A robust approach to the cell switch-off problem in 5G ultradense networks

Francisco Luna*, Pablo H. Zapata-Cano
Dept. of Languages and Computer Science
Universidad de Málaga
Málaga, Spain
Email: {flv,phzc}@lcc.uma.es

Juan F. Valenzuela-Valdés, Pablo Padilla
Dept. Signal Theory, Telematics and Communications
Universidad de Granada
Granada, Spain
Email: {juanvalenzuela,pablopadilla}@ugr.es

***Abstract*—Ultra-dense networks (UDNs) are recognized as one of the key enabling technologies of the fifth generation (5G) networks, as they allow for an efficient spatial reuse of the spectrum, which is required to meet the traffic demands foreseen for the next coming years. However, the power consumption of UDNs, with potentially hundreds of small base stations (SBSs) within each macrocell, is a major concern for the cellular operators, and has to be properly addressed prior to the actual deployment of these 5G networks. Among the different existing approaches to address this issue, a widely accepted strategy lies in the selective deactivation of SBSs, but without compromising the QoS provided to the User Equipments (UEs). This is known as the Cell Switch-Off (CSO) problem. The typical formulation of this problem is based on estimations of the traffic demand of the User Equipments (UEs) within the network. But these estimations could not be met. This work approaches these uncertain scenarios by extending the CSO problem with additional objectives that account for the robustness of the solutions to disturbances in these traffic estimates. To do so, a computationally demanding Monte-Carlo sampling is used to evaluate each solution. To manage such an increasingly large computing cost, a parallel version of the NSGA-II algorithm that is able to run on a computing platform composed of more than 500 cores has been used. It is able to compute in roughly 2 hours, an accumulated execution time of more than 42 days, which is within the expected timeframe of operators to make changes in the network configuration.**

***Index Terms*—Robustness, parallelism, cell switch-off, multi-objective optimization, metaheuristics.**

I. Introduction

Switching off base stations is a well-known strategy for saving the energy consumption of the newly envisioned fifth generation (5G) cellular networks [1]. Indeed, it has been standardized by the 3rd Generation Partnership Project (3GPP) [2]. In order for these 5G networks to provide the foreseen traffic demands and the massive connectivity featured, network densification, i.e., the deployment of a large number of heterogeneous small base stations (SBSs) [3], [4], is a key enabler technology, as it increases the spatial reuse of the spectrum and thus the system capacity. The resulting networks are known as ultra-dense networks (UDNs) [5].

But with such a large number of SBSs functioning in the network, the energy consumption of UDNs clearly contrasts with one of the main design issues required for 5G systems [6]: they have to operate with 90% energy savings. In this context, different approaches have been proposed in the literature to address the problem of energy consumption [7]. Among them, switching off a subset of SBSs within a UDN, known as the CSO (Cell Switch-Off) problem, is one of the main approaches. The CSO problem has been also addressed from different perspectives, ranging from clustering techniques [8], [9] to its formulation as an optimization problem [10]. As an optimization problem, CSO has been tackled with both simple heuristics [11], [12] and metaheuristics [13], [14], [15], [16] algorithms. A common assumption in all these approaches is that the traffic demand in the network, represented by a number of mobile users or User Equipments (UEs), is usually estimated [17]. As so, the optimization algorithms are aimed to look for solutions, or the subset of SBSs to be switched off, based on such estimates. The point is that, if these estimates do have inaccuracies, the solutions found might be useless as they could be quite overfitted and may cause the network to start performing poorly, dropping the QoS indicators down to unacceptable levels. Our approach in this work is to make the solutions to the CSO problem robust [18], i.e., to add them some degree of insensitivity to disturbances [19] in the estimation of the traffic demands. To do so, additional objectives that measure such robustness are introduced, resulting in the definition of the Robust CSO (RCSO) problem.

This paper extends a previous work [14], where a multi-objective formulation of the CSO problem has targeted, simultaneously, the minimization of power consumption and the maximization of the total capacity that the network is capable of providing to the UEs. The goal here is to provide the solutions to this problem with robustness, that is, addressing the RCSO problem defined above. Since measuring the robustness of a solution is a mathematically and challenging task [20], an approximated procedure based on a probabilistic sampling has been used (Monte Carlo integration) [21], [22]. It lies in evaluating every newly generated solution over H different environments or, equivalently, over H different traffic demands. Therefore, for each solution, we have H different objective values from which its robustness has to be computed. This is undertaken by extending the two-objective CSO problem to a four-objective RCSO problem that aims at minimizing the average power consumption and maximizing the average capacity over the H samples, but also minimizing

978-1-7281-4484-9/19/$31.00 ©2019 IEEE

their two variances, as indicators of robustness. That is, the lower the variance of the sample, the higher the insensitivity of the solution to the variations of the traffic demands. To the best of our knowledge, this is the very first time the RCSO problem is addressed in the literature.

