INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 13(6), 956-971, December 2018.

Multi-Objective Tabu Search to Balance Multihoming Loads in Heterogeneous Wireless Networks

J. A. Huertas, Y. Donoso

Jorge A. Huertas
Center for Optimization and Applied Probability (COPA)
Industrial Engineering Department
Universidad de los Andes, Bogotá, Colombia
huertas.ja@uniandes.edu.co

Yezid Donoso*
Computing and Systems Engineering Department
Universidad de los Andes, Bogotá, Colombia
*Corresponding author: ydonoso@uniandes.edu.co

Abstract: The advantages of the increasing usage of mobile devices that operate under the multihoming scheme are changing the communications world drastically. Therefore, next generation networks operators have the challenging task to distribute connections of mobile devices efficiently over their access networks, creating a big heterogeneous wireless network for telecommunications. We present a mixed integer-linear programming (MILP) model to balance the load of multiple services over wireless networks taking into account three key indicators: connection loads of access networks, connection costs, and battery consumption of connections. To solve the multi-objective problem, we propose a multi-objective Tabu Search procedure that is capable to find non-supported solutions in the online efficient set. To test the performance of our multi-objective Tabu Search, we tested it over four instances of the literature. In the first instance, a small instance, our procedure finds the true efficient set of solutions. For the other three instances, large instances with over a thousand mobile devices, our procedure finds good online efficient sets of solutions in less than 30 seconds. Finally, using appropriate multi-objective metrics, we compare the results of our multi-objective Tabu Search against the results of a *state of the art* multi-objective genetic algorithm in the literature for the same problem, outperforming the genetic algorithm in every instance tested.

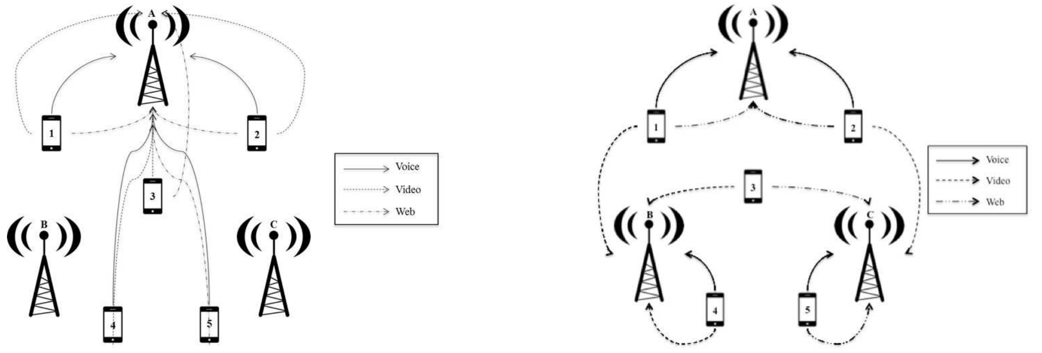
Keywords: Heterogeneous networks, Multihoming, Vertical handover, Optimization, Multi-objective, Tabu Search.

1 Introduction

Multihoming refers to the ability mobile devices have to connect to different access networks at a time through multiple network interfaces [12]. On the other hand, vertical handover (VHO) is the ability to change the access network that provides any service, while maintaining its quality [10]. Multihoming facilitates VHO, making it more seamless to the user [11]; and enhances the flow of data of multiple services across various network interfaces [2], allowing prioritized data to flow through next generation networks (e.g., 4G), and less important data to flow through legacy networks. This behaviour converges towards heterogeneous wireless networks (HWNs).

Thanks to the exponential growth of mobile devices that operate under a multihomming scheme, authorities operating access networks must rely on tools capable to perform VHO processes, while ensuring users' quality of service (QoS). Standards such as IEEE 802.21 MH provide only the framework for performing VHO, yet the decision-making algorithms to do so are an open discussion topic among the research world [15].

The methods in charge to perform VHO in a WHN must recover useful information from mobile devices to decide how to efficiently distribute the connections across different available access points. To illustrate this issue, assume that an operator has three access networks (i.e., A, B, and C), and there are five mobile devices using at least one of three services (i.e., Web, Video, and Voice). A possible bad “distribution" of the connections is presented in Figure 1(a), where all the services are provided by the same access network, which may not have enough bandwidth to do so. Assuming that all services in our example demand the same bandwidth, have the same connection cost, and generate the same power consumption, Figure 1(b) presents a balanced distribution of the connections, which is a better scenario for QoS.



(a) (b)

Figure 1: Connections example: (a) Possible distribution of connections; (b) Balanced solution

In this paper we present an optimization model to achieve a balanced distribution of connections to multiple access networks, taking into account three key indicators: load of access networks, cost of connections, and battery consumption due to connections. To solve the model we propose a multi-objective Tabu Search heuristic. We test our procedure under different small and large-scale instances and compare our results against a genetic algorithm in the literature.

The remainder of this paper is organized as follows: Section 2 discusses the relevant related work in the literature; Section 3 formally introduces the problem description; Section 4 presents the proposed multi-objective Tabu Search; Section 5 presents the computational experiments, as well as its results; finally, Section 6 concludes the paper and outlines future work.

2 Literature review

Multihoming and Load Balancing strategies have been topics of several research projects. For example, [3] presents a model that minimizes the load of networks by re-allocating services from the more loaded network. [1] presents a queuing model that combines multihoming and network coding to improve the allocation of network resources and the communication performance in heterogeneous technologies. [4] proposes an optimization model to perform fairness balance in WHS. Additionally, [9] extended the previous work to a multi-objective model to balance WHNs considering service loads, connection cost, and battery consumption. The two former models only balance the services in the intersection of the access networks' coverage zone. We attempt to extend these models by balancing services in the union of all the coverage zones of the access networks.

