employed by the biased-randomized algorithm is below a few seconds, which is still a little higher than the one employed by the greedy algorithm. This is due to the fact that the biased-randomized algorithm is run several times in a multi-start process. Nevertheless, it is still an acceptable computing time. Moreover, if necessary the biased-randomized algorithm could be easily parallelized –by simply running each iteration of the multi-start process in a different core or computer–, so its computing time would be the same as the one employed by the greedy algorithm –i.e., in the order of milliseconds.

The results are visually compared in Figs. 4 and 5a, where we can see that the performance of the biased-randomized algorithm is very similar in terms of average throughput, compared to Baron's ex-post configuration. The greedy heuristic is clearly outperformed. This means that the capacity of the RAT nodes is well-exploited in the biased-randomized algorithm. In Fig. 4 the average throughput values are shown in a different scale to make possible the comparison of the three algorithms.

Considering the average energy consumed by one SMD to perform the application, Fig. 5b shows that the biased-randomized algorithm clearly outperforms the greedy heuristic. The same observation can be inferred from Fig. 5c for the average time that a device needs to accomplish the application. Both the average energy and time values are very close to the respective results of Baron. Thus, the biased-randomized algorithm can be exploited by the SMDs to reach near-optimal configurations in real time, while providing solutions that tend to minimize the overall energy consumption. This is useful for offloading applications which need a fast link selection

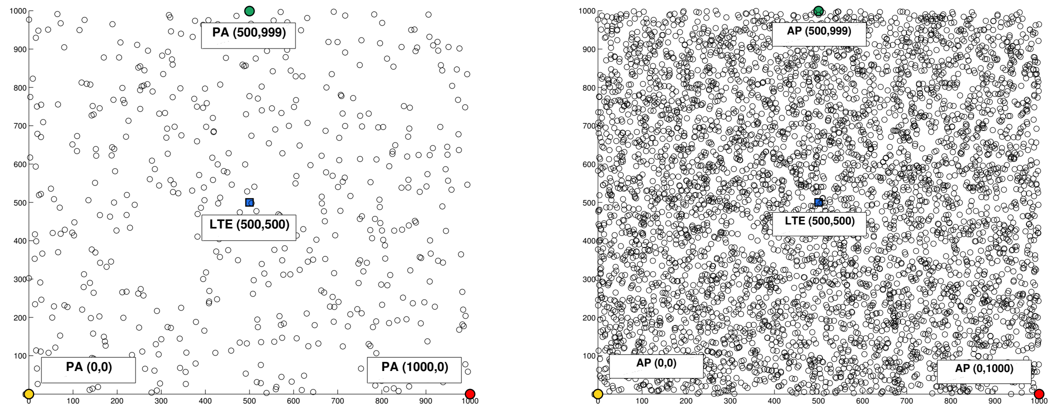



Fig. 3. Scenarios in case of 500 and 5000 SMDs, respectively, three Wi-Fi access points, and one LTE eNodeB.

TABLE I
THROUGHPUT [MBPS]

| N SMDs | 500 | | 1000 | | 2000 | | 5000 | |
|---|---|---|---|---|---|---|---|---|
| | Avg. | Dev. [%] | Avg. | Dev. [%] | Avg. | Dev. [%] | Avg. | Dev. [%] |
| Nearest Node | 1.5693 | -50.26 | 0.7847 | -50.33 | 0.3926 | -50.36 | 0.1570 | -50.46 |
| Greedy | 3.1551 | ref. | 1.5799 | ref. | 0.7909 | ref. | 0.3169 | ref. |
| Biased Rand. | 3.1699 | +0.47 | 1.5840 | +0.26 | 0.8004 | +1.20 | 0.3240 | +2.24 |
| Baron | 3.1707 | +0.50 | 1.5844 | +0.29 | 0.7999 | +1.14 | 0.3240 | +2.22 |

TABLE II
ENERGY [W*S]