But modelling a UDN accurately involves computing complex signal processing methods which, along with the high dimension of the problem instances (thousands of SBSs and UEs), provokes that evaluating just one single solution of the problem is a computationally demanding task. Given that H samples have to be evaluated to compute the objectives of each solution, the resulting run times become unaffordable for practical applications. We have addressed this issue by using a parallel version of NSGA-II [23], named mwNSGA-II, that is able to incorporate as many parallel computing nodes as available, without changing substantially the evolutionary loop of the algorithm. The results presented in this work are able to use up to 500 processor cores and reduce the runtime of an execution of a complex RCSO instance from 42 days to two hours.

The rest of the paper is organized as follows. The next section defines the RCSO problem, by detailing the UDN modelling and the formulation of both the CSO and RCSO problems. Section III describes the parallelization of the NSGA-II algorithm. The experimentation performed and the discussion of the results is included in Sect. IV. Finally, the last section summarizes the main conclusions of the work as well as the lines for future research.

II. The robust CSO problem

This section is devoted to detailing the UDN model used, as well as the formulation of both the CSO and the RCSO problems.

A. System modelling

We have a target service area of 500×500 square meters, which has been discretized using a grid of 100×100 points (also called "pixels" or area elements), each covering a 25 m^2 area where the signal power is assumed to be constant. Ten different regions have been defined with different propagation conditions. In order to compute the received power at each point, $P_{rx}[dBm]$, the following model has been used:

$$P_{rx}[dBm] = P_{tx}[dBm] + PLoss[dB] \quad (1)$$

where, P_{rx} is the received power in dBm, P_{tx} is the transmitted power in dBm, and $PLoss$ are the global signal losses, which depend on the given propagation region, and are computed as:

$$PLoss[dB] = GA + PA \quad (2)$$

where GA is the total gain of both antennas, and PA are the transmission losses in space, computed as:

$$PA[dB] = \left(\frac{\lambda}{2 * \pi * d}\right)^K \quad (3)$$

where d is the Euclidean distance to the SBS, K is the exponent loss, which ranges randomly in $[2.0, 4.0]$ for each of the 10 different regions. The signal to interference plus noise ratio (SINR) for UE k, is computed as:

$$SINR_k = \frac{P_{rx,j,k}[mW]}{\sum_{i=1}^{M} P_{rx,i,k}[mW] - P_{rx,j,k}[mW] + P_n[mW]} \quad (4)$$

where $P_{rx,j,k}$ is the received power by UE k from SBS j, the summation is the total received power by UE k from all the SBSs operating at the same frequency that j, and P_n is the noise power, computed as:

$$P_n = -174 + 10 \log_{10} BW_j \quad (5)$$

being BW_j the bandwidth of SBS j, defined as 5% of the SBS operating frequency (see Table I). Finally, the capacity of the UE k is:

$$C_k^j[bps] = BW_k^j[Hz] * \log_2(1 + SINR_k) \quad (6)$$

where BW_k^j is the bandwidth assigned to UE k when connected to SBS j, assuming a round robin scheduling, that is:

$$BW_k^j = \frac{BW_j}{N_j} \quad (7)$$

where N_j is the number of UEs connected to SBS j, and UEs are connected to the SBS with the highest SINR, regardless of its type.

In order to model an heterogeneous network, four different types of cells of decreasing size are considered: femtocells, picocells, microcells, and macrocells. Two subtypes of femto, pico and microcells are also defined, summing up 7 cell types. The SBSs are deployed using independent Poisson Point Processes (PPP) with different densities (defined by λ_P^{SBS}). UEs are also deployed using a PPP (defined by λ_P^{UE}), but using social attractors (SAs), following the procedure defined in [24]. This deployment scheme uses two factors, α and μ_β, that indicate how strong SBSs attract SAs and how SAs attract UEs. They have been set to $\alpha = \mu_\beta = 0.25$.

The power consumption of a transmitter is computed based on the model presented in [25], which considers that the device is transmitting over the fiber backhauling. Hence, the regular power consumption of SBS j, P_j, is expressed as:

$$P_j = \alpha * P + \beta + \delta * S + \rho \quad (8)$$

where P denotes the transmitted or radiated power of the transmitter, the coefficient α represents the efficiency of transmit power produced by an radio-frequency amplifier and feeder losses, the power dissipated owing to signal processing and site cooling is denoted by β, the dynamic power consumption per unit data is given by δ, being S the actual traffic demand served by the SBS, and, finally, the power consumption of the transmitting device is represented by the coefficient ρ.