Multiple meta-heuristics have proven to be an excellent way to solve combinatorial problems. Simulated Annealing algorithms mimic the cooling processes of metals to find good solutions. Particle Swarm algorithms reproduces the movements of swarms across the globe to solve problems. Genetic algorithms combine the way we evolve to find good solutions across generations. All of these heuristics and meta-heuristics reproduce the nature's processes to find solutions to specific problems. [3] presents multiple heuristics to solve problems related to computer networks. For the load balancing across heterogeneous networks problem [9] uses a multi-objective genetic algorithm to obtain an efficient set of solutions across generations. Unlike them, we specifically focus on Tabu Search to solve the problem.

Introduced by [7], Tabu Search (TS) allows to search the solution space and enables early stopping of algorithms due to local optima. It has been applied to solve a large variety of problems, including set covering problems.

Regarding wireless sensor netowrks (WSNs) problems, numerous work has been accomplished using Tabu Search meta-heuristic. [14] provides a TS-based routing protocol (TSRP) for data routing in WSNs. [6] proposes a centralized clustering method for a data collection mechanism in WSNs that seeks the optimization of the consumption of energy while collecting data. They achieve their mechanism through a Tabu Search meta-heuristic that returns better solutions in terms of cost and execution than other meta-heuristics such as simulated annealing or distributed methods.

Regarding the multi-objective nature of problem we address, [8] provides a framework for the usage of Tabu Search in WSNs problems with multiple objectives. It also addresses multiple WSNs related problems, such as the wireless LAN planning problem (WLP problem), and the evaluation of wireless sensor networks. Using multi-objective Tabu Search meta-heuristics, the authors achieve high-quality solutions regarding the Paretto Optimal frontier, proving that this type of heuristic is suitable to handle problems with numerous combinatorial and continuous variables, such as the ones related to WSNs.

As showed in the previous paragraphs, although many WSN-related problems have been solved using a Tabu Search approach, having extremely good results, many HWN-related problems have been tackled using other types of heuristics. We attempt to fill this gap by balancing multiple service loads in a WHN using a multi-objective Tabu Search procedure.

3 Problem description

We proposed a formulation based on the model presented in [4] and extended to consider multiple objectives in [9]. While both models balance connections from services in the intersection of the access networks' coverage zones, we do so in the union of the available access networks' coverage space.

Let $\mathcal{N}$, $\mathcal{K}$, and $\mathcal{I}$ be the set of the available access networks, mobile devices, and all services in use, respectively. Subset $\mathcal{S}_k$ references the services being used in mobile device $k \in \mathcal{K}$. let c_j and w_j be the cost of connection to the access network $j \in \mathcal{N}$ and the theoretical bandwidth of network $j \in \mathcal{N}$, respectively. Let f_k be the maximum cost the user of mobile device $k \in \mathcal{K}$ is willing to pay to connect to any access network. Let d_i be the demanded bandwidth by service $i \in \mathcal{I}$.

Let m_s be the minimum signal strength required by any mobile device to connect to any network. Let s_{kj} be the signal strength perceived by mobile device $k \in \mathcal{K}$ from network $j \in \mathcal{N}$. To model the power consumption generated in a mobile device when connected to a network we use a power consumption indicator that relates the signal strength perceived by the device. Let o_{kj} be the indicator of power consumption generated in mobile device $k \in \mathcal{K}$ if connected to network $j \in \mathcal{N}$. The power consumption indicator is inversely proportional to the signal strength perceived; it is low (i.e., 1) if the perceived signal is above a signal upper limit u_s; high (i.e., 3) if the perceived signal is below a signal lower limit l_s; and medium (i.e., 2) if it lays between l_s and u_s. Eq. 1 describes how the power consumption is defined.

$$o_{kj} = \begin{cases} 1, & \text{if} \quad u_s < s_{kj}; \\ 2, & \text{if} \quad l_s \leq s_{kj} \leq u_s; \\ 3, & \text{if} \quad l_s > s_{kj}; \end{cases} \qquad \forall\, k \in \mathcal{K}, j \in \mathcal{N}. \tag{1}$$

Let p_k be the percentage of battery remaining in mobile device $k \in \mathcal{K}$. Let b_k be an indicator of the battery percentage remaining in mobile device $k \in \mathcal{K}$; it takes a high value (i.e., 3) if the battery percentage p_k is above a battery-percentage upper limit u_b; it takes a low value (i.e., 1) if p_k is bellow a battery-percentage lower limit l_b; finally, it takes a medium value (i.e., 2), if p_k is between l_b and u_b. Eq. 2 describes how the battery percentage indicator is defined.

$$b_k = \begin{cases} 1, & \text{if} \quad l_b > p_k; \\ 2, & \text{if} \quad l_b \leq p_k \leq u_b; \\ 3, & \text{if} \quad u_b < p_k; \end{cases} \qquad \forall\, k \in \mathcal{K}. \tag{2}$$