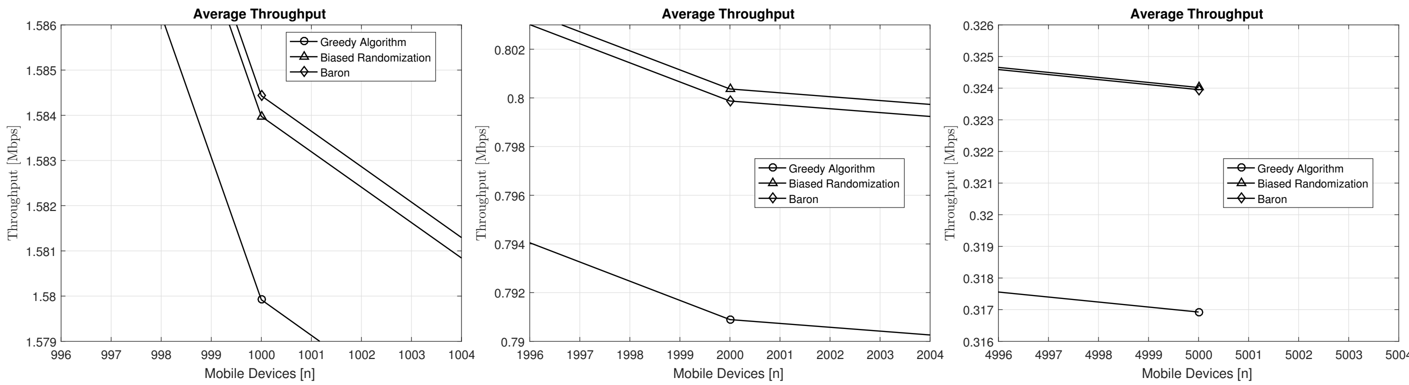
| N SMDs | 500 | | 1000 | | 2000 | | 5000 | |
|---|---|---|---|---|---|---|---|---|
| | Avg. | Dev. [%] | Avg. | Dev. [%] | Avg. | Dev. [%] | Avg. | Dev. [%] |
| Local | 22500.0000 | +445.26 | 22500.0000 | +173.19 | 22500.0000 | +36.80 | 22500.0000 | -2.17 |
| Nearest Node | 4609.3141 | +11.70 | 9819.6194 | +19.23 | 18832.6714 | +14.50 | 46060.3241 | +100.26 |
| Greedy | 4126.5039 | ref. | 8236.0653 | ref. | 16447.8885 | ref. | 22999.8480 | ref. |
| Biased Rand. | 4106.4181 | -0.49 | 8239.5161 | +0.04 | 14848.2169 | -9.73 | 19711.6979 | -14.30 |
| Baron | 4104.4986 | -0.53 | 8210.9327 | -0.31 | 14716.6647 | -10.53 | 19347.2418 | -15.88 |

TABLE III
TIME FOR APPLICATION COMPUTATION [S]

| N SMDs | 500 | | 1000 | | 2000 | | 5000 | |
|---|---|---|---|---|---|---|---|---|
| | Avg. | Dev. [%] | Avg. | Dev. [%] | Avg. | Dev. [%] | Avg. | Dev. [%] |
| Local | 25000.0000 | +685.69 | 25000.0000 | +294.13 | 25000.0000 | +97.47 | 25000.0000 | +19.01 |
| Nearest Node | 3553.3186 | +11.67 | 7561.2457 | +19.20 | 14494.3626 | +14.49 | 35438.7108 | +68.70 |
| Greedy | 3181.9261 | ref. | 6343.1271 | ref. | 12659.9142 | ref. | 21007.3416 | ref. |
| Biased Rand. | 3166.4755 | -0.49 | 6345.7816 | +0.04 | 14003.7438 | +10.61 | 20918.6292 | -0.42 |
| Baron | 3164.9989 | -0.53 | 6323.7944 | -0.30 | 13745.0152 | +8.57 | 20432.3306 | -2.74 |

TABLE IV
TIME FOR COMPLETING THE ALGORITHMS [S]

| N SMDs | 500 | 1000 | 2000 | 5000 |
|---|---|---|---|---|
| Greedy | 0.41 | 0.85 | 1.65 | 4.07 |
| Biased Rand. | 3.34 | 6.64 | 11.74 | 28.44 |
| Baron | stopped after 1000 sec. | | | |




(a) Throughput for 1000 SMDs (b) Throughput for 2000 SMDs (c) Throughput for 5000 SMDs

Fig. 4. Comparison of the performance in terms of throughput for different SMDs number.

decision, for example when the users are moving and the configuration must be updated very often.