The detailed parametrization of the scenarios addressed is included in Table I, in which the column Eq. links the parameter to the corresponding equation in the formulation detailed above. The names in the last two columns, XY, stand for the

TABLE I: Model parameters for cells and users

Cell	Parameter	Eq.	LL	HH
macro	G_{tx}	(2)	14	
	f	(5)	2 GHz (BW = 100 MHz)	
	α	(8)	21.45	
	β	(8)	344440	
	δ	(8)	2	
	$\rho[W]$	(8)	1	
micro1	G_{tx}	(2)	12	
	f	(5)	3.5 GHz (BW = 175 MHz)	
	α	(8)	15	
	β	(8)	10000	
	δ	(8)	1	
	$\rho[W]$	(8)	1	
	λ_P^{micro1} $[SBS/km^2]$		100	300
micro2	G_{tx}	(2)	10	
	f	(5)	5 GHz (BW = 250 MHz)	
	α	(8)	15	
	β	(8)	10000	
	δ	(8)	1	
	$\rho[W]$	(8)	1	
	λ_P^{micro2} $[SBS/km^2]$		100	300
pico1	G_{tx}	(2)	5	
	f	(5)	10 GHz (BW = 500 MHz)	
	α	(8)	9	
	β	(8)	6800	
	δ	(8)	0.5	
	$\rho[W]$	(8)	1	
	λ_P^{pico1} $[SBS/km^2]$		500	700
pico2	G_{tx}	(2)	7	
	f	(5)	14 GHz (BW = 700 MHz)	
	α	(8)	9	
	β	(8)	6800	
	δ	(8)	0.5	
	$\rho[W]$	(8)	1	
	λ_P^{pico2} $[SBS/km^2]$		500	700
femto1	G_{tx}	(2)	4	
	f	(5)	28 GHz (BW = 1400 MHz)	
	α	(8)	5.5	
	β	(8)	4800	
	δ	(8)	0.2	
	$\rho[W]$	(8)	1	
	λ_P^{femto1} $[SBS/km^2]$		1000	3000
femto2	G_{tx}	(2)	3	
	f	(5)	66 GHz (BW = 3300 MHz)	
	α	(8)	5.5	
	β	(8)	4800	
	δ	(8)	0.2	
	$\rho[W]$	(8)	1	
	λ_P^{femto2} $[SBS/km^2]$		1000	3000
UEs	λ_P^{UE} $[UE/km^2]$		1000	3000

deployment densities of SBSs and UEs, respectively, so that X = {L,H}, meaning either low or high density deployments (λ_P^{SBS} parameter of the PPP), and Y = {L,H}, indicates a low or high density of deployed UEs (λ_P^{UE} parameter of the PPP), in the last row of the table. The parameters G_{tx} and f of each type of cell refer to the transmission gain and the operating frequency (and its available bandwidth) of the antenna, respectively. Two instances have been used in this work, namely LL and HH. In the former, a lower density for both SBSs and UEs is applied, whereas the latter is setup with a high density for SBSs and UEs.

B. The CSO problem

Let $\mathcal{B}$ be the set of the SBSs deployed after the randomized procedure described in the previous section. A solution to the CSO problem is a binary string $s \in \{0,1\}^{|\mathcal{B}|}$, where s_i indicates whether SBS i is activated or not. The first objective to be minimized is therefore computed as:

$$\min f_{Power}(s) = \sum_{i=1}^{|\mathcal{B}|} s_i P_i \tag{9}$$

where P_i is the power consumption of SBS i (Eq. 8).

Let $\mathcal{U}$ be the set of the UEs also deployed as described in the section above. In order to compute total capacity of the system, the UEs are first assigned to the SBS that services it with the highest SINR. Let $\mathcal{A} \in \{0,1\}^{|\mathcal{U}|x|\mathcal{B}|}$ be the matrix where $a_{ij} = 1$ if SBS j serves UE i with the highest SINR, and $a_{ij} = 0$ otherwise. Then, the second objective to be maximized, which is the total capacity provided to all the UEs, is calculated as:

$$\max f_{Cap}(s) = \sum_{i=1}^{|\mathcal{U}|} \sum_{j=1}^{|\mathcal{B}|} a_{ij} BW_i^j \tag{10}$$

where BW_i^j is the shared bandwidth of SBS j provided to UE i (Eq. 7).

