A Set Cover Based Algorithm for Cell Switch-Off with Different Cell Sorting Criteria

Tamer Beitelmal and Halim Yanikomeroglu

Department of Systems and Computer Engineering, Carleton University, Canada
{tamer, halim}@sce.carleton.ca

***Abstract*—The traffic distribution in cellular networks fluctuates in both time and space. This fluctuation results in some base stations (cells) being underutilized in light traffic conditions. Despite being underutilized, these cells still consume substantial amount of their energy. One possible technique to preserve this wasted energy is implementing the Cell Switch-Off (CSO) approach. In this approach, a common practice is to switch off cells based on their current loads. However, not only the cell load affects the switch-off procedure but also the order in which the cells are switched off (cell sorting). Hence, in this paper, we investigated different cell sorting criteria. The results illustrated that more energy can be preserved when sorting cells based on the number of users they can serve compared with the case of sorting cells based on their current load. To implement the CSO approach, we proposed a centralized greedy-add algorithm devised from the well known set cover problem. Simulation results showed that our algorithm outperformed the benchmark algorithm when the number of users per cell is large. The two algorithms were compared using the Urban-Micro (UMi) evaluation scenario.**

***Index Terms*—Cell Switch-Off, Cell Sorting, Energy Saving, Green Communications, Capacitated Set Cover Problem, Un-splitable Demand, Dissimilar Demand.**

I. Introduction

Cellular networks have become the preferred mean to access the Internet due to its ability to provide ubiquitous data coverage. This has resulted in a significant increase in the demand for high data rate over cellular networks. To cope with this explosive demand, cellular operators are employing several novel and sophisticated techniques. One basic technique to increase the network capacity is deploying more base stations (cells). However, the exponential growth in the number of cells raises the concern regarding not only the operational costs but also the environmental impact [1]. Towards next generation green communication networks, Energy Efficiency (E2) became an important performance metric. Cellular networks are expected to rapidly grow and become a major contributor for the carbon emission. A significant amount, up to 80%, of the total energy consumption in cellular networks comes from cells [2]. Therefore, different solutions were proposed to reduce cell energy consumption including, implementing energy efficient hardware, utilizing solar energy sources, and reducing the number of active cells [1].

During the planning phase, cells are deployed in large numbers to cope with the peak traffic. Therefore, several cells become underutilized or even redundant outside the peak traffic. Despite being underutilized, these cells still consume a major portion of their energy. Therefore, Cell Switch-Off (CSO) approach was proposed to preserve energy by entirely switching off some cells. This switching off shouldn't compromise users' satisfaction.Implementing the CSO approach is a complicated problem. In particular, random or inappropriate switching off may deteriorate the overall system performance.

Several algorithms were proposed to implement the CSO approach and most of them rely on heuristic methods [3], [4], and [5]. Using heuristics is encouraged in such complicated scenarios as they provide good solutions in a timely manner. A simple greedy-drop algorithm was proposed in [3] and was dubbed *Cell-Zooming*. In this algorithm, cells are sequentially switched off based on their loads starting with the least loaded one. The algorithm terminates when it encounters the first cell that can not be switched off because one or more of its users can not be served by any of the neighboring cells. An improved version of this algorithm, *Improved Cell-Zooming*, was proposed in [4]; performance enhancement was obtained by slightly adjusting the termination criterion such that the algorithm does not terminate prematurely, rather it checks all the cells in the network for possible switch-off. Applying this new termination criterion resulted in more energy saving, by switching off more cells. A greedy-add algorithm was proposed in [5] to switch on as few cells as possible to accommodate all the demand. Cells are switched on based on their load and the algorithm terminates when every user is assigned to an active cell. Different other algorithms inspired from other fields of research were considered to tackle the CSO approach, i.e., a utility-based algorithm [6] and a genetic algorithm [4]. The common practice is to switch off cells based on their current load, i.e., [3], [4], and [5]. However, switching off cells does not depend only on the cell's own load but also on several factors such as the load of neighbor cells and the order in which cells are switched off (cell sorting).

