Escola d'Enginyeria de Telecomunicació i Aeroespacial de Castelldefels
UNIVERSITAT POLITÈCNICA DE CATALUNYA

MASTER THESIS

TITLE: Reducing the power consumption in Green 5G Networks under system uncertainty

MASTER DEGREE: Master in Science in Telecommunication Engineering & Management

AUTHOR: Sandra Pérez Díaz

ADVISOR: Enrica Valeria Zola

DATE: October 23, 2017

Title : Reducing the power consumption in Green 5G Networks under system uncertainty

Author: Sandra Pérez Díaz

Advisor: Enrica Valeria Zola

Date: October 23, 2017

Abstract

Along this Master thesis, we develop an heuristic model based on a given mixed integer lineal problem (MILP). The problem of energy-efficient user association is approached, as well as the backhaul (BH) routing for 5G Heterogeneous Networs with point-to-point millimeter wave mesh BH links.

The developed heuristic model minimizes the total power consumption of the access network and BH links, subject to some constraints on both, the achievable user rate versus its demand and the maximum link capacity on both the access and BH.

The outcome of the model provides the optimal user association and BH routing strategy. In order to achieve the goal of this Master thesis we use OptaPlanner which is a constraint satisfaction solver that allows us to develop the pursued heuristic using Java. This step consists on creating the UML class diagram in order to identify and implement the respective parameters in OptaPlanner.

Moreover, in this project we also modify the achieved heuristic in order to be able to be robust against user demand deviations. We use the theory of Γ-robustness and derive a robust MILP formulation. We consider different local search algorithms, such as Tabu Search and Lace Acceptance Hill Climbing. In order to decide which one is better we study their effect over our heuristic.

In addition, we contemplate the influence over, not only, the different Γ values, but also different maximum deviation values. We check that the higher Γ value is, the more realistic the scenarios will be, however the power consumption will also increase.

Using several scenarios, we have been tested that the proposed model can achieve a good performance of the heuristic. Furthermore, we quantitatively analyze the trade-off between power consumption versus protection level and robustness.

I would like to thank Enrica Valeria Zola (advisor of my master thesis at EETAC) for giving me the opportunity of developing this project, and guiding and supporting me through the project.

I would also to remember my family and friends for supporting me along, not only, this project, but also the master's degree. Especially, I would like to mention you, for making myself who I am.

Thank you, David, for being Dandra with me, always and forever.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

Acronyms

5G fifth generation

AL access link

BH backhaul

BW bandwidth

eNB eNodeB

FFD first fit decreasing

GBR guaranteed bit rate

HetNet heterogeneous networks

HO heuristic optimization

LAHC late acceptance hill-climbing

LOS line-of-sight

LP linear programming

LS local search

LTE long term evolution

MILP mixed integer linear programming

MIMO multiple-input multiple-output

mmWave millimeter wave

PoR price of robustness

PRB physical resource block

QoS quality of service

RO robust optimization

RR realized robustness

SC small cell

SINR signal-to-interefence-plus-noise ratio

SLA service-level agreement

TS tabu search

UE user equipment

VN virtual node

CHAPTER 1. INTRODUCTION

This Chapter will briefly introduce the pursued project. Besides, it includes a summary of the main objectives as well as the structure of this report. Moreover it shows a timetable where the reader can identify the different tasks that involve the work and their lengths.

1.1. The project

In this master thesis, we tackle the problem of energy-efficient user association as well as the backhaul (BH) routing for fifth generation (5G) heterogeneous networks (HetNet) with point-to-point millimeter wave millimeter wave (mmWave) mesh BH links. We develop a heuristic model based on a given mixed integer linear programming (MILP).

This model minimizes the total power consumption of the access network and BH links, subject to some constraints on both, the achievable user rate and the maximum cell and BH link capacity. The outcome of the model provides the optimal user association (i.e., to which cell of the HetNet the UE connects to) and BH routing strategy (i.e., which path UE will data follow among the many possible paths in the BH mesh).