This model considers x^i_{kj}, a binary decision variable that takes the value of 1, if the mobile device $k \in \mathcal{K}$ connects to network $j \in \mathcal{N}$ to satisfy service $i \in \mathcal{S}_k$; 0, otherwise. Also, it considers y_{kj}, another binary decision variable that takes the value of 1, if mobile device $k \in \mathcal{K}$ connects to network $j \in \mathcal{N}$ to satisfy any of its services $i \in \mathcal{S}_k$; 0, otherwise. Finally, α, β, and γ are non-negative decision variables representing the maximum load of the networks, the maximum connection cost to the networks, and the maximum consumption indicator from connections to the networks, respectively. Thus the formulation of our MILP is as follows:

$$\text{minimize} \quad \alpha \tag{3}$$

$$\text{minimize} \quad \beta \tag{4}$$

$$\text{minimize} \quad \gamma \tag{5}$$

Subject to,

$$y_{kj} \leq \sum_{i\in\mathcal{S}_k} x^i_{kj}, \qquad \forall\ k \in \mathcal{K}, j \in \mathcal{N}; \tag{6}$$

$$x^i_{kj} \leq y_{kj}, \qquad \forall\ k \in \mathcal{K}, j \in \mathcal{N}, i \in \mathcal{S}_k; \tag{7}$$

$$\sum_{j\in\mathcal{N}} x^i_{kj} = 1, \qquad \forall\ k \in \mathcal{K}, i \in \mathcal{S}_k; \tag{8}$$

$$d_i \cdot x^i_{kj} \leq w_j, \qquad \forall\ k \in \mathcal{K}, j \in \mathcal{N}, i \in \mathcal{S}_k; \tag{9}$$

$$c_j \cdot y_{kj} \leq f_k, \qquad \forall\ k \in \mathcal{K}, j \in \mathcal{N}; \tag{10}$$

$$o_{kj} \cdot y_{kj} \leq b_k, \qquad \forall\ k \in \mathcal{K}, j \in \mathcal{N}; \tag{11}$$

$$m_s \cdot y_{kj} \leq s_{kj}, \qquad \forall\ k \in \mathcal{K}, j \in \mathcal{N}; \tag{12}$$

$$\sum_{k\in\mathcal{K}}\sum_{i\in\mathcal{S}_k} \frac{d_i}{w_j} \cdot x^i_{kj} \leq \alpha, \qquad \forall\ j \in \mathcal{N}; \tag{13}$$

$$\sum_{k\in\mathcal{K}} c_j \cdot y_{kj} \leq \beta, \qquad \forall\ j \in \mathcal{N}; \tag{14}$$

$$\sum_{k\in\mathcal{K}} o_{kj} \cdot y_{kj} \leq \gamma, \qquad \forall\ j \in \mathcal{N}; \tag{15}$$

$$x^i_{kj} \in \{0,1\}, \qquad \forall\ k \in \mathcal{K}, j \in \mathcal{N}, i \in \mathcal{S}_k; \tag{16}$$

$$y_{kj} \in \{0,1\}, \qquad \forall\ k \in \mathcal{K}, j \in \mathcal{N}; \tag{17}$$

$$\alpha, \beta, \gamma \geq 0. \tag{18}$$

Objective function (3) minimizes the maximum load of the networks, objective function (4) minimizes the maximum connection cost to a network, and objective function (5) minimizes the maximum battery consumption from connections to the networks. Constraints (6) and (7) relate decision variables y_{kj} and x^i_{kj} by determining if a mobile device connects to a network to satisfy the demand of one of its services, or not. Nevertheless, to ensure that every service in the mobile devices is served, constraints (8) guarantee the accessibility of the services in every mobile device to a network. Constraints (9) guarantee that the services being used can only be provided by networks with enough bandwidth to meet its demand, not overloading any network. Constraints (10) guarantee that the connection cost to a network does not exceed the maximum connection cost established by the user of the mobile device. Constraints (11) ensure that the consumption indicator generated by the connection to a network does not exceed the battery level of a mobile device. One of the big contributions of this paper is the ability to guarantee accessibility to the services being used in all the mobile devices located in the overall coverage area of the access networks. Constraints (12) guarantee the connection of a mobile device to a reachable network. Constraints (13) capture the maximum load of the available networks. Constraints (14) capture the maximum connection cost to the networks. Constraints (15) capture the maximum power consumption indicator due to connection to networks. Finally, constraints (16)-(18) define the nature of the decision variables.

4 Multi-objective Tabu Search approach

Tabu Search (TS) heuristic explores the solution space of a problem in an iterative way, prohibiting previous solutions to be visited during a number of iterations; and storing some of these previous solutions (which is a good aspect to extend to the multi-objective procedure). We propose a multi-objective TS (MOTS) procedure for solving the model described in Section 3.

The remainder of this section is organized as follows: Sub-section 4.1 introduces the encoding of our MOTS. Sub-section 4.2 shows the move operator. Sub-section 4.3 presents the memory management used. Sub-section 4.4 defines the dominance criteria for our MOTS. Sub-Section 4.5 discusses some of the diversification and intensification strategies used. Sub-section 4.6 presents the pseudo-codes of the MOTS. Finally, Sub-section 4.7 presents appropriate measures to devise the quality of the MOTS.

4.1 Encoding

In the problem defined in Section 3, the decision variable that defines the structure of the problem is x^i_{kj}; it tells how the connections and the multihoming are managed. The other decision variables depend on this binary decision variable. The encoding of the solution is guided by this variable; therefore, we encapsulate the values of the solution in a three-dimensional array $\mathbf{x}$ as the one presented in Fig. 2.

