Author manuscript, published in "Internationale Network Optimization Conference (INOC) 6701 (2011) 484-489"
DOI : 10.1007/978-3-642-21527-8_53

Energy saving in fixed wireless broadband networks*

David Coudert
Mascotte, INRIA, I3S, CNRS, Université de Nice Sophia
Sophia Antipolis, France
david.coudert@inria.fr

Napoleão Nepomuceno
Institut for Matematik og Datalogi
Syddansk Universitet, Campusvej 55, DK-5230 Odense
napoleao@imada.sdu.dk

Issam Tahiri
Mascotte, INRIA, I3S, CNRS, Université de Nice Sophia
Sophia Antipolis, France
issam.tahiri@inria.fr

April 21, 2011

Abstract

In this paper, we present a mathematical formulation for saving energy in fixed broadband wireless networks by selectively turning off idle communication devices in low-demand scenarios. This problem relies on a fixed-charge capacitated network design (FCCND), which is very hard to optimize. We then propose heuristic algorithms to produce feasible solutions in a short time.

1 Introduction

Fixed broadband wireless communications is a sector of the communication industry that holds great promise for delivering private high-speed data connections [1]. Such network comprises remote locations, each of them served by a radio base station (RBS), connected by means of high-capacity microwave radio links. A bidirectional link connecting two RBSs requires a dedicated pair of outdoor units (ODUs), each one directly coupled to a high-directional antenna.

Commonly, in this context, the network is built in a robust fashion to guarantee fault protection and to support the extremely bursty traffic behaviors. As a drawback, since ODUs consume substantial power whenever the link is up, it brings forth important energy waste to provide extra resources which could be used only in critical situations. Therefore, the traffic fluctuation over the time offers an opportunity to energy savings by handling traffic efficiently and turning off devices used to keep microwave radio links whose capacities are underused.

In this work, we consider the problem of deciding both the network's configuration and flows that minimize the total energy expenditure. Particularly, by configuration, we mean the choice of which communication devices we need to keep on to successfully meet the traffic requirements.

*This work has been partially supported by project APRF RAISOM (PACA & FEDER), ANR DIMAGREEN, and Villum Kann Rasmussen foundation.

This problem relies on a fixed-charge capacitated network design (FCCND), which is very hard to optimize [6]. Among others, [4] and [7] tackled similar problems on different networks. We present an exact formulation for this problem and propose heuristics that may be employed to produce good feasible solutions in a short time.

2 Problem modeling and linear formulation

The network topology is modeled by a digraph $H = (V, E)$ where every node $v \in V$ represents a base station and every arc $vw \in E$ represents a radio link. Every link has a capacity c_{vw} and can be either active (while consuming energy) or not. Traffic demands are defined by $|D|$ pairs (s^d, t^d), with $s^d, t^d \in V$ and by an average volume per demand h^d. We assume that H is symmetric since in the type of studied networks, radio links are usually symmetric. This implies that for every node $v \in V$, the entering neighborhood is the same as the leaving neighborhood, i.e. $\delta^+(v) = \delta^-(v) = \delta(v)$. Also the cost of an active link is constant and equal to CL (as shown in [5]). Another assumption that we consider in this work, and which is not always true, is the possibility of routing the traffic of the same demand d through different paths from s^d to t^d (multi-routing).

The problem can be formulated as a mixed integer program (MIP). We define two types of variables: to represent the state of link vw we consider a binary decision variable, being equal to 1 if the link is active and 0 otherwise. Since symmetric links must be in the same state, and in order to reduce the total number of binary variables, we use a single variable u_{vw} with $v < w$ (assuming some ordering of the nodes) for the pair of symmetric links vw and wv. We also employ a variable x^d_{vw} to indicate the volume fraction of the demand d which is routed through the link vw. In the MIP formulation, Eq. (1) is the objective function, Eq. (2) are the capacity constraints on the links, and Eq. (3) are the flow conservation constraints.