While the starting point of this project is a given exact MILP problem, this work tries to go a step further developing a heuristic able to be robbust against UEs demand deviations. In other words, the proposed heuristic tries to be conservative to the possible variations in UEs demands. We use the theory of Γ robustness and derive a robust MILP formulation. The achieved outcome is a novel heuristic based on two different search methods: tabu search (TS) and late acceptance hill-climbing (LAHC).

Using several scenarios, we show that the proposed heuristic can achieve near optimal qualities in a short time. Moreover, we also analyze the trade-off between energy consumption versus protection level and robustness.

The main goal of this project is thus to develop a fast and robust heuristic for the mentioned problem of optimization the user association in 5G HetNet regarding the power consumption efficiency in both, the access link (AL) and the BH.

1.2. Objectives

Several objectives have been pursues along this project.

First of all, an optimization problem [3] that tries to solve the user association in 5G scenario was coded in Java. The user association involves both the BH links and the AL, where the BH describes the connection between the macro and the small cell (SC) while the AL describes the one between them and the user equipment UE.

The target of the optimization is to connect all the UEs with a cell trying to minimize the power consumption while guaranteeing a given demanded rate to the UEs. The initial problem in [3] has been slightly modified in order to be able to have several nodes in the BH mesh with a fixed link to the Internet. This has been achieved by adding a virtual node (VN), that way, the macro is able to be switched off.

The second objective was to develop a fast heuristic problem programmed in java that try to find the best association in a given prefixed execution time. This reduction is intended in both cases, the BH and the AL, following the defined scenario in the input data file, where the interconnections between SC and connection between them and the UE are specified.

Moreover, the third objective of this project was to implement a robust environment where some UEs demands are allowed to deviate from the given (e.g., average) value. This assumption is far more realistic, as the user demand cannot be known exactly beforehand, while a trend can be defined through statistical observation of the UE behaviour. However, a deviation in the input data may turn infeasible the given solution to the problem, that is: the association pattern and BH routing may not follow the constraints of the problem anymore, thus decreasing the UE satisfaction on the service received.

This results in an increase in the network power consumption as more resource blocks at the cells need to be left free to cope with the possible increase in the demand; however, the robust allocation will be feasible also in case of a variation in the UE demand, which increases the UE satisfaction in the provided service. That will increase the power consumption of the results, but by contrast it will allow to be more conservative and to avoid more capacity constraint violations.

This new approach will help the heuristic to get closer to realistic cases, where fluctuations on the given parameters may occur.

The fourth objective was to test the proposed heuristics in simple scenarios where the output is easily predictable, thus confirming the proper functioning of the proposed implementation. In Chap. 4 we can find a full explanation of a basic problem which will help the reader to understand the heuristic implementation for more complex scenarios.

Finally, the fifth objective was to observe the impact of several parameters on the results. In Chap. 5 we check the impact of the robustness in the proposed heuristic, as well as the use of different local search algorithms or termination times.

1.3. Structure

This work will be distributed as it will be described in this section in order to help the reader to follow its structure.

- In **Chapter 1**, the project itself and the objectives of this work will be briefly presented to the reader, following with the structure of the complete project, as well as the timetable that collect the information about the required times for the project development.
- In **Chapter 2**, we will find a summary of the main concepts that will be needed for the understanding of the entire work. These concepts talk about 5G mobile networks, which is the communication technology involved along the development of this project. There is also an explanation about the concept of optimization problem and heuristic, which are key concepts for the project development. Moreover the concept of robustness will be described, which allows us to find a join association that faces some deviations on UE demand. This is, it will protect the simulation against changes that some UEs may suffer on their demands, which, at the end, represents a more real scenario. Finally this chapter will include a briefly scheme about the state of art on 5G heterogeneous networks.
- **Chapter 3** will describe the system model as well as the proposed analytic model where all the needed equations will be analyzed. Furthermore the input data files will be explained and finally the complete description of the pursued heuristic. This last part, will include both the described problem and a basic example which will help the reader to understand clearly the function of the developed heuristic as well as the obtained results.
- **Chapter 4** will present the achieved results, where the involved scenarios are presented and the parameters are also.
- To conclude, **Chapter 5** will expound not only some conclusions which are obtained from this work, but also the possible next steps to take into account.
- The reader can also find the used bibliography along this work.
- In **Appendix A** the reader can find more obtained results for different scenarios.