Fig. 6 and Fig. 7 report the different configurations (of the devices placed in the observed area) for an overall number of SMDs equal to 500 and 5000, respectively. Comparing these configurations, we can note that the case referring to the biased-randomized algorithm is very similar to the ex-post optimal solution, where the SMDs connected to the same RAT are clustered. In particular, in these figures the nodes represent SMDs connected to each eNodeB/AP of the same gray or, alternatively, performing local computation, represented as black dots. It is possible also to highlight that the biased-randomized approach allows to adapt the link selection based on the node density, while the greedy approaches only take into account the nodes density in a limited way. This confirms the 'socially-aware' optimization behavior of the proposed biased-randomized approach.

## VI. Conclusion

In this paper we have presented different strategies for an efficient link selection in pervasive wireless environments. In particular, we have proposed a novel randomized algorithm that allows to consider a more global point of view –instead of just a greedy or individual one– during the real-time resource allocation. Thus, the mobile communication system is managed in a cooperative way, considering the QoS of the entire population of mobile users. This vision could be adopted for the provision of services in a smart city, improving performance and well-being of the community through a social use of HetNets and MCC. Furthermore, it can be seen as an enhancing method from an eco-friendly perspective, since it aims to save SMD's energy. Our approach is based on the use of biased-randomization techniques, which have been used in the past to solve similar combinatorial optimization problems in the fields of logistics, transportation, and production. This work extends their use to the field of smart cities and mobile telecommunications. Some numerical results contribute to illustrate the potential of the proposed approach.

## References

[1] World Urbanization Prospects - Population Division - United Nations. [Online]. Available: http://esa.un.org/unpd/wup/

[2] Cloud and Mobile Network Traffic Forecast - VNI - Cisco. [Online]. Available: http://www.cisco.com/c/en/us/solutions/service-provider/visual-networking-index-vni/

[3] C. Manville, G. Cochrane, J. Cave, J. Millard, J. K. Pederson, R. K. Thaarup, A. Liebe, M. Wissner, R. Massink, and B. Kotterink, "Mapping Smart Cities in the EU," Tech. Rep., 2014. [Online]. Available: http://www.europarl.europa.eu/RegData/etudes/etudes/join/2014/507480/IPOL-ITRE_ET(2014)507480_EN.pdf

[4] *Designing the urban future: Smart Cities*. New York, USA: Scientific American, 2014.

[5] C. Zhu, V. Leung, L. Shu, and E. Ngai, "Green Internet of Things for Smart World," *IEEE Access*, no. 3, pp. 2151–2162, November 2015.

[6] M. Dohler, C. Ratti, J. Paraszczak, and G. Falconer, "Smart cities [guest editorial]," *IEEE Commun. Mag.*, vol. 51, no. 6, pp. 70–71, Jun. 2013.

[7] H. T. Dinh, C. Lee, D. Niyato, and P. Wang, "A survey of mobile cloud computing: architecture, applications, and approaches," *Wireless Communications and Mobile Computing*, vol. 13, no. 18, pp. 1587–1611, Dec. 2013.

[8] B. Batty, *The New Science of Cities*. Cambridge, MA, USA: The MIT Press, 2013.

[9] R. Hu, Y. Qian, S. Kota, and G. Giambene, "Hetnets - a new paradigm for increasing cellular capacity and coverage [guest editorial]," *IEEE Wireless Communications*, vol. 18, no. 3, pp. 8–9, Jun. 2011.

[10] L. Zhou, C. Zhu, R. Ruby, X. Wang, X. Ji, S. Wang, and J. Wei, "QoS-aware energy-efficient resource allocation in OFDM-based heterogeneous cellular networks," *International Journal of Communication Systems*, 2015.