C. Adding robustness

This work introduces the possibility of handling disturbances in the environmental variables of the CSO problem, i.e., the traffic demand represented by the positions of the UEs in the service area. These disturbances can be modelled by using a multivariate random variable $\mathbf{P}_{UE} = \{p_1, \ldots, p_{|\mathcal{UE}|}\}$ that represents the positions of the UEs. Given a solution s, the power consumption and the capacity of the UDN are defined by a bivariate random variable $T = (f_{Power}, f_{Cap}) = \vec{f}(s, \mathbf{P}_{UE})$, that is, the evaluation of f_{Power} and f_{Cap} depends on the randomized positions of the UEs. In this work we assume that p_i can be uniformly randomly chosen from $[0, 500] \times [0, 500]$, that is, the UEs can be randomly placed everywhere on the service area.

As the evaluation of the solutions is now a random variable, the concrete objective functions to be optimized are based on statistics of T. The mean, μ, and the standard deviation, σ, have been used respectively as location and dispersion statistics. The former is a measure of the quality of the solution according to the objective, whereas the latter is a measure of its robustness. Unfortunately, these measures cannot be computed analytically for reasonably complex problems, so Monte Carlo integration is used to estimate them by sampling over H simulations of U [20]. This means evaluating H times more function evaluations, with the subsequent computational cost. The four objective functions of the RCSO problem are therefore:

$$
\begin{aligned}
f_1 &= \min\{\mu_{f_{Power}}\} && (11)\\
f_2 &= \min\{\sigma_{f_{Power}}\} && (12)\\
f_3 &= \max\{\mu_{f_{Cap}}\} && (13)\\
f_4 &= \min\{\sigma_{f_{Cap}}\} && (14)
\end{aligned}
$$

that is, minimize the power consumption of the UDN (f_1), maximize the capacity provided to the UEs (f_3) and also look for more robust solutions, i.e., those that are lesser sensitive to disturbances, what means minimizing the standard deviation over the H samples, namely f_2 and f_4.

III. The parallel NSGA-II algorithm

The Non-Dominated Sorting Genetic Algorithm II, NSGA-II, was proposed by Deb *et al.* [26]. It is a genetic algorithm based on generating a new population from the original one by applying the typical genetic operators (selection, crossover, and mutation); then, the individuals in the new and old population are sorted according to their rank, and the best solutions are chosen to create a new population. In case of having to select some individuals with the same rank, a density estimation based on measuring the crowding distance to the surrounding individuals belonging to the same rank is used to get the most promising solutions.

The parallelization of NSGA-II, named as mwNSGA-II, follows the classical master-worker paradigm, and is developed within the framework presented in [23]. A multi-threaded master uses a thread to handle the connection with each worker, and communicate via tasks, which are just containers of tentative solutions that are sent for remote evaluation. The main advantage of this parallel framework is its capability to take advantage of a higher number of computing elements (processors, cores, etc.) than that of the population size of the evolutionary algorithm, which usually is the limiting factor for parallelism in the master-slave approach. That is, the remote evaluation of the N solutions of the population. To do so, the parallelization devised breaks down the synchronization requirements imposed by the evolutionary loop of the sequential algorithms by generating as many solutions as workers available, and incorporating the evaluated solutions into the population regardless of the generation it was created. The framework mentioned above is implemented in Java, and the native socket interface is used as the technology for handling the remote communications. Additional details can be found in [23].

IV. Experimentation

This section includes the full description of the experimentation conditions: first, the parallel computing platform used to run the algorithms, and, second, the analysis of the results obtained.

A. Parallel computing platforms

Two different platforms have been used. The first one, named as DCC, is composed of the computers of the teaching labs of the Department of Computer Science at the University of Málaga (UMA). It has 320 machines with two cores (640 total cores), 8GB of RAM, Windows 10 (64 bits). There are two types of machines: 192 with Pentium Dual-Core CPU E5500 2,80 GHz, and 128 with Pentium CPU G3220 3.60 GHZ. The interconnection network is a 10GB Ethernet. The workers in this platform are deployed with the the Condor system[1]. Given the memory requirements of the UDN modelling with a large number of SBSs and UEs, only the LL instance (see Table I) has been able to be computed in this platform.

In order to run the HH instance, the facilities of the Supercomputing and Bioinformatics center, namely Picasso, of UMA has been used. It is composed of a 48 nodes x 2 E5-2670 processors x 8 cores at 2.6 GHz and with 64GB of RAM each, 7 shared memory machines with 2TB of RAM each (7 x 8 E7-4870 processors x 10 cores at 2.4 GHz), and 168 nodes x 2 Intel E5-2670 processors x 8 cores at 2.6 GHz with 32 GB of RAM. The full hardware description can be found in `http://www.scbi.uma.es/site/scbi/hardware`. The total amount of resources is composed of 4016 cores and an Infiniband QDR/FDR interconnection network, but at most 512 can be used simultaneously. The deployment of the workers within this platform is performed with the Slurm workload manager[2].