In this paper, we investigated three different cell sorting criteria. Simulations indicated that by sorting cells based on the number of users they can serve, more cells can be

978-1-4799-4640-2/14/$31.00 ©2014 IEEE

switched off when compared with sorting cells based on their current load. The improved cell-zooming of [4] was used as a benchmark to compare the performance of our proposed algorithm. In order to properly implement the CSO approach, we investigated and successfully applied the well known set cover problem to the context of CSO. We proposed a greedy-add algorithm devised from the set cover problem after applying the necessary modification to comply with the special characteristics of CSO. To the best of our knowledge, the set cover problem has not been used in the context of CSO. Results showed that the proposed algorithm outperformed the benchmark algorithm of [4] when the number of users per cell is 10 or more.

The contributions of this paper are as follows:

- Investigate different cell sorting criteria (the order in which cells are switched on) and compare their impact on energy saving.
- Implement the CSO approach as a set cover problem and provide a greedy-add algorithm that provides a good solution.

The set cover problem is introduced in Section II. In Section III, problem formulation and the proposed algorithm are described. The evaluation model and the results are shown in Section IV. Section V concludes this paper.

II. Set Cover Problem and its Applications in Wireless Networks

The set cover problem is usually denoted as $(\mathcal{U}, \mathcal{S}, \mathcal{C})$, where $\mathcal{U}$ is a universe of n elements, $\mathcal{S} = \{S_1, S_2, ..., S_m\}$ is a set of m subsets from $\mathcal{U}$, and $\mathcal{C} = \{c_1, c_2, ..., c_m\}$ is the set of the cost associated to each subset $S_k \in \mathcal{S}$. A *set cover* is a collection of subsets from $\mathcal{S}$ such that all elements in $\mathcal{U}$ are included in at least one subset. The objective of the set cover problem is to find the set cover $\mathcal{S}^* \subseteq \mathcal{S}$ that minimizes the cost [7]. If all the subsets have the same cost, the problem becomes $(\mathcal{U}, \mathcal{S}, 1)$ that is the unweighted case. The objective of the unweighted set cover problem is to find the minimum collection of subsets $\{S_1, S_2, ..., S_m\}$ that covers all elements of $\mathcal{U}$.

There are two versions of the problem, un-capacitated and capacitated set cover. In the un-capacitated version, every subset S_k is assumed to have unlimited resources and can serve all the elements in the set. While in capacitated set cover, each element i has a demand b_i and each subset has limited resources to distribute among its elements.

The set cover problem is a fundamental problem in combinatorial optimization, therefore, it has a wide range of applications. In the rest of this section its relevant applications in different wireless networks are highlighted.

In wireless mesh networks, the set cover problem was used in the planning phase. The objective was to optimally locate the gateways in order to minimize the initial cost while covering all the routers [8]. It was also used to solve the problems of broadcasting [9] and energy saving [10] in ad hoc networks. Minimizing the interference is another application that was applied in both Wireless Sensor Networks (WSNs) [11] and cellular networks [12]. Nevertheless, the common application of the set cover problem in wireless networks is to solve the coverage problem in WSNs [13]. According to the survey paper [10], the coverage problem in WSNs is classified into two categories: area coverage and discrete point coverage. The objective in the former category is to monitor an area, while in the latter is to cover a set of points.

The set cover problem in WSNs is somehow similar to the CSO approach. Hence the objective in both is to cover some points with the minimum number of sets. This objective corresponds to energy saving. However, implementing the CSO approach as a set cover problem can not follow the exact procedure as in WSNs. This is because of three major differences that should be carefully addressed. The first and the key difference is the capacity constraint. While in CSO the cell capacity is a vital factor, it is not a real constraint in WSNs as only low data rates are transmitted. The second difference is the coverage pattern. The coverage area in WSNs is usually assumed to be circular [13], however, in the CSO, it is not circular due to shadowing effect caused by obstacles, reflection and diffraction. The third difference is the need for energy saving mechanism. While being a must (power scarcity) in WSNs, it is a need (power saving) in cellular networks.