1.4. Methodology

The followed methodology in order to elaborate this Master Thesis can be enumerate as:

- Researching state of the art. This will include topics which have been studied during the Master such as 5G and Optimization, but also new concepts like heuristic or robustness.
- Learn how to properly use the tools such as OptaPlanner (which is programmed in Java). Performing the complete diagram of classes that allows the system to resolve the pursued association.
- Learn about different heuristic and local search algorithms, in order to get into the best solution easily (Late Acceptance and TS).

- Once the concepts, the tools and the algorithms are clear it was required the implementation of the pursued problem, this is the join association between UEs and macro.
- After the heuristic model was implemented, we went a step further trying to learn how to introduce robustness to our proposal.
- Once we know the robustness approach concept, the next step is to add it into our heuristic model in OptaPlanner. Therefore, the given solution will be closer to a more realistic scenario.
- Setting up interesting use cases and finally analyze all the obtained results.

Fig. 1.1 shows the time distribution of the project. The work has been separated into several parts, such as the establishment of the topic, some researching that involves both the theoretical knowledge but also the process of learning how to use the tools (OptaPlanner). Besides, there was a part about layout the heuristic as well as its development. Finally we can find a part for testing and another for writing the report.

As it can be appreciated in Fig. 1.1, the researching block took part during the first period of time of the project, but this task was overlapped with the rest of the tasks due to the need for learning some concepts to implement both, the layout and the development of the complete project. Similarly the testing and the developing part were overlapped thus it was required to test the correct operation of the code while the different test were made.

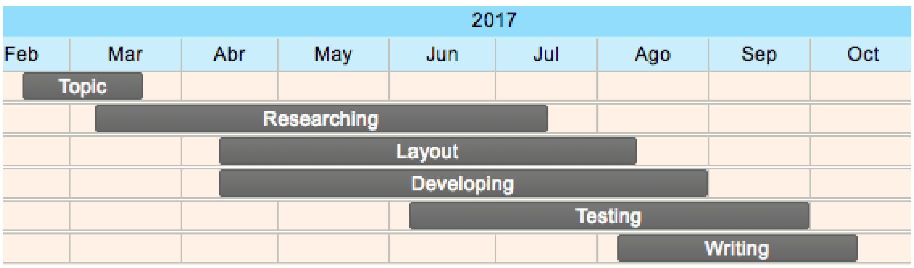

Figure 1.1: Gantt chart

CHAPTER 2. BACKGROUND INFORMATION

Our scenario consists of a 5G network where we can distinguish Long term evolution (LTE) access links and mmWave links for the BH. The goal of this project is to optimize the joint association between macro and UEs. In order to achieve it, the project will include the development of a heuristic model that at the same time is able to consider robustness in the UEs demand.

This chapter will show a brief resume of the concepts that have been involved in this master thesis. Firstly, a summary of the most important aspects about 5G will be described. Secondly the concept of optimization problem will be addressed and finally the robustness concept will be clarified. This chapter will act as summary for the reader with the purpose of a better comprehension of the complete work.

2.1. 5G Networks

Nowadays, more than 5 billion devices demand wireless connections that runs voice, data, and other applications in wireless networks, so its reflected improvements on wireless networks are highly required. Therefore, 5G mobile networks have some extraordinary features and advantages. Some of those will be a better coverage area and an increase on the data rate (around 1 Gbps) at the edge of the cell, as well as a low battery consumption. Furthermore, 5G allows a lower battery consumption thanks to the switching on/off capability. In addition, significant improvements in quality of service (QoS) are expected in the 5G while improving the cost and energy efficiency of the network. Due to the above advantages, 5G wireless system is becoming very much essential, even though it is not yet standardized [4].