We want to remark that the two platforms are shared among multiple users and, as a consequence, guaranteeing that the same amount of resources (cores) is available for all the runs has not been possible. Indeed, using 512 cores in Picasso has been actually difficult, as the it provides service to both users from UMA and from the Spanish Supercomputing Network. Note also that both DCC and Picasso are composed of heterogeneous nodes, what is also a challenge for parallel algorithms, but it is an issue properly addressed in the mwNSGA-II design.

B. Algorithmic settings

The mwNSGA-II algorithm uses, as genetic operators, Two Point Crossover with a crossover rate of 0.9, and Bit Flip mutation with a mutation rate of $1/L$, where $L = |\mathcal{B}|$, the number of SBSs of the UDN. Binary tournament is the selection operator and the stopping condition is to compute 100000 function evaluations. The number of samples to compute the robustness of the solutions is $H = 100$. The algorithm has been implemented within the jMetal framework[3]. Given the particularities of the parallel platforms, we have been limited to run only 5 independent runs of the algorithms.

[1] `https://research.cs.wisc.edu/htcondor/`
[2] `https://slurm.schedmd.com/`
[3] `https://github.com/jMetal/jMetal`

C. Results

This section has been further structured into three different subsections for a better organization of the discussion of the results. The first one shows the performance of the parallel implementation on the two platforms described above. Then, a visual inspection of the approximated Pareto fronts reached and, finally, a sensitivity analysis of the robust solutions is undertaken.

1) Parallel performance: In order to show the performance reached by the parallel execution of mwNSGA-II, Table II includes the following indicators. The wall-clock time, defined as the actual runtime of the algorithm, which is computed by the master node of the algorithm by substracting the initial starting time of the run to the time when all the workers have finished. The accumulated run time of all the workers involved in the parallel computation to evaluate the 100000 function evaluations. In order to measure this metric, the workers are developed to write into a log file the time required to evaluate every single solution it receives, then, from these log files, the total aggregated runtime is summed up. The parallel efficiency, computed as the percentage of the actual CPU usage of the workers (without considering the communications overhead). This is measured by substracting the value of the previous metric to the total runtime of the workers (provided by the `time` Linux command line tool). Finally, the number of workers involved in the computation, which is reported either by Slurm or Condor. All the results are averaged over 5 independent runs.

TABLE II: Parallel performance indicators for mwNSGA-II

	LL_{DCC}	$\text{LL}_{Picasso}$	$\text{HH}_{Picasso}$
Wall-clock time (h)	1.29	0.90	2.10
Accumulated run time (h)	316.25	321.04	1015.09
Parallel efficiency (%)	98.52	95.32	99.28
Average number of workers	245.20	375.20	486.60

Let us first analyze the first two rows of the table. The mwNSGA-II algorithm has been able to reduce the computational time of the search from dozens of days to roughly 1-2 hours. This reduction is specially relevant in the HH instance in the Picasso platform, in which the algorithm computed 100000 function evaluations in 2.10 hours. Aggregating the runtime of all the 486 workers involved in the computation, that is, if only one single core had been used, it would have taken 42.30 days (1015.09 hours). It is remarkable that the results are consistent regardless of the underlying parallel computing platform, as mwNSGA-II reaches an equivalent performance on both DCC and Picasso, even with when a different number of workers is involved in the computation. Indeed, the accumulated runtime is almost the same (316.25 and 321.04 hours, respectively). To corroborate this fact, we have measured the average evaluation time of a single RCSO solution (the computation of the $H = 100$ samples) in the two platforms: it is 11.62 and 12.39 seconds in DCC and Picasso, respectively, for the LL instance, and 35.15 seconds for the HH instance in Picasso.

The truly interesting point is the parallel efficiency attained by mwNSGA-II. This indicator shows how the algorithm takes full advantage of the computational power provided by the platforms. It can be seen that the three experiments have reached a parallel efficiency over 95%, with special attention to the results of the HH instance in Picasso. In this case, mwNSGA-II has achieved a performance of 99.28%, involving almost 500 heterogeneous processing elements in the computation. This can be justified by two facts. On the one hand, as stated above, one single evaluation of the HH instance in the RCSO formulation takes 35.15 seconds on average and, taking into account that the interconnection network in Picasso uses 100 Gbps Infinibad fiber links, the ratio communication/computation is very favourable for the algorithm to reach such a performance. On the other hand, the algorithmic design of mwNSGA-II is able to profit from the parallel platform by avoiding bottlenecks in the master node, which is the most critical one in the master/worker paradigm to determine its performance. It can be therefore concluded that the communications are well overlapped by the computation and the load balancing mechanism is efficient.