III. Problem Formulation

The objective of the CSO approach is to switch off as many cells as possible while providing users with their required rates. Our formulation is based on the observation that this objective can be obtained by using the set cover problem. Moreover, there are several algorithms to solve the set cover problem and they provide a very good solution with polynomial time complexity [7]. The set cover formulation is a consistent strategy to implement the CSO approach, while its available solutions can be modified to address the special characteristics of the CSO. The results shown later on, proves the efficacy of the proposed CSO algorithm based on a set cover formulation. To the best of our knowledge, the set cover problem was not used in the context of CSO.

The CSO approach is formulated as a $(\mathcal{U}, \mathcal{S}, 1)$ set cover problem, where $\mathcal{U}$ is the set of User Equipments (UEs) in the network, $\mathcal{S}$ is the subset of UEs belongs to each cell, and 1 corresponds to the unweighted set cover. In the CSO context, unweighted means that all cells have the same cost, where the cell cost is equals to the consumed power when the cell is active[1].

One possible formulation for the CSO as a set cover problem is the following:

[1]This formulation can be easily extended to the case of cells with different costs, i.e., HetNets scenarios with macro, micro, or pico cells.

$$\text{minimize} \quad \sum_{j \in \mathcal{S}} y_j \tag{1a}$$

$$\text{subject to} \quad \sum_{j \in \mathcal{S}} x_{ij} y_j = 1, \quad \forall i \in \mathcal{U}, \quad \forall j \in \mathcal{S} \tag{1b}$$

$$x_{ij}, y_j \in \{0, 1\}, \quad \forall i \in \mathcal{U}, \quad \forall j \in \mathcal{S}. \tag{1c}$$

The following notations are introduced to be used in the problem formulation and algorithm description:

y_j : binary variable, $= 1$ if cell j is active.

x_{ij} : binary variable, $= 1$ if UE i is connected to cell j.

$\mathbf{X} = [x_{ij}]$, UE to cell assignment matrix.

r_i : minimum required rate for UE i.

ω_{ij} : spectral efficiency between UE i and cell j.

W_j : total bandwidth of cell j.

$b_{ij} = \dfrac{r_i}{\omega_{ij}}$, required bandwidth for UE i if served by cell j.

$\mathbf{B} = [b_{ij}]$, required bandwidth matrix.

N_j : set of UEs covered by cell j.

M_j : set of UEs currently served by cell j.

S_j : set of UEs can be served by cell j.

$\mathbf{V}$: set of connected/served UEs.

$\mathbf{L}$: set of active cells.

This un-capacitated set cover problem (1) can be solved by a simple greedy-add algorithm, Algorithm 2.2 of [7], that provides a good solution. Despite the similarities, CSO approach has distinct characteristics that should be considered when implementing it a set cover problem. To accommodate these characteristics, we made three modifications to the simple greedy-add algorithm.

The first modification is the following capacity constraint

$$\sum_{i \in \mathcal{U}} b_{ij} x_{ij} \leq W_j, \quad \forall j \in \mathcal{S}. \tag{2}$$

The input of the original algorithm is the cover set that is the set of all UEs that can be covered by each cell. Instead of complicating the formulation in (1) by adding an extra constraint, the constraint in (2) is enforced by constructing a *service set* which is a modified cover set to be used as an input to the algorithm. Only a subset of UEs covered by cell j will be included in its service set S_j such that the sum of the demand does not exceed the total bandwidth W_j. In other words, a cell may not be able to serve all UEs in its cover set, i.e., the service set is a subset of the cover set. Using the service set as an input instead of the cover set allows the use of Algorithm 2.2 of [7] that is originally for un-capacitated set cover. The term served UEs is used instead of covered UEs to emphasize the capacity aspect.