$$\min \sum_{vw \in E} CL \cdot u_{vw} \tag{1}$$

$$s.t. \sum_{d=1}^{D} x^d_{vw} \leq c_{vw} u_{vw} \qquad \forall vw \in E \tag{2}$$

$$\sum_{w \in \delta(v)} x^d_{wv} - \sum_{w \in \delta(v)} x^d_{vw} = \begin{cases} -h^d, & \text{if } v = s^d, \\ h^d, & \text{if } v = t^d, \\ 0, & \text{otherwise} \end{cases} \qquad \forall v \in V, \forall d = 1, \ldots, |D| \tag{3}$$

$$x^d_{vw} \in [0, h^d] \qquad \forall vw \in E, \forall d = 1, \ldots, |D| \tag{4}$$

$$u_{vw} \in \{0, 1\} \qquad \forall vw \in E \tag{5}$$

3 Hybrid algorithm

The model cited in the previous section is a mixed integer linear program. Even though it can be handled by a solver like “CPLEX”, this may take a very long time on relatively large networks (containing more than a hundred nodes). The number of variables and constraints can be huge and not even fit in memory. For such networks, we built a hybrid solution by combining a heuristic based on simulated annealing and a linear program with real variables (Multi Commodity Flow - MCF). The former would be the master and on every iteration it chooses the links to turn on/off. The latter, which is the slave, will only find out whether there is a feasible solution with this configuration or not (see Figure 1). Actually the linear program

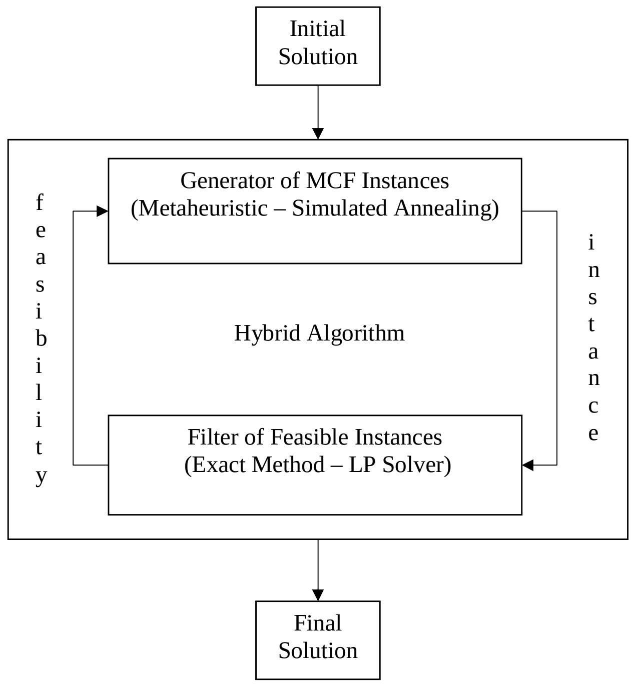


Figure 1: The framework of the hybrid algorithm

will have the same formulation except that the link state variables u_{vw} will now be constant. Therefore there will be no more integer variables in the program, which makes it faster to solve.

At each iteration of the simulated annealing process, a new network configuration is generated from the current solution by switching the state (Up➔Down or Down➔Up) of a couple of symmetric radio links. This couple is randomly chosen. Then, we apply the filter to check if this new configuration leads to a feasible solution. In the affirmative case, we attribute the total power utilization as energy score value of this solution. Otherwise, we discard this infeasible solution. We then follow the original description of the simulated annealing process [9] , where the algorithm replaces the current solution by the new solution with a probability that depends on the difference between the corresponding energy score values and a global parameter T (called the temperature).

4 Sparse cuts based algorithm

As an alternative approach, we propose a heuristic to construct a good configuration. It is also hybrid since the LP solver is still responsible for testing the feasibility of a configuration. This method is based on the sparse cuts of the graph. Roughly speaking, a cut is considered sparse if it has a high average load per link. Since this notion is mostly studied in the context of undirected graphs, we will consider for this heuristic the underlying undirected graph of our initial symmetric digraph.

Let us define the sparsity more formally. Let $G = (V, E)$ be the undirected graph. Let D be the set of demands (pairs of vertices). Let c be the capacity function that given a subset of edges returns the sum of capacities of all edges in this subset. Let also h be the function that for a given subset of demands returns the sum of volumes of demands in this subset. For any cut separating S (a subset of V) from $\bar{S}$, let $\alpha(S)$ denote the sparsity of this cut, with:

Algorithm 1 Maximize the number of shutdown links

Require: $G_0 = (V, E)$ is the initial graph
Require: c is a capacity function on the edges
Require: $cutsList$ is the list of precomputed sparse cuts
Require: D is the set of demands
Ensure: $G' = (V, E')$ with $E' \subseteq E$ is a subgraph ensuring the routing of all demands.
  $nonPotentialLinks \Leftarrow \emptyset$
  $E' \Leftarrow E$
  while $nonPotentialLinks \neq E$ **do**
    SORT($cutsList$) in descending order of average load per link in up state
    $potentialList \Leftarrow \emptyset$
    for i from 1 to SIZE($cutsList$) **do**
      $potentialList \Leftarrow potentialList|$LINKS$(cutsList[i])$
    $potentialList \Leftarrow potentialList|(E - (potentialList \cup nonPotentialLinks))$
    for i from SIZE($potentialList$) downto 1 **do** {in LIFO order}
      if ROUTING_LP$((V, E' - potentialList[i]), c, D)$ is feasible **then**
        $E' \Leftarrow E' - potentialList[i]$
        $nonPotentialLinks \Leftarrow nonPotentialLinks \cup potentialList[i]$
        break
      else
        $nonPotentialLinks \Leftarrow nonPotentialLinks \cup potentialList[i]$