2) What type of solutions does provide the RCSO problem?: Recall that the RCSO problem formulation has four objectives, namely the mean and standard deviation of the power consumption and the capacity of the UDN over $H = 100$ samples, thus visually inspecting the approximated Pareto fronts means plotting 4D data. We have addressed this issue by displaying boxes, in which the position of the center of the boxes is defined by the average values of the power, $\mu_{f_{Power}}$, and the capacity, $\mu_{f_{Cap}}$, whereas the width and the height are set up with the standard deviation, $\sigma_{f_{Power}}$ and $\sigma_{f_{Cap}}$. As the values of the objectives differ several orders of magnitude, the values shown are actually coefficients of variations in order to properly improve its readability. The idea is that, the higher the area, the lower the robustness. Figure 1 displays the result of a representative execution of mwNSGA-II for the HH instance using the described approach.

The conclusions that this figure shows are of great interest for the network operators. It can be seen that solutions with a low power consumption (the left hand side of the plot) tend to be more robust (the area of the squares are smaller) and, as long as the UDN has to provide higher capacity to the UEs (right part), solutions become more sensitive to the traffic data (less robust). That is, the UDN configurations that leave more SBS switched on are more sensitive to changes in the traffic demands, what is expected as UEs are random placed when computing the H disturbances and provokes stronger variations in the capacity as well, specially if they are located close to SBSs with a large available bandwidth. A second interesting finding is that solutions are much more robust with respect to the power consumption than with respect to the capacity. This is reflected as the width of the boxes is smaller than their height, that is, $\sigma_{f_{Power}}$ much smaller than $\sigma_{f_{Cap}}$. Indeed, the ratio $\frac{\sigma_{f_{Power}}}{\mu_{f_{Power}}} = 6.51e-9$ averaged over all the solution of the approximated front shown in Fig. 1,

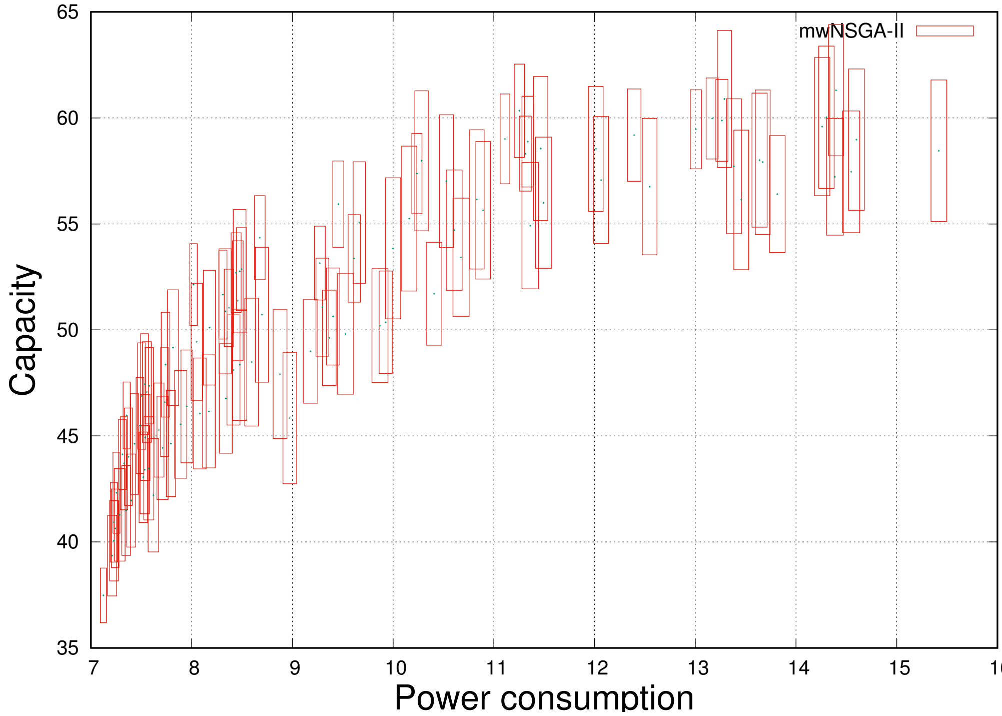

Fig. 1: A representative approximate Pareto front of the HH instance of the RCSO problem.

whereas the ratio $\frac{\sigma_{f_{Cap}}}{\mu_{f_{Cap}}} = 12.50$. In any case, this set of solutions provides the network operator with a highly valuable information on how rather close configurations may better react to unexpected traffic demands.