The second modification is regarding the possibility of splitting the demand. If the demand of a UE can be satisfied by more than one cell, then the demand is referred to as *splittable* demand. In the original set cover formulation, the splitable demand is introduced by the constraint

$$\sum_{j \in \mathcal{U}} x_{ij} y_j \geq 1, \quad \forall i \in \mathcal{U}, \quad \forall j \in \mathcal{S}. \tag{3}$$

However, in the CSO approach, the demand of any UE must be satisfied by a single cell, that is referred to as *un-splittable* demand. The modification was reflected by replacing constraint (3) by constraint (1b) in the CSO formulation (1).

The third modification was due to the type of the demand. In capacitated set cover, UE i requires the same demand b_i from any cell. However in the context of cellular networks, the demand of a UE i, required bandwidth, differs based on which cell is providing the service, we referred to it as *dissimilar* demand. Therefore the notation b_{ij} is needed to differentiate between the different required bandwidth from each cell j. Although there are algorithms to solve the capacitated set cover [14], we were not able to utilize them because they are very specific and not designed for dissimilar demand.

The Proposed Algorithm

In this paper, we proposed a two-stage centralized algorithm. The first stage, described in Algorithm 1, is used to obtain the service set that enforces the capacity constraint of (2). To find the service set S_j for cell j, first, the UEs from M_j are added to S_j, then the cell bandwidth W_j is arbitrary filled up by adding new UEs from N_j starting with UE i^* that requires the least b_{i^*j}.

Algorithm 1: Obtain Service Set

input : $\mathbf{B}$, $\mathbf{X}$, $\mathbf{V}$, W_j
output : $\mathcal{S} = \{S_1, S_2, ..., S_m\}$ service set $\forall j \in S$

for each cell j **do**
  $S_j \leftarrow S_j \cup \{M_j \backslash \mathbf{V}\}$
  $Z = \sum_{\forall r \in S_j} b_{rj}$, utilized bandwidth
  while $Z < W_j$ **do**
    find $i^* = \text{argmin}_{i^* \in \{N_j \backslash \mathbf{V}\}} b_{i^*j}$
    if $Z + b_{i^*j} < W_j$ **then**
      $Z \leftarrow Z + b_{i^*j}$
      $S_j \leftarrow S_j \cup \{i^*\}$
      else
        Terminate while loop
      end
    end
  end
end
return $\mathcal{S}$

The algorithm terminates after obtaining the service sets $S = \{S_1, S_2, ..., S_m\}$, where m is the number of cells in the network.

Algorithm 2: Greedy_add Algorithm

input :
M_j , $\mathbf{X}$, $\mathbf{B}$, Ω
$\mathcal{U} = \{e_1, e_2, ..., e_n\}$, the set of all UEs
output :
L, X

1 $\mathbf{L} \leftarrow \phi$, the set of active cells
2 $\mathbf{V} \leftarrow \phi$, the set of connected UEs
3 **while** $V \neq \mathcal{U}$ **do**
4 Call Algorithm 1, to obtain service set $\mathcal{S}$
5 **switch** *Sorting Criteria* **do**
6 **case** *MaxLoad*
7 find $j^* = \text{argmax}_{j^* \in \{S \backslash \mathbf{L}\}} (\sum_{\forall i \in \{M_{j^*} \backslash \mathbf{V}\}} b_{ij^*})$
8 **end**
9 **case** *MaxUsers*
10 find $j^* = \text{argmax}_{j^* \in \{S \backslash \mathbf{L}\}} |S_{j^*}|$
11 **end**
12 **case** *MaxCentres*
13 $T_j = \{a \in \{M_j \backslash \mathbf{V}\} | \omega_{aj} \geq \omega_{th}\} \quad \forall j \in S$
14 find $j^* = \text{argmax}_{j^* \in \{S \backslash \mathbf{L}\}} |T_{j^*}|$
15 **end**
16 **endsw**
17 $\mathbf{L} \leftarrow \mathbf{L} \cup \{j^*\}$
18 $\mathbf{V} \leftarrow \mathbf{V} \cup \{S_{j^*}\}$
19 Update **X**
20 **end**
21 return **L, X**