[11] L. Lei, Z. Zhong, K. Zheng, J. Chen, and H. Meng, "Challenges on wireless heterogeneous networks for mobile cloud computing," *IEEE Wireless Commun. Mag.*, vol. 20, no. 3, pp. 34–44, Jun. 2013.

[12] A. U. R. Khan, M. Othman, S. A. Madani, and S. U. Khan, "A survey of mobile cloud computing application models," *IEEE Communications Surveys and Tutorials*, vol. 16, no. 1, pp. 393–413, 2014.

[13] S. Barbarossa, S. Sardellitti, and P. Di Lorenzo, "Communicating while computing: Distributed mobile cloud computing over 5G heterogeneous networks," *IEEE Signal Processing Magazine*, vol. 31, no. 6, pp. 45–55, Nov. 2014.

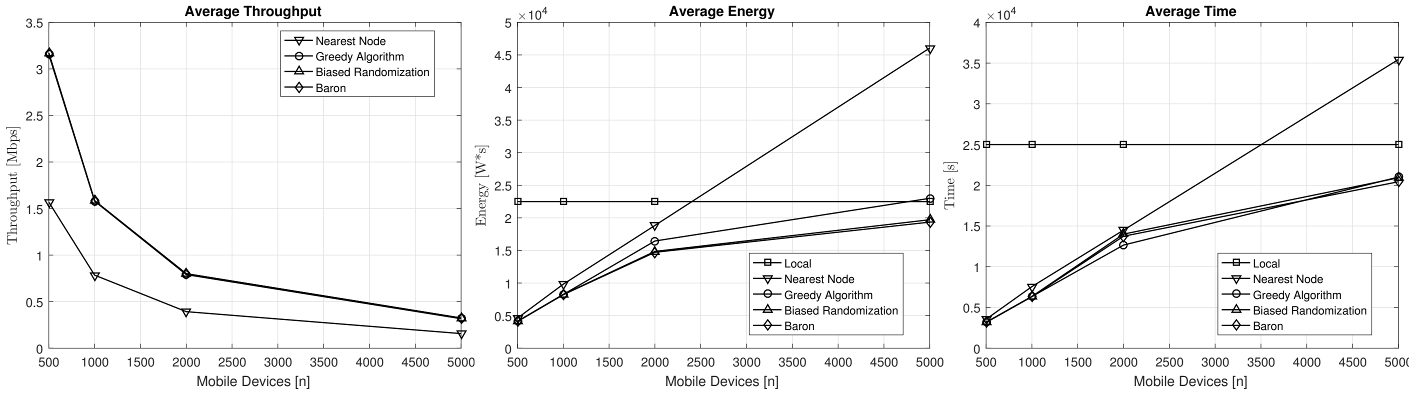
[14] X. Chen, "Decentralized computation offloading game for mobile cloud computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 4, pp. 974–983, Apr. 2015.

[15] 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," *Journal of the Operational Research Society*, vol. 62, no. 6, pp. 1085–1097, Jun. 2011.

[16] 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.