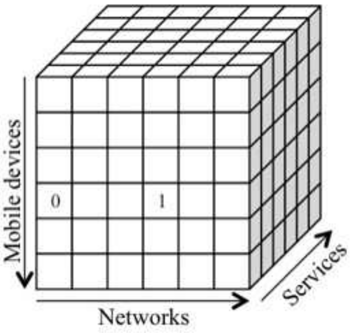


Figure 2: Encoding of the solution

The structure of $\mathbf{x}$ is faithful to the variable notation: the first index (i.e., the rows) references the mobile devices; second index (i.e., columns), the networks; third index (i.e., depth), the services. Each cell of the array can take the values defined by the variable x^i_{kj}, and the value of the remaining decision variables can be obtained by iterating $\mathbf{x}$. For example, in Fig. 2, the fourth mobile device is connecting to the fourth access network to provide the first service (i.e. $x^1_{4,4} = 1$); no other access network can provide this service for this mobile device (i.e., $x^1_{4,j} = 0, \forall\ j \neq 4$).

4.2 Move operator

Constraints (8) guarantee that every service in use in every mobile device gets connected to a network. To guarantee such constraint, we defined a *swap* move. More formally, any possible swap move has the form:

$$swap(x^i_{kj}, x^i_{kj\prime}), \qquad \forall\ k \in \mathcal{K}, i \in \mathcal{S}_k, j, j\prime \in \mathcal{N} \mid x^i_{kj} = 1 \land x^i_{kj\prime} = 0. \tag{19}$$

Move operator (19) defines the neighbourhood of the current solution $\mathbf{x}$. With this definition, the neighbourhood of $\mathbf{x}$ includes both feasible and infeasible reachable solutions. In our MOTS we only move to a feasible solution, due to the convexity of the model defined in Section 3.

4.3 Memory management

Managing short-term and long-term memory in TS is crucial for a good performance of the procedure [7]. For short-term memory we defined a *temporary* list of tabued moves $\mathbf{T}$ given the chronological realizations of the swap move. At any iteration of the algorithm, a new cell in $\mathbf{x}$ takes the value of 1. We prohibit the removal of the new cell (i.e., the cell takes the value of 0) during certain number of iterations. More formally, the tabu move (k, j, i, l) tells a cell x^i_{kj} cannot be zero (i.e., $x^i_{kj} \neq 0$) during l iterations. Constant insertion and deletion of tabu moves to the temporary list allows the exploration of the solution space over solutions that have not been reachd yet; this is a key process in any meta-heuristic [3].

On the other hand, some services won't be used in some devices, because of subset $\mathcal{S}_k, \forall\, k \in \mathcal{K}$. Additionally, a device $k \in \mathcal{K}$ cannot connect to a given network $j \in \mathcal{N}$ if the perceived signal s_{kj} is below m_s, or the connection cost c_j is above f_k. These cases are checked through a *constant* tabu list that allows predefined infeasibility to be taken into account. Some of the impossible connections are preprocessed for the procedure, and taken into account with a dynamic 3D array to preserve the dimensionality of the WSNs related problems.

4.4 Dominance

Let $\alpha(\mathbf{x})$, $\beta(\mathbf{x})$ and $\gamma(\mathbf{x})$ be the associated values of the maximum load of a network, the maximum connection cost to a network, and the maximum consumption of battery level for connecting to a network in solution $\mathbf{x}$, respectively. Let $\mathbf{x}$ and $\mathbf{x\prime}$ be two different feasible solutions to the problem. Solution $\mathbf{x}$ dominates $\mathbf{x\prime}$ if

$$\begin{array}{l}(\alpha(\mathbf{x}) < \alpha(\mathbf{x\prime}) \textit{ and } \beta(\mathbf{x}) \leq \beta(\mathbf{x\prime}) \textit{ and } \gamma(\mathbf{x}) \leq \gamma(\mathbf{x\prime})) \textit{ or} \\ (\alpha(\mathbf{x}) \leq \alpha(\mathbf{x\prime}) \textit{ and } \beta(\mathbf{x}) < \beta(\mathbf{x\prime}) \textit{ and } \gamma(\mathbf{x}) \leq \gamma(\mathbf{x\prime})) \textit{ or} \\ (\alpha(\mathbf{x}) \leq \alpha(\mathbf{x\prime}) \textit{ and } \beta(\mathbf{x}) \leq \beta(\mathbf{x\prime}) \textit{ and } \gamma(\mathbf{x}) < \gamma(\mathbf{x\prime})).\end{array} \tag{20}$$

Additionally, solution $\mathbf{x}$ alternates with $\mathbf{x\prime}$ if

$$\alpha(\mathbf{x}) = \alpha(\mathbf{x\prime}) \textit{ and } \beta(\mathbf{x}) = \beta(\mathbf{x\prime}) \textit{ and } \gamma(\mathbf{x}) = \gamma(\mathbf{x\prime}) \tag{21}$$

A solution $\mathbf{x}$ is efficient when there is no other solution $\mathbf{x\prime}$ that dominates it.

On the other hand, an efficient solution can be either *supported* or *non-supported*. Supported solutions are found through the linear convex combination of the objective functions (i.e., weighted sum of objectives). Non-supported efficient solutions cannot be found through the weighted sum of objective functions. For further information on multiobjective concepts, the interested reader is referred to [3, 5].

We defined a set of current solutions $\bar{\mathbf{S}} = \{\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^v\}$ that changes iteratively depending on the *swap* operator. At the end of any iteration, we obtain the efficient solutions $\mathbf{S}$ in $\bar{\mathbf{S}}$ (i.e., $\mathbf{S} \subseteq \bar{\mathbf{S}}$). After n iterations, the algorithm stops, finding the *online* efficient set $\mathbf{S}$, which is the set containing all the efficient solutions found after n iterations of the procedure.