The second stage, the main algorithm, is described in Algorithm 2. The input of this Algorithm is the obtained set of service sets, S, from Algorithm 1. This algorithm is a modified version of the simple greedy-add algorithm for the un-capacitated set cover, Algorithm 2.2 of [7]. The algorithm starts with the assumption that all cells are switched off and all UEs are unconnected. In each iteration, the algorithm selects a cell j^* to be switched on. The order in which cells are selected (cell sorting) highly impacts future cells switch-off decisions. The common practice is to select a cell based on its current load. However, switching off a cell does not depend solely on the cell's own load but also on several factors such as the available bandwidth of neighbor cells and the channel quality between its UEs and other cells. Therefore, we investigated three different cell sorting criteria and analyzed their effect on the final number of switched off cells. The cell selection procedure is explained in Steps 6-16 in Algorithm 2. Furthermore, the three different cell sorting criteria are introduced as follows:

1) **MaxLoad**: In this case, the next cell to select is the cell with the highest load. Selecting cells based on their loads is the common practice in literature.
2) **MaxUsers**: In this case, the next cell to select is the cell that can serve the maximum number of unconnected UEs.
3) **MaxCentres**: In this case, the next cell to select is the cell that has the maximum number of centre UEs (UEs with good channel quality). This case emphasis on selecting the cell that has the large number of UEs with good channel quality. Having a good channel duality with a certain cell implies your channel is not as good with other cells, hence requires relatively more resources to satisfy its demand. A UE i is a centre UE for cell j if $\omega_{ij} \geq \omega_{th}$, where ω_{th} is the spectral efficiency of a central UE and is assumed to be equal to 10 bps/Hz.

After selecting a cell j^* to switch on, all UEs on its service set, $\{S_{j^*}\}$ are added to the set **V** and the assignment matrix **X** is updated accordingly. The service set is updated at each iteration, by calling Algorithm 1, based on the updated values of **V** and **X**. The algorithm terminates when all UEs are connected. Finally, the cells in set **L** will stay active while all other cells are switched off.

Table I
Simulation Parameters

Cellular layout	square
Inter-site distance	200 m
Antenna pattern	omni directional
Cell transmitted power	41 dBm
Bandwidth	10 MHz
Carrier frequency (f_c)	2.5 GHz
UE distribution	random and uniform
Probability of indoor UEs	50%
Number of users per cell	5, 10, 15, 20, 25
Required rate r_i	500 kpbs
UE noise figure	5 dB
Thermal noise	-174 dBm/Hz
Shadowing standard deviation LOS	4 dB
Shadowing standard deviation NLOS	6 dB
Traffic type	full queue

IV. Evaluation Model and Results

In this paper we did the down-link analysis for a cellular layout with 100 square cells with omni-directional antenna pattern. We simulated the Urban Micro-cell (UMi) scenario that represents a small cell environment according to the evaluation guideline of [15]. In UMi scenario, the line of sight (LOS) signal is not the common path in this environment because of buildings. The pathloss calculation is based on the two pathloss models: LOS or Non-LOS (NLOS) and according to the probability of each of them [15]. The frequency reuse is 1, i.e., the whole spectrum is used in each cell. The necessary simulation parameters are listed in Table I. The inter-site distance (cell size) and the transmitted power correspond to micro-cell values. The number of UEs per cell and their required rate are to selected to create a lightly loaded network which is the typical case for CSO approach. For highly loaded networks, all the cells must be active and hence the CSO approach is not applicable.

In our simulation, we assumed that the cover set of each cell includes all the UEs in the area. This assumption is valid because of the natural dense cell deployment of UMi scenario. To make the problem tractable, we assumed that the Inter-Cell Interference (ICI) is managed by some certain interference management techniques[2]. Therefore, the spectral efficiency (ω_{ij}) was calculated based on the Signal-to-Noise Ratio value between UE i and cell j, SNR_{ij}, using the equation

$$\omega_{ij} = \log_2(1 + SNR_{ij}). \tag{4}$$

The proposed algorithm represent a centralized approach where all cells are connected to a central entity (cloud) that has global information about SINR values between all UEs and all cells. We ran the simulation for 100 different realizations. In each realization, UEs are dropped randomly in the area. The average number of switched off cells was obtained by taking the average over the 100 realizations. The energy saving is linearly proportional to the number of switched off cell. This is because cells are consuming significant amount of energy even if they are not transmitting, therefore the energy consumed for transmitting can be neglected. In this layout of 100 cells, the percentage of energy saving is equal to the number of switched off cells, i.e., switching off 30 cells corresponds to 30% energy saving.