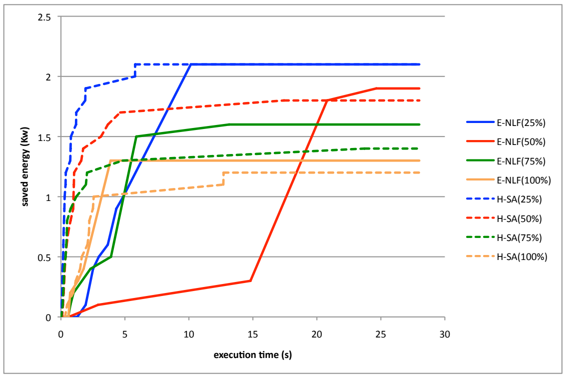
$\alpha(S) = c(E(S, \bar{S}))/h((S * \bar{S}) \cap D)$. If we consider a uniform version in which all edges have capacity 1 and to every pair of vertices corresponds a demand with the same volume 1, then the sparsity is $\alpha(S) = |E(S, \bar{S})|/|S| * |\bar{S}|$.

As said before, our heuristic is based on sparse cuts; those which have a small sparsity. Finding the minimum-sparsity cut, also called the sparsest cut, is a well-known combinatorial optimization problem in the literature. Although it is NP-complete, many work targeting improved approximations have been carried out [11, 10, 3, 8], and recently an approximation that relies on a semi definite program (SDP) relaxation to achieve a $\sqrt{\log n}$ approximation factor has been proposed [2]. In order to obtain many different sparse cuts, we run the approximation many times while adding each time a small randomness in the objective function of the SDP.

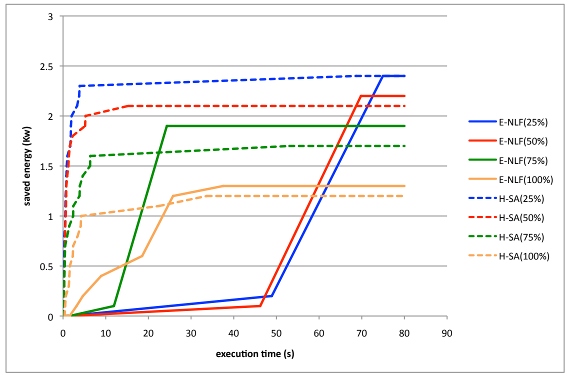
The motivation behind using sparse cuts is the fact that the sparser the cut is, the more loaded its edges tend to be, and the less efficient deleting one of them would be. So every link will have an estimated load equal to the average load per link of the sparsest cut covering this edge. The heuristic will try to remove the links in ascending order of estimated load. At each iteration, the feasibility of routings will be checked with a linear program and, after removing a link, the sparsity of the cuts will be updated. An efficient way to implement this heuristic is described in Algorithm 1 which performs $\Omega(m)$ calls to the LP solver (m being the number of edges).

5 Simulation results

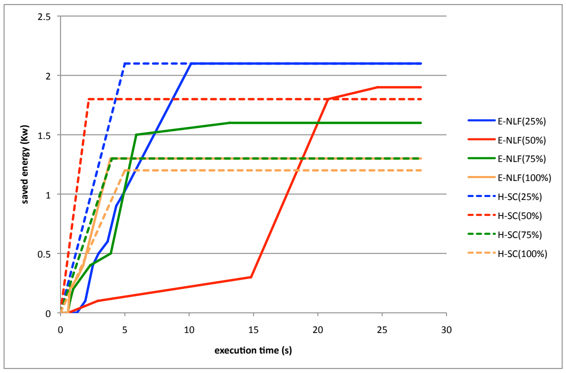
We will follow the same scenario for the different approaches previously explained: the exact formulation using the linear program, the hybrid algorithm, and the sparse cuts based algorithm. The simulations have all been executed on SNDlib topologies which represent backbones (France with 45 links and Norway with 51 links). The main reason for this choice is that we were not