4.5 Search strategies

Diversification strategies in TS are conceived to explore the solution space that has not been visited until a given iteration. Intensification strategies are made to guide the search through solutions found better historically [7]. To generate both diversification and intensification strategies in our MOTS, at any iteration we move to a random non-dominated solution of the current neighbourhood. With this decision we explore the remaining solution space, because the new solution is non-dominated, while searching good solutions, for the same reason.

4.6 Pseudo-codes

Algorithms 1, 2 and 3 show the pseudo-codes of our MOTS. In Algorthm 1, we define a set of current solutions (i.e., Line 3.) that is iteratively mutated (i.e., Line 6.) to explore the whole search space. Additionally, iteratively we compare the set of current solutions with online efficient set (i.e., Line 7.) to replace it. In Algorithm 2, the random component of the procedure is taken into account in Line 3. The mutation of the current solution is performed in Lines 4. to 8.; also, the addition of the new tabu move is executed in these lines as well (i.e., Line 7.). Finally, during the dominance check in Algorithm 3 we replace the online efficient set in Line 23.

Algorithm 1 Multi-objective Tabu Search

```
Input: $n, v$
Output: $\mathbf{S}$
 1: procedure MOTS($n, v$)
 2:     $\mathbf{S}, \mathbf{T} \leftarrow \emptyset$
 3:     $\bar{\mathbf{S}} \leftarrow$ generateInitialSolutions($v$)
 4:     $\mathbf{S} \leftarrow$ checkDominance($\mathbf{S}, \bar{\mathbf{S}}$)
 5:     for $it < n$ do
 6:         $\bar{\mathbf{S}} \leftarrow$ move($\bar{\mathbf{S}}, \mathbf{T}$)
 7:         $\mathbf{S} \leftarrow$ checkDominance($\mathbf{S}, \bar{\mathbf{S}}$)
 8:         $\mathbf{T} \leftarrow$ iterateTabuList($\mathbf{T}$)
 9:         $it \leftarrow it + 1$
10:     end for
11:     return $\mathbf{S}$
12: end procedure
```

Algorithm 2 Move

```
Input: $\bar{\mathbf{S}}, \mathbf{T}$
Output: $\bar{\mathbf{S}}$
 1: procedure move($\bar{\mathbf{S}}, \mathbf{T}$)
 2:     for $index < |\bar{\mathbf{S}}|$ do
 3:         if $random(0, 1) \geq 0.5$ then
 4:             $\mathbf{x} \leftarrow \bar{\mathbf{S}}(index)$
 5:             $\mathbf{x}' \leftarrow$ non–dominated solution of neighborhood($\mathbf{x}$)
 6:             $(k, j, i) \leftarrow$ swap done to achieve $\mathbf{x}'$
 7:             $\mathbf{T} \leftarrow \mathbf{T} + \{(k, j, i, l)\}$
 8:             $\bar{\mathbf{S}}(index) \leftarrow \mathbf{x}'$
 9:         end if
10:         $index \leftarrow index + 1$
11:     end for
12:     return $\bar{\mathbf{S}}$
13: end procedure
```

Algorithm 3 Check Dominance

```
Input: $\mathbf{S}, \bar{\mathbf{S}}$
Output: $\mathbf{S}$
procedure CHECKDOMINANCE($\mathbf{S}, \bar{\mathbf{S}}$)
    for $index < |\bar{\mathbf{S}}|$ do
        $\mathbf{x} \leftarrow \bar{\mathbf{S}}(index)$
        if $|\mathbf{S}| = 0$ then
            $\mathbf{S} \leftarrow \{\mathbf{x}\}$
        else
            $dominates$, $alternates$, $isDominated$, $enters \leftarrow$ FALSE
            $\mathbf{D} \leftarrow \emptyset$
            for $index_2 < |\mathbf{S}|$ do
                $\mathbf{x}\prime \leftarrow \mathbf{S}(index_2)$
                $dominates \leftarrow \mathbf{x}$ dominates $\mathbf{x}\prime$          ▷ see Eq. 20.
                $isDominated \leftarrow \mathbf{x}\prime$ dominates $\mathbf{x}$          ▷ see Eq. 20.
                $alternates \leftarrow \mathbf{x}$ alternates $\mathbf{x}\prime$          ▷ see Eq. 21.
                $enters \leftarrow$ TRUE
                if $dominates =$ TRUE then
                    $\mathbf{D} \leftarrow \mathbf{D} + \{\mathbf{x}\prime\}$
                end if
            end for
            if $dominates =$ TRUE then
                $\mathbf{S} \leftarrow \mathbf{S} - \mathbf{D}$
            end if
            if $enters =$ TRUE $\wedge$ $isDominated =$ FALSE $\wedge$ $alternates =$ FALSE then
                $\mathbf{S} \leftarrow \mathbf{S} + \{\mathbf{x}\}$
            end if
        end if
        $index \leftarrow index + 1$
    end for
    return $\mathbf{S}$
end procedure
```

4.7 Quality of MOTS

To measure the quality of our MOTS we compute the *Spacing* and the *Spread* of the online efficient set, as [13] did to measure the performance of a multi-Objective Particle Swarm Optimization procedure. These are suitable metrics to measure multi-objective procedures performance. The Spacing of an efficient set measures how distant are the efficient solutions in an efficient set; as this metric takes into account the standard deviation of the distances between the efficient solutions, a low value of spacing tells that the solutions are uniformly spaced, thus, a low variable is desirable. The Spread measures how close the efficient solutions are to the optimal values of the single-objective problems.