The proposed algorithm was compared with the benchmark, greedy-drop, algorithm from [4], and the results were illustrated in Figure 1. This figure presents the energy saving, Y-axis, for different number of UEs per cell, X-axis. For the sake of fair comparison, we ran both algorithms using the same cell sorting criterion, *MaxLoad*, based on their loads. For a very small number of UEs per cells, 5, the benchmark algorithm performed slightly better. However, when the number of UEs per cell increases to 10 or more, our algorithm outperformed the benchmark algorithm by achieving up to 20% more energy saving in the case of 25 UEs per cell. This improvement is credited to the advantage of the greedy-add approach over the greedy-drop one. In the greedy-drop approach, the UEs of a cell are handed over before switching it off. This may not result in utilizing 100% of the bandwidth of other cells. Hence, the focus is to get red of the cell load and not intended to maximize the load in some cells. On the other hand, the greedy-add approach concentrates the load in any cell before switching it on by trying to load it to the maximum. This maximum cell loading resulted in increasing the number of switched off cells when using our greedy-add algorithm.

Figure 2 demonstrated the energy saving when applying the different cell sorting criteria, MaxLoad, MaxUsers and MaxCentres. As shown in the figure, the selection of cell sorting criterion affects the achieved energy saving. Both

[2]The extension is straight forward when the ICI is considered and the results will be qualitatively very similar as in Figures 1 and 2.

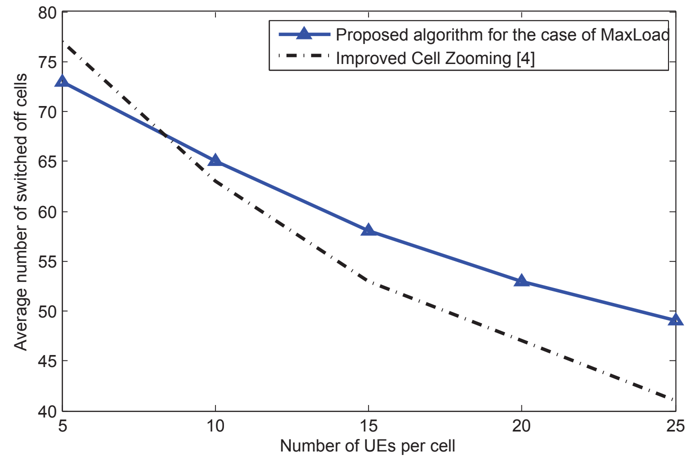


Figure 1. Energy saving for the proposed algorithm vs the benchmark algorithm from [4] using MaxLoad cell sorting criterion.