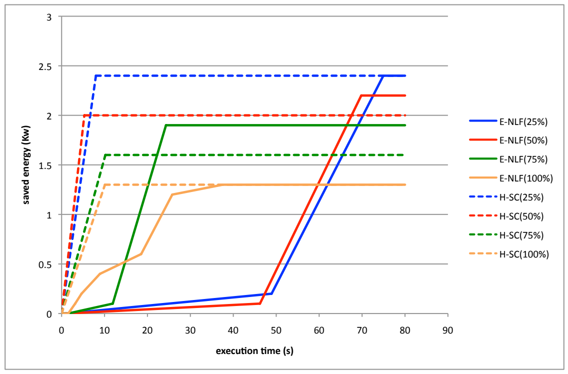
(a) France H-SA vs E-NLF



(b) Norway H-SA vs E-NLF



(c) France H-SC vs E-NLF



(d) Norway H-SC vs E-NLF

Figure 2: The saved power as a function of the execution time

able to find instances of topologies for fixed wireless networks in the litterature. Link capacities and energy consumption were set for a typical fixed wireless network scenario. For the traffic matrix, we consider a uniform traffic matrix. The demand volume for every topology will take four values. Supposing that P is the maximum weight that we can achieve for that topology in an all-to-all scheme, these values would be $\{\ 100\%P, 75\%P, 50\%P, 25\%P\}$.

All the simulations have been executed on the same kind of machine equipped with dual core processors operating at 3GHz and 2GB of RAM. As MIP solver, we used CPLEX 12.1 to which MIP emphasis was set to "feasibility over optimality". In order to solve SDPs, we use the CSDP software. For notation, we use E-NLF for the exact (node-link) formulation. The two heuristics are labeled H-SA for the simulated-annealing-based heuristic and H-SC for the sparse cuts based heuristic.

Since our objective is to show that some heuristics can yield good solutions when the decision making is constrained by a limited execution time, we are not interested in the best solutions found by those heuristics. Hence, in Figure 2 we have reported the behavior of the algorithms before the Exact algorithm finds the optimal solution or near optimal (for the topology of Norway, in some cases, there was a slight improvement of the objective function after a thousand of seconds).

6 Conclusion

We first gave a mathematical formulation to the problem of minimizing energy consumption in fixed broadband wireless networks. Then we compared the behavior of two heuristic algorithms with that of an LP solver. We found out that heuristics would give a better solution if the execution time is very limited, especially when the size of the network increases. Thanks to the sparse cuts based algorithm we were able to verify that these cuts are an important argument to consider while designing heuristics for the studied problem. Therefore, one of our perspectives is to find an efficient way of choosing k cuts out of the sparsest in a given graph.

References

[1] Harry Anderson. *Fixed Broadband Wireless System Design.* John Wiley & Sons, 2003.

[2] Sanjeev Arora, Satish Rao, and Umesh Vazirani. Geometry, flows, and graph-partitioning algorithms. *Commun. ACM*, 51:96–105, October 2008.

[3] Shuchi Chawla, Anupam Gupta, and Harald Racke. Embeddings of negative-type metrics and an improved approximation to generalized sparsest cut. In *16th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 102–111, 2005.

[4] Luca Chiaraviglio, Marco Mellia, and Fabio Neri. Reducing power consumption in backbone networks. In *IEEE International Conference on Communications*, pages 1–6, 2009.

[5] D. Coudert, N. Nepomuceno, and H. Rivano. Power-efficient radio configuration in fixed broadband wireless networks. *Computer Communications*, 33(8):898–906, 5 2010.

[6] F. Giroire, D. Mazauric, J. Moulierac, and B. Onfroy. Minimizing routing energy consumption: from theoretical to practical results. In *International Conference on Green Computing and Communications (GreenCom'10)*, page 8p, Hangzhou, China, 2010.

[7] F. Idzikowski, S. Orlowski, C. Raack, H. Woesner, and A. Wolisz. Saving energy in ip-over-wdm networks by switching off line cards in low-demand scenarios. In *Proceedings of the 14th conference on Optical network design and modeling*, ONDM'10, pages 42–47, 2010.

[8] Rohit Khandekar, Satish Rao, and Umesh Vazirani. Graph partitioning using single commodity flows. In *STOC '06: Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, pages 385–390, New York, NY, USA, 2006. ACM.

[9] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

[10] Tom Leighton and Satish Rao. Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. *J. ACM*, 46(6):787–832, 1999.

[11] David B. Shmoys. Cut problems and their application to divide-and-conquer, 1996.