Let q be the number of solutions in the efficient set. Let e_i be the distance to the nearest neighbour within the set of solution i. Let $\bar{e}$ be the average distance of e_i. Finally, let e^*_δ be the distance of the optimal solution of objective $\delta \in \{\alpha, \beta, \gamma\}$ to its nearest neighbour within the efficient set. Eq. (22) and (23) present the Spacing and Spread equations, respectively.

$$Spacing(\mathbf{S}) = \sqrt{\frac{1}{q}\sum_{e=1}^{q}(\bar{e} - e_i)^2} \tag{22}$$

$$Spread(\mathbf{S}) = \frac{\sum_{\delta\in\{\alpha,\beta,\gamma\}} e^*_\delta + \sum_{e=1}^{q} |\bar{e} - e_i|}{\sum_{\delta\in\{\alpha,\beta,\gamma\}} e^*_\delta + q\cdot\bar{e}} \tag{23}$$

5 Computational experiments

To test the performance of our MOTS we used four instances instances based on the literature: a small instance and three large instances.

5.1 Small instance

We used the same instance presented in the first case study in [9]. Tables 1 - 4 present the parameters used in the instance.

Table 1: Mobile devices information.

Mobile devices	**Voice**	**Video**	**Web**	f_k **(\$ units)**	p_k **(% charge)**
K1	1	1	0	62	85
K2	1	0	1	44	34
K3	1	1	1	98	74
K4	1	0	1	73	28
K5	1	1	1	82	50

Table 2: Networks information.

Network technology	w_j **(Mbps)**	c_j **(\$ units)**
LTE	70	80
wifi g	54	0
HSPA+	15	40

Table 3: Perceived signal of networks by mobile devices.

Mobile devices	**LTE**	**wifi g**	**HSPA+**
K1	87	78	64
K2	55	68	28
K3	92	47	88
K4	0	85	25
K5	91	95	93

Table 4: Services' demanded bandwidth.

Service	**Voice**	**Video**	**Web**
d_i **(Mbps)**	0.1	3	0.5

We ran our experiments using a Macbook Pro with 8GB RAM with intel i7 processor running at 3.2GHz with turbo boost. We used Gurobi Optimizer on Java to obtain the results of the single objective problem described in Section 3. Table 5 summarizes the results obtained. Column one show the objective optimized, column 2 the id of the solution obtained, columns 3,4, and 5 present the value of the decision variables.

Table 5: MIP results.

Objective	**Solution**	α	β	γ
minimize α	1	0.086	160	6
minimize β	2	0.213	0	9
minimize γ	3	0.207	160	4
minimize γ	4	0.447	80	4

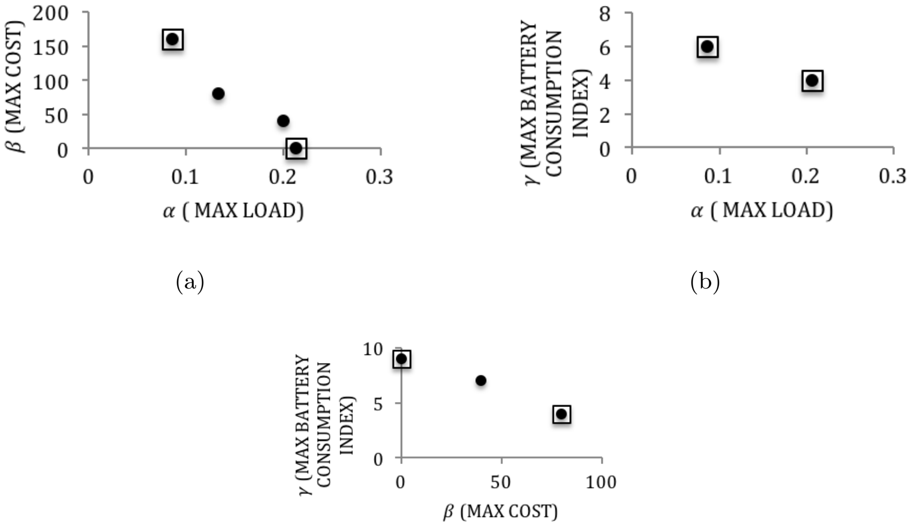
When minimizing the load (i.e., solution 1), the value of α is minimum regarding the ones obtained with the other solutions. As well, when minimizing the cost of connections, the value of β reduces to 0, this is because all services are provided by the wifi g network, which has no cost. Finally, minimizing γ we found two alternated solutions; solutions 3 and 4 have the same value for γ, yet the value of the other variables are different.

To test our MOTS, we generated objective function pairs in order to graph the solutions of the meta-heuristic. After fine-tuning of the parameters, we set $v = 10$, $n = 2,000$, and $l = 1,000$. Fig. 3 presents the efficient set for the three pairs of objective functions (i.e., α vs. β, α vs. γ, and β vs. γ), and the contrasted solution of the single objective problem. Black dots represent the efficient set; white squares represent the points associated to the optimal solutions of the single-objective problem. Note that solutions 1 and 2 (minimizing load and cost, respectively) are the extreme points of the efficient set in Fig. 3 (a); Solutions 1 and 3 are the efficient set of Fig. 3 (b); finally, Solutions 3 and 4 are the extreme points of the efficient set in Fig. 3 (c). Thus, our mots finds optimal efficient set for small instances of the problem.

Our meta-heuristic is able to calculate not only efficient supported solutions, but also the non-supported ones. Fig. 4 illustrates how the efficient set in Fig. 3(a) distributes over supported and non-supported solutions. Black-dotted solutions can be found through the convex linear combination of objective functions α and β; white-dotted solutions are non-supported ones. Traditional optimization of the MILP presented in Section 3 would have never find non-supported efficient solutions. This proves our MOTS is a good method to solve this type of problems.