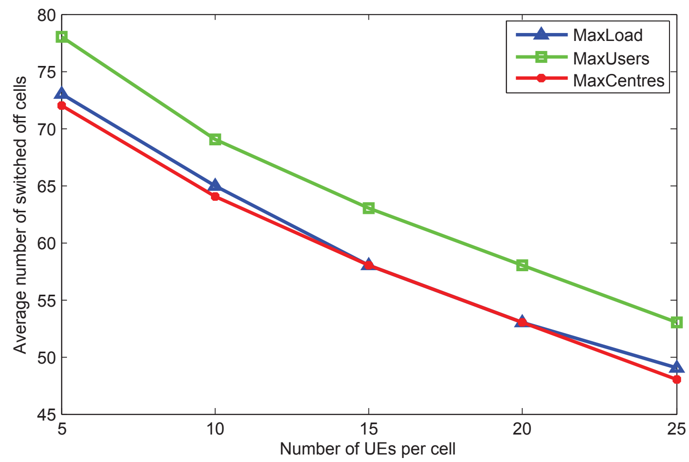
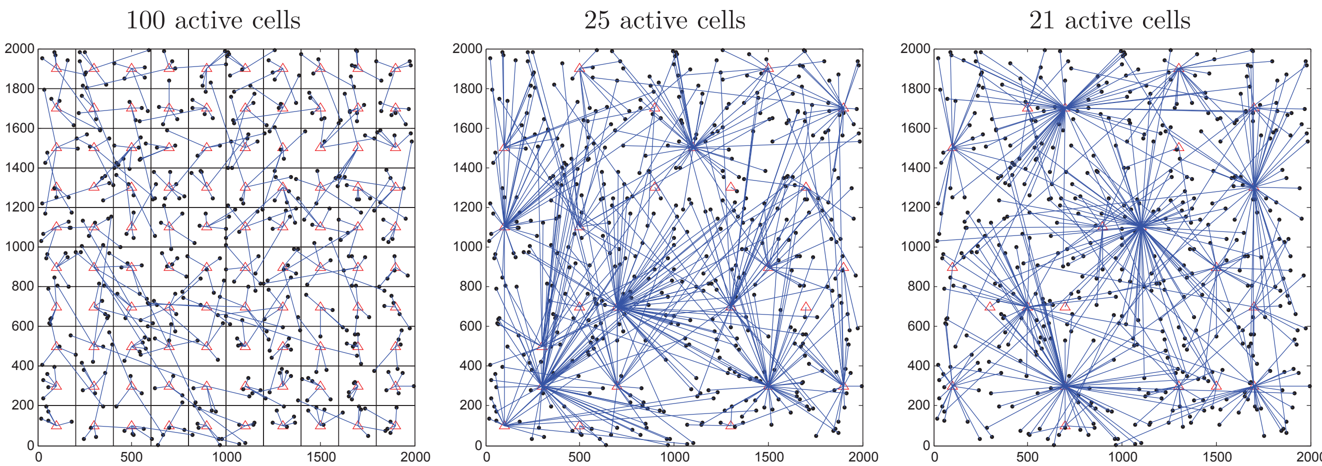


Figure 2. Energy saving for different cell sorting criteria.

MaxLoad and MaxCentres criteria performed very close to each other regardless of the number of UEs per cell. However, the MaxUsers criterion outperformed the two other sorting criteria in terms of energy saving. A gain of about 5% was constant through all the different number of UEs per cell. This improvement can be explained as: in MaxUsers, the next cell to switch on is determined based on the number of UEs that a cell can serve. Thus the selection, includes the future contribution of this particular cell to accommodate unconnected UEs. However, in MaxLoad, the selection is based on the current load of the cell without any indication of how much it can contribute for the unconnected UEs. In the latter case, the selected cell might not be the best cell to select as it can not contribute much for the unconnected UEs.

Figure 3 is a sample for the case of 5 UEs per cell to demonstrate the impact of cell sorting criterion on the cell switch-off procedure and the resulted UE to cell assignments. Figure 3a shows the initial UE to cell assignment before applying the proposed CSO algorithm. Figures 3b



(a) Initial UE to cell assignment. (b) UE to cell assignment using MaxLoad. (c) UE to cell assignment using MaxUsers.

Figure 3. The impact of cell sorting criterion on the UE to cell assignments, a sample for the case of 5 UEs per cell.

and 3c illustrated the difference in UE to cell assignment when using different sorting criteria, namely, MaxLoad and MaxUsers, respectively. As shown in Figure 3 , the traffic concentration as well as the set of switched on cells highly depend on the cell sorting criterion.

V. Conclusion

The CSO is a promising approach for energy saving in cellular networks by switching off some appropriate cells. Most of the existed algorithms for implementing the CSO approach switch off cells based on their current load. However, the switch-off procedure is affected by the order in which cells are switched off (cell sorting). Therefore, in this paper, we investigated three different cell sorting criteria and compared their impact on the total energy saving. Simulations showed that switching off cells based on the number of UEs they can serve provided the best performance among the investigated cell sorting criteria.