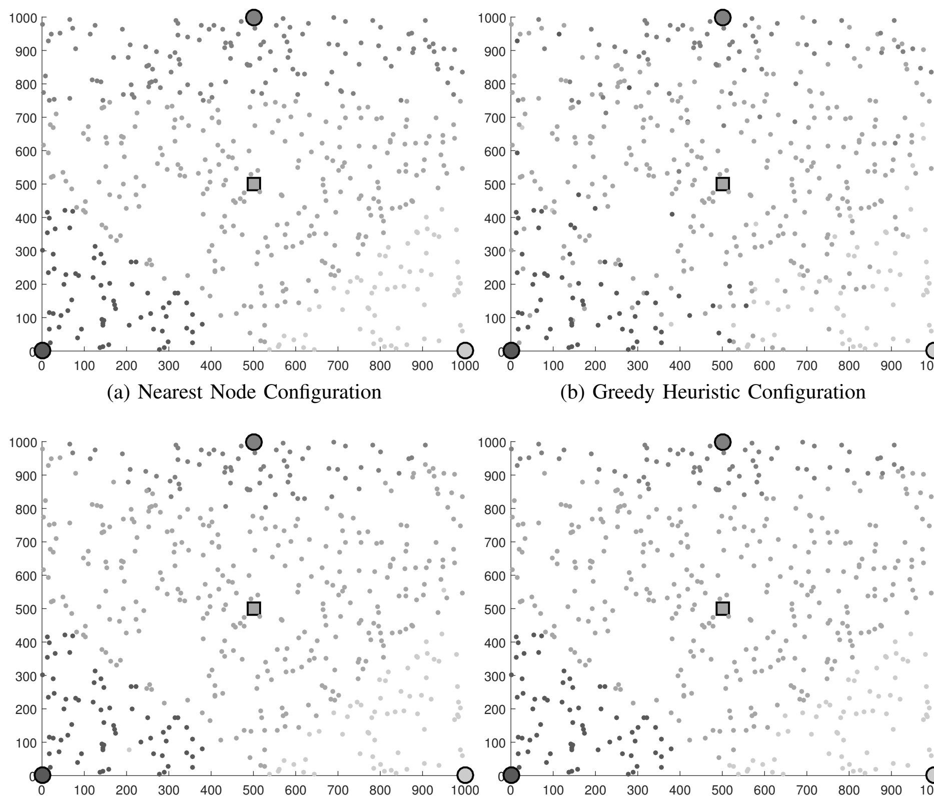
[17] W. Zhang, Y. Wen, and H.-H. Chen, "Toward transcoding as a service: energy-efficient offloading policy for green mobile cloud," *IEEE Network*, vol. 28, no. 6, pp. 67–73, Nov. 2014.

[18] B. Han, P. Hui, V. S. A. Kumar, M. V. Marathe, J. Shao, and A. Srinivasan, "Mobile data offloading through opportunistic communications and social participation," *IEEE Transactions on Mobile Computing*, vol. 11, no. 5, pp. 821–834, May 2012.




(a) Throughput (b) Energy (c) Latency

Fig. 5. Performance in terms of throughput, energy consumption and latency for a variable number of devices



(a) Nearest Node Configuration (b) Greedy Heuristic Configuration

(c) Biased-randomized Configuration (d) Ex-post Optimal Configuration

Fig. 6. Link selection in case of 500 SMDs. Each gray indicates that the SMD is connected to the RAT marked with the same gray.


[19] F. Liu, P. Shu, H. Jin, L. Ding, J. Yu, D. Niu, and B. Li, "Gearing resource-poor mobile devices with powerful clouds: architectures, challenges, and applications," *IEEE Wireless Communications*, vol. 20, no. 3, pp. 14–22, Jun. 2013.

[20] 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.

[21] E. Cuervo, A. Balasubramanian, D.-k. Cho, A. Wolman, S. Saroiu, R. Chandra, and P. Bahl, "Maui: Making smartphones last longer with code offload," in *Proc. of MobiSys '10*, San Francisco, CA, USA, Jun. 2010, pp. 49–62.

[22] B.-G. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti, "CloneCloud: Elastic execution between mobile device and cloud," in *Proc. of EuroSys '11*, Salzburg, Austria, Apr. 2011, pp. 301–314.