Finally, we ran our MOTS taking into account all three objectives. As a result, we obtained the efficient set presented in Table 6 in a computational time of 0.96 seconds. Note that all optimal solutions of table 5 are present in the efficient set found.

For the optimal efficient set shown in Table 6, we obtained a Spacing equal to 13.54 and a Spread equal to 1.157. [9] reported a Spacing of 21.505 for this instance.



(a)

(b)

(c)

Figure 3: Efficient set when balancing pair of objectives: (a) α vs. β ;(b) α vs. β ; (c) β vs. γ

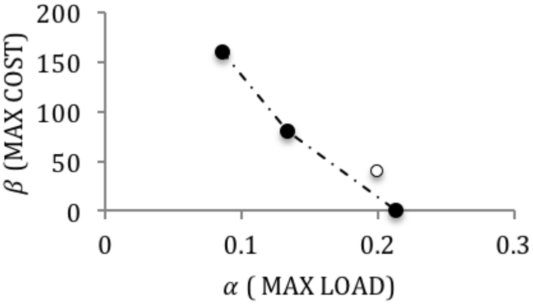


Figure 4: Supported and non-supported solutions for $\alpha - \beta$.

Table 6: Multi-objective efficient set.

Solution	α	β	γ
5	0.207	40	7
6	0.200	40	9
7	0.207	80	5
8	0.447	80	4
9	0.213	0	9
10	0.207	160	4
11	0.086	160	6
12	0.133	80	7

5.2 Large instances

To test the scalability of our MOTS we reproduced the experiments in [9]. We generated three random instances changing the amount of mobile devices: Instance 2 with 20 mobile devices, Instance 3 with 500 devices, and Instance 4 with 1,000 mobile devices. We maintained, at most, five services per device that can connect to seven access networks, having the information presented in [9]. Table 7 and Table 8 present the information of the networks and services used, respectively.

Table 7: Large-scale instances' networks information.

Network technology	w_j **(Mbps)**	c_j **($ units)**
LTE	70	80
wifi n	300	0
wifi g	54	0
wiMAX	15	60
HSPA+	15	40
HSDPA	2	20
UMTS	0.3	10

Table 8: Large-scale instances' networks information.

Service	d_i **(Mbps)**
Voice	0.1
Video	3.0
Web	0.5
Game	2.0
Chat	0.2

We ran our MOTS over the three large instances 30 times. Since the number of possible solutions is higher over the instances, after fine-tuning of the parameters we set $v = 10$, $n = 5,000$, and $l = 2,500$. The results obtained are different from the ones in [9] because we assumed some of the parameters that [9] did not specified in the case studies. Table 9 presents our results. Column 1 references the instance. Columns 2, 3 and 4 reference the average minimum values of α, β, and γ, respectively. Column 5 references the average number of solutions in the efficient set. Column 6 shows the average time running the algorithm. Finally, columns 7 and 8 present the average Spacing and Spread of the efficient set, respectively. Table 10 presents the standard deviation of the values shown in Table 9.

Table 9: Large-scale instances' average results.

Instance	α	β	γ	**Solutions**	**Time (s)**	**Spacing**	**Spread**
Instance 1	0.086	000.000	4.000	8.000	0.960	13.539	1.156
Instance 2	0.427	156.000	10.067	26.867	7.308	8.357	0.958
Instance 3	2.186	548.667	27.333	18.300	14.557	16.590	1.013
Instance 4	4.447	1,134.667	50.400	21.433	20.618	19.499	1.001

As the mobile devices grow, the minimum values of the objective functions are greater too. Nevertheless, the average time to solve the problem is quite reasonable, taking into account the

Table 10: Large-scale instances' standard deviations results.

Instance	α	β	γ	**Solutions**	**Time (s)**	**Spacing**	**Spread**
Instance 1	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Instance 2	0.0500	24.7295	0.2494	156.0224	0.3881	4.9527	0.2154
Instance 3	0.4250	46.7470	0.4618	126.3764	0.3095	6.8807	0.2512
Instance 4	0.5419	76.4933	0.5384	140.1392	0.8100	7.1564	0.1929

average number of solutions in the efficient set. Due to the units of the objective functions, the average Spacing of the solutions obtained are obviously large, yet reasonable. Finally, the Spread of the solutions obtained oscillates around one, which is good. Once we take into account the values presented in Table 10, it is possible to see that the procedure is quite stable regarding CPU time. Also, the standard deviation of the Spreads are really low, ensuring that our meta-heuristic is always near to the real optimal values of the single-objective solutions. Taking into account the units of the instances, the variances of the Spacing are reasonable, telling that the uniformity of the efficient sets found is quite stable.

We are only presenting the minimum values found for each objective, and the decision of using one solution against another one depends on the decision-taker. Nevertheless, considering the metrics measured, the solutions are well distributed along the search space. The values of the standard deviation of β are larger than the other ones, because of the monetary units of this statistic, making reasonable to oscillate around the 50 monetary units.

We compared the results of our average Spacing metric per instance against the Spacing obtained with the genetic algorithm in [9]. Table 11 shows the Spacing obtained in each of the instances by [9] with the genetic algorithm and the average Spacing obtained with our proposed MOTS. As they do not provide any spacing metric for the fourth instance (i.e., 1,000 mobile devices), we only compare Spacings of the second and third instances.

Table 11: Spacing results.