Furthermore, we formulated the CSO approach as a set cover problem. Using this formulation, we proposed a greedy-add algorithm for the CSO approach. This proposed algorithm was shown to outperform the benchmark algorithm, Improved Cell-Zooming, when the number of UEs per cell is large.

Acknowledgment

The authors would like to thank David Gonzalez G, Universitat Politecnica de Catalunya, Spain, and Yaser Fouad, Carleton University, Canada, for their valuable feedback.

Tamer Beitelmal's PhD is funded by the Ministry of Higher Education and Scientific Research (MOHESR), Libya, through the Libyan-North American Scholarship Program.

References

[1] Z. Hasan, H. Boostanimehr, and V. Bhargava, "Green cellular networks: A survey, some research issues and challenges," *IEEE Communications Surveys Tutorials*, vol. 13, no. 4, pp. 524–540, Fourth Quarter 2011.

[2] I. Humar, X. Ge, L. Xiang, M. Jo, M. Chen, and J. Zhang, "Rethinking energy efficiency models of cellular networks with embodied energy," *IEEE Network*, vol. 25, no. 2, pp. 40–49, Mar. 2011.

[3] Z. Niu, Y. Wu, J. Gong, and Z. Yang, "Cell zooming for cost-efficient green cellular networks," *IEEE Communications Magazine*, vol. 48, no. 11, pp. 74–79, Nov. 2010.

[4] F. Alaca, A. Bin Sediq, and H. Yanikomeroglu, "A genetic algorithm based cell switch-off scheme for energy saving in dense cell deployments," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Dec. 2012, pp. 63–68.

[5] S. Zhou, J. Gong, Z. Yang, Z. Niu, and P. Yang, "Green mobile access network with dynamic base station energy saving," in *Proc. ACM MobiCom*, vol. 9, no. 262, Sept. 2009, pp. 10–12.

[6] A. Yildiz, T. Girici, and H. Yanikomeroglu, "A pricing based algorithm for cell switching off in green cellular networks," in *Proc. IEEE VTC2013-Spring*, June 2013.

[7] V. V. Vazirani, *Approximation Algorithms*. Springer, 2001.

[8] X. Xu, S. Tang, X. Mao, and M. Li, "Distributed gateway placement for cost minimization in wireless mesh networks," in *Proc. IEEE 30th International Conference on Distributed Computing Systems (ICDCS)*, June 2010, pp. 507–515.

[9] S. Agathos and E. Papapetrou, "On the set cover problem for broadcasting in wireless ad hoc networks," *IEEE Communications Letters*, vol. 17, no. 11, pp. 2192–2195, Nov. 2013.

[10] M. Cardei and J. Wu, "Energy-efficient coverage problems in wireless ad-hoc sensor networks," *Computer Communications*, vol. 29, no. 4, pp. 413–420, Feb. 2006.

[11] H. Aslanyan and J. Rolim, "Interference minimization in wireless networks," in *Proc. IEEE/IFIP 8th International Conference on Embedded and Ubiquitous Computing (EUC)*, Dec. 2010, pp. 444–449.

[12] F. Kuhn, P. von Rickenbach, R. Wattenhofer, E. Welzl, and A. Zollinger, "Interference in cellular networks: The minimum membership set cover problem," in *Proc. 11th Computing and Combinatorics Conference (COCOON)*, vol. 3595. Springer, Sept. 2005, pp. 188–198.

[13] S. Jenjaturong and C. Intanagonwiwat, "A set cover-based density control algorithm for sensing coverage problems in wireless sensor networks," in *Proc. IEEE 3rd International Conference on Cognitive Radio Oriented Wireless Networks and Communications (CrownCom 2008)*, May 2008, pp. 1–6.

[14] N. Bansal, R. Krishnaswamy, and B. Saha, "On capacitated set cover problems," in *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*. Springer, Aug. 2011, pp. 38–49.

[15] ITU-R, "Report ITU-R M.2135-1; Guidelines for evaluation of radio interface technologies for IMT-Advanced," 2009.