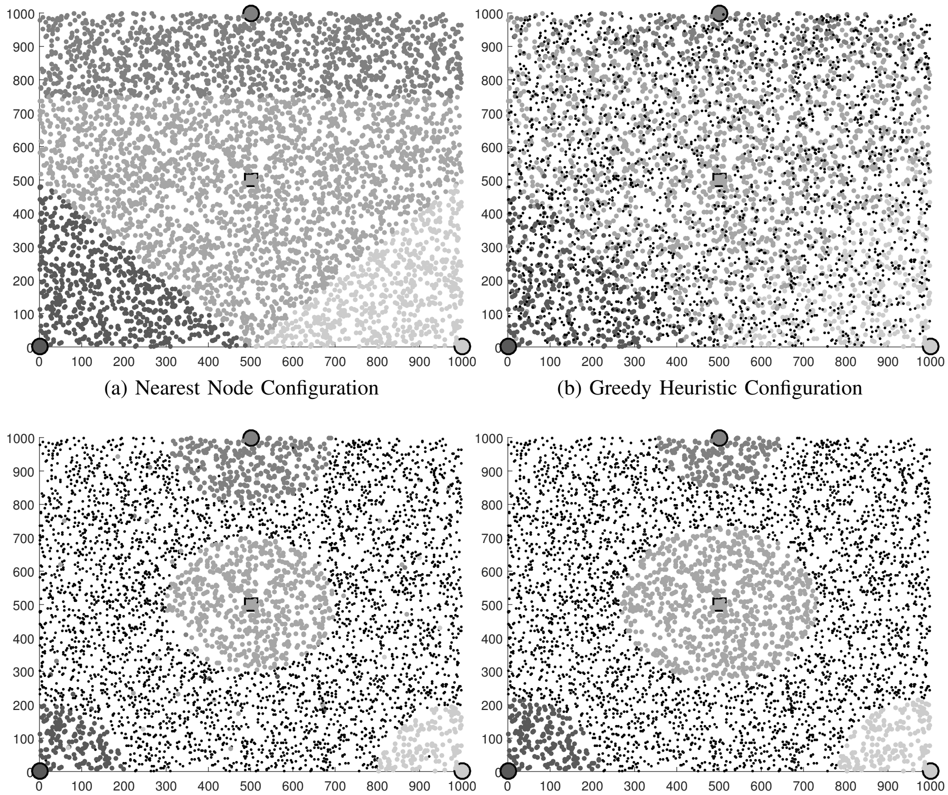
[23] A. Ur, R. Khan, M. Othman, A. N. Khan, S. Akhtar, A. Sajjad, A. Madani, A. R. Khan, A. N. Khan, M. Othman, S. A. Abid, and S. A. Madani, "MobiByte: An Application Development Model for Mobile Cloud Computing," *Journal of Grid Computing*, vol. 13, no. 4, pp. 605–628, 2015.

[24] D. Huang, P. Wang, and D. Niyato, "A dynamic offloading algorithm for mobile computing," *IEEE Transactions on Wireless Communications*, vol. 11, no. 6, pp. 1991–1995, Jun. 2012.

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

[26] 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. 1–32, Jan. 2015.

[27] 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, Jan.




(a) Nearest Node Configuration (b) Greedy Heuristic Configuration

(c) Biased-randomized Configuration (d) Ex-post Optimal Configuration

Fig. 7. Link selection in case of 5000 SMDs. Each gray indicates that the SMD is connected to the RAT marked with the same gray, whereas the black SMDs are computing the application locally.

2007.

[28] 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, Jun. 2007.

[29] 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, Mar. 2010.

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

[31] 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 Communications*, vol. 25, no. 2, pp. 173–189, Apr. 2012.

[32] A. A. Juan, H. R. Lourenco, 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, Jan. 2014.

[33] M. Nawaz, E. 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://www.sciencedirect.com/science/article/pii/0305048383900889

[34] A. Ur, R. Khan, and A. N. Khan, "Context-Aware Mobile Cloud Computing and Its Challenges," *IEEE Computer Society*, vol. 2, no. 3, pp. 42–49, 2015.

[35] R. E. Burkard, "Quadratic assignment problems," in *Handbook of Combinatorial Optimization*, P. M. Pardalos, D.-Z. Du, and R. L. Graham, Eds. Springer New York, 2013, pp. 2741–2814.

[36] M. Tawarmalani and N. V. Sahinidis, "A polyhedral branch-and-cut approach to global optimization," *Mathematical Programming*, vol. 103, no. 2, pp. 225–249, Jun. 2005.

[37] T. A. Feo and M. G. C. Resende, "Greedy randomized adaptive search procedures," *Journal of Global Optimization*, vol. 6, no. 2, pp. 109–133, Mar. 1995.