3) Sensitivity analysis: In order to evaluate the impact of introducing robustness in the CSO problem formulation, that is, the definition of the RCSO problem, this section develops a sensibility analysis of the solutions obtained by a sequential run of NSGA-II that only considers one traffic demand scenario. To do so, we have generated a completely different traffic demand pattern in which users follow the well known random waypoint mobility model. Under this new demand, the robust solutions, i.e., the switching off plan of small base stations, of the RCSO problem computed by mwNSGA-II and those in which robustness is not taken into consideration.

TABLE III: Average power consumption and capacity over a randomly generated mobile traffic pattern in the HH instance

	CSO	RCSO
Average power	1.55e-02	9.67e-03
Average capacity	1.39e+01	2.62e+01

Table III includes the average power and average capacity of 100 time steps of the UEs moving within the service area of the HH instance, that is, 100 different traffic demands. These averages are computed with the solutions provided by the CSO and the RCSO formulations of the problems. Here, the conclusions are also quite direct. The solutions provided by the RCSO problem reach not only a lower power consumption, but also a higher capacity. That is, if the problem of switching off SBSs of UDNs is addressed from a robust optimization approach, unexpected or missestimated traffic demands can be better serviced by the network, thus increasing the QoS and also the user experience.

V. Conclusions and Future Work

This work has addressed the robust version of the Cell Switch-Off problem that is, to the best of our knowledge, the very first time it has been defined in the literature. It encompasses the task of how the power consumption and the capacity of UDN networks vary when the traffic demand estimates have disturbances. In order to provide the solutions with robustness, the approach used lies in using Monte Carlo integration to measure their expected performance under different scenarios. The original bi-objective CSO problem has been extended to the four-objective RCSO problem in which both the mean and the standard deviation of the power consumption and the capacity of the network are to be optimized. The the standard deviation is the indicator that accounts for the solution robustness.

But the RCSO formulation has provoked a considerable increase in the computation time, because each solution has to be evaluated over H different traffic demands. To address this issue, this work has used a parallel version of the NSGA-II algorithm, mwNSGA-II, that follows a master-slave approach. The results over two different instances and on two different parallel computing platforms have provided very promising solutions. Indeed, the algorithm is able to reach more than 95% of parallel efficiency, and is able to reduce the total expected computation time of the algorithm from 42.3 days to 2.1 hours in the larger HH problem instance. As to the analysis of the solution quality, the solutions reached are more robust in regions of low power consumption configurations of the search space. The larger the capacity of the network, the lower the expected robustness. A sensibility analysis has also shown that the solutions of the RCSO problem are less sensitive to disturbances than those computed by the original CSO formulation.

Future lines of research lies in further providing the solutions with increasing robustness by designing specialized search operators that thoroughly explore search space. We also want to use realistic traffic information from cellular operators to evaluate the impact of the RCSO solutions with real-world data. The evaluation of other multi-objective metaheuristics and the impact of a higher values of the sampling ($H >> 100$) is also under research.

ACKNOWLEDGEMENTS

This work has been supported by the TIN2016-75097-P project of the Spanish National Program of Research, Development and Innovation. Francisco Luna also acknowledge support from Universidad de Málaga under grant PPIT.UMA.B12017/15. The authors thankfully acknowledges the computer resources, technical expertise and assistance provided by the SCBI (Supercomputing and Bioinformatics) center of the University of Málaga.

REFERENCES

[1] M. Feng, S. Mao, and T. Jiang, "Base station on-off switching in 5g wireless networks: Approaches and challenges," *IEEE Wireless Communications*, vol. 24, no. 4, pp. 46–54, 2017.

[2] 3GPP, "Small Cell Enhancements for E-UTRA and E-UTRAN—Physical Layer Aspects," 3rd Generation Partnership Project (3GPP), Tech. Rep., 2014. [Online]. Available: http://www.3gpp.org/ftp/Specs/html-info/36872.htm

[3] X. Ge, S. Tu, G. Mao, C.-X. Wang, and T. Han, "5G Ultra-Dense Cellular Networks," *IEEE Wireless Communications*, vol. 23, no. 1, pp. 72–79, feb 2016.

[4] D. Lopez-Perez, M. Ding, H. Claussen, and A. H. Jafari, "Towards 1 Gbps/UE in Cellular Systems: Understanding Ultra-Dense Small Cell Deployments," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2078–2101, 2015.

[5] M. Kamel, W. Hamouda, and A. Youssef, "Ultra-Dense Networks: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 4, pp. 2522–2545, 2016. [Online]. Available: http://ieeexplore.ieee.org/document/7476821/

[6] J. G. Andrews, S. Buzzi, W. Choi, S. V. Hanly, A. Lozano, A. C. K. Soong, and J. C. Zhang, "What Will 5G Be?" *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 6, pp. 1065–1082, jun 2014.