	[9] spacing	**MOTS average spacing**
Instance 1	21.505	13.54
Instance 2	35.542	8.357
Instance 3	30.168	16.590
Instance 4	–	19.499

As the Spacing metric takes into account the standard deviation between solutions in the efficient set, a low value is always better. It is possible to notice that, although optimality was only proved in the first instance, our meta-heuristic produces better online efficient sets than the ones obtained with the genetic algorithm. In all the comparable instances our average Spacing outperforms the ones obtained by [9].

6 Conclusions

We developed a MILP model that takes into account the multihoming scheme; additionally, it only takes into account the destination network during the VHO process. This is a versatile consideration, because in real life, operators know *a priori* the initial networks that the mobile devices are connected to. Our model balance service loads in the union of the coverage areas of the access networks, representing a major contribution to the literature.

We tested our MILP using commercial optimizer Gurobi in a small instance to obtain optimal single-objective solutions. Its result were compared to the efficient set obtained with our MOTS, and matched with the extreme points of the non-dominated frontier. We found more than one non-supported solution, proving that the meta-heuristic was quite successful compared to the convex linear combination of the objective functions; but furthermore, our MOTS reaches the real efficient set for small instances of the problem.

We additionally tested the MOTS over large instances, and achieved good solutions within a reasonable computational time. In addition, we obtained some metrics on the online efficient set proving the efficiency of our meta-heuristic, yet optimality could not be proved. We compared the metrics of the online efficient sets obtained with our MOTS against a multi-objective genetic algorithm in the literature under the same instances. Our results are always better than those we compared to.

With the intuition behind our MOTS we were able to calculate good solutions, yet optimality was not proved. Future work includes merging efficient exact algorithms over networks with our meta-heuristic as a procedure to find good initial solutions; therefore, it could be possible to achieve optimality while solving multi-objective large scale problems.

Author contributions

Jorge Huertas formulated the new MILP model. He also designed and programmed the multi-objective Tabu Search, as well as the computational experiments. Yezid Donoso was the investigation project director. He contributed guiding and reviewing the investigation. All authors contributed equal to the paper.

Conflict of interest

The authors express no conflict of interests.

Bibliography

[1] Capela, N.; Sargento, S. (2012). Optimizing network performance with multihoming and network coding, In *2012 IEEE Globecom Workshops*, IEEE, Anaheim, CA, USA, 2012.

[2] Chu, H.D.; Kim, H.; Seok, S.J. (2013). Flow based 3G/WLAN vertical handover scheme using MIH model, In *The International Conference on Information Networking 2013 (ICOIN) IEEE*, 658–663, 2013.

[3] Donoso, Y.; Fabregat, R. (2007). *Multi-Objective Optimization in Computer Networks Using Metaheuristics*, Auerbach Publications, Boston, MA, USA, 2007.

[4] Donoso, Y.; Lozano-Garzon, C.; Camelo, M.; Vila, P. (2014). A Fairness Load Balancing Algorithm in HWN Using a Multihoming Strategy, *International Journal of Computers Communications & Control*, 9(5), 1841-9844, 2014.

[5] Duque, D.; Lozano, L.; Medaglia, A.L. (2014). An exact method for the biobjective shortest path problem for large-scale road networks, *European Journal of Operational Research*, 242, 788–797, 2014.

[6] El Rhazi, A.; Pierre, S. (2009). A Tabu Search Algorithm for Cluster Building in Wireless Sensor Networks, *IEEE Transactions on Mobile Computing* 8(4), 433–444, 2009.

[7] Glover, F.; Laguna, M. (1997). Tabu Search, Springer, US Boston, MA, 1997.

[8] Jaffrès-Runser, K.; Gorce, J.M.; Comaniciu, C. (2008). A Multiobjective Tabu Framework for the Optimization and Evaluation of Wireless Systems, In *Tabu Search*, I-Tech Education and Publishing, 2008.

[9] Lozano-Garzon, C.; Molina, M.; Donoso, Y. (2016). A Multi-Objective Approach for a Multihoming Load Balancing Scheme in WHN,*International Journal of Computers Communications* & *Control*, 11(2), 1841-9836, 2016.

[10] Marquez-Barja, J.; Calafate, C.T.; Cano, J.C.; Manzoni, P. (2011). An overview of vertical handover techniques: Algorithms, protocols and tools, *Computer Communications* 34, 985–997, 2011.

[11] Patriarca, F.; Salsano, S.; Bonola, M.; Cerqua, P. (2012). UPMT per-application mobility management solution, In *Proceedings of the 10th ACM international symposium on Mobility Management and wireless access - MobiWac '12*, New York, New York, USA (2012); ACM Press, ISBN 9781450316231, 2012.

[12] Peer Azmat Shah; Yousaf, M.; Qayyum, A.; Hasbullah, H.B. (2012). Effectiveness of multihoming and parallel transmission during and after the vertical handover, In *2012 International Conference on Computer & Information Science (ICCIS)*, 625–629 IEEE, 2012.

[13] Salazar-Lechuga, M.; Rowe, J. (2005). Particle Swarm Optimization and Fitness Sharing to solve Multi-Objective Optimization Problems, In *IEEE Congress on Evolutionary Computation*, 2, 1204–1211, 2005.

[14] Semchedine, F.; Bouallouche-Medjkoune, L.; Bennacer, L.; Aber, N.; Aïssani, D. (2012). Routing Protocol Based on Tabu Search for Wireless Sensor Networks, *Wireless Personal Communications* 67(2), 105–112,2012.

[15] Yan, X.; Ahmet, Y.; Şekercioğlu, S.; Narayanan, S. (2010). A survey of vertical handover decision algorithms in Fourth Generation heterogeneous wireless networks, *Computer Networks* 54, 1848–1863, 2010.