[7] P. Gandotra, R. K. Jha, and S. Jain, "Green Communication in Next Generation Cellular Networks: A Survey," *IEEE Access*, vol. 5, pp. 11 727–11 758, 2017. [Online]. Available: http://ieeexplore.ieee.org/document/7939957/

[8] W. Lai, C.-S. Shieh, C.-S. Ho, and Y.-R. Chen, "A clustering-based energy saving scheme for dense small cell networks," *IEEE Access*, vol. 7, pp. 2880–2893, 2019.

[9] J. Li, H. Wang, X. Wang, and Z. Li, "Optimized sleep strategy based on clustering in dense heterogeneous networks," *Eurasip Journal on Wireless Communications and Networking*, vol. 2018, no. 1, 2018.

[10] D. Gonzalez G., J. Hamalainen, H. Yanikomeroglu, M. Garcia-Lozano, and G. Senarath, "A Novel Multiobjective Cell Switch-Off Framework for Cellular Networks," *IEEE Access*, vol. 4, pp. 7883–7898, 2016.

[11] Q.-N. Le-The, T. Beitelmal, F. Lagum, S. S. Szyszkowicz, and H. Yanikomeroglu, "Cell Switch-Off Algorithms for Spatially Irregular Base Station Deployments," *IEEE Wireless Communications Letters*, vol. 6, no. 3, pp. 354–357, jun 2017.

[12] F. Lagum, Q.-N. Le-The, T. Beitelmal, S. S. Szyszkowicz, and H. Yanikomeroglu, "Cell Switch-Off for Networks Deployed With Variable Spatial Regularity," *IEEE Wireless Communications Letters*, vol. 6, no. 2, pp. 234–237, apr 2017.

[13] D. González González, E. Mutafungwa, B. Haile, J. Hämäläinen, and H. Poveda, "A Planning and Optimization Framework for Ultra Dense Cellular Deployments," *Mobile Information Systems*, vol. 2017, pp. 1–17, 2017.

[14] F. Luna, R. Luque-Baena, J. Martínez, J. Valenzuela-Valdés, and P. Padilla, "Addressing the 5g cell switch-off problem with a multi-objective cellular genetic algorithm," in *IEEE 5G World Forum, 5GWF 2018 - Conference Proceedings*, 2018, pp. 422–426.

[15] A. Salem, S. El-Rabaie, and M. Shokair, "Energy efficient ultra-dense networks (udns) based on joint optimisation evolutionary algorithm," *IET Communications*, vol. 13, no. 1, pp. 99–107, 2019.

[16] T. Beitelmal, S. Szyszkowicz, G. David González, and H. Yanikomeroglu, "Sector and site switch-off regular patterns for energy saving in cellular networks," *IEEE Transactions on Wireless Communications*, vol. 17, no. 5, pp. 2932–2945, 2018.

[17] M. Katsigiannis, *A demand model for mobile data traffic in the 5G Era: Case of Finland*. IGI Global, 2018.

[18] Z. He, G. Yen, and Z. Yi, "Robust multiobjective optimization via evolutionary algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 2, pp. 316–330, 2019.

[19] J. Branke, *Evolutionary Optimization in Dynamic Environments*. Kluwer, 2002.

[20] H.-G. Beyer and B. Sendhoff, "Robust optimization – a comprehensive survey," *Comput. Methods Appl. Mech. Engrg.*, vol. 196, no. 33 - 34, pp. 3190 – 3218, 2007.

[21] I. Paenke, J. Branke, and Y. Jin, "Efficient search for robust solutions by means of evolutionary algorihtms and fitness approximations," *IEEE Trans. on Ev. Comp.*, vol. 10, no. 4, pp. 405 – 420, 2006.

[22] X. Fei, J. Branke, and N. Gulpinar, "New sampling strategies when searching for robust solutions," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 2, pp. 273–287, 2019.

[23] F. Luna, G. R. Zavala, A. J. Nebro, J. J. Durillo, and C. A. C. Coello, "Distributed Multi-Objective Metaheuristics for Real-World Structural Optimization Problems," *The Computer Journal*, vol. 59, no. 6, pp. 777–792, 06 2016.

[24] M. Mirahsan, R. Schoenen, and H. Yanikomeroglu, "HetHetNets: Heterogeneous Traffic Distribution in Heterogeneous Wireless Cellular Networks," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 10, pp. 2252–2265, 2015.

[25] N. Piovesan, A. Fernandez Gambin, M. Miozzo, M. Rossi, and P. Dini, "Energy sustainable paradigms and methods for future mobile networks: A survey," *Computer Communications*, vol. 119, no. December 2017, pp. 101–117, 2018. [Online]. Available: https://doi.org/10.1016/j.comcom.2018.01.005

[26] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182 – 197, 2002.