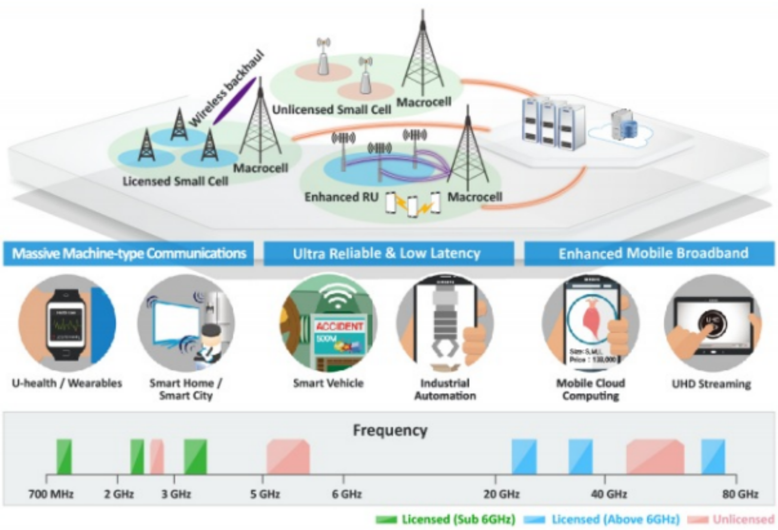

Figure 2.1: 5G scenario [*https://www.ericsson.com/research-blog/5g-challenges-research/*]

To resolve the possible challenges such as the cost and energy efficiency or the energy consumption, it becomes essential to adopt a network infrastructure that can efficiently integrate various wireless technologies and to enable interconnection of existing and new deployed technologies. Towards this end,it is important, without any doubt, to include deployment of super dense HetNet with different types of cells, multiple radio access technologies, massive multiple-input multiple-output (MIMO) technologies at macro and UE as well as the use of both microwave and mmWave frequency bands [4].

HetNets will be deployed as a technique to improve the network capacity, coverage and efficiency by mixing different kind of cells such as macrocells, microcells, picocells, and femto cells. Therefore, SCs are expected to be a key feature of 5G cellular networks as they constitute a viable solution to provide higher end user throughput. The two benefits of adding more SCs are:

i The distance between them and the UEs is reduced and, in consequence, the signal-to-interefence-plus-noise ratio (SINR).

ii The SC bandwidth shared between UEs is smaller hence the number of resources will be larger for each UE (extra capacity improvement).

As a result, future cellular networks are expected to show a denser scenario [5].

The difficulty of connection between SCs and the macrocell is reflected in the need of a new and more economic wireless BH solution. That settlement could be the use of mmWave between SCs and the core network, due to the available bandwidth at these frequencies, which results in high capacity connections [6].

In addition, mmWave frequencies enable only short range point-to-point line-of-sight (LOS) radio links and the connection to the local aggregation gateway would most probably require a number of hops. Therefore, the drawback is that if the transmission distance is shorter than 200 meters the mmWave links may not be established [7–9], thereby a multi-hop architecture will be needed in order to allow every SC to reach the macrocell [8].

Due to the complications of the LOS in the AL (i.e., the connections between UEs and the SCs and/or the macrocell), much researching is still ongoing on how to solve those problems (especially in a mobile UEs scenario). Thus it is more common to assume the use of microwave links through LTE techonology.

2.2. Optimization Problem

An optimization problem is the problem of finding the best solution from all feasible solutions, in other words, maximizing or minimizing some function relative to some set. The function allows comparison of the different choices for determining which might be the “best”.

As a problem that tries to find the best feasible solution, it involves that a large set of solutions (feasible and not feasible) will be contemplated. In order to find those feasible solutions, a set of constraints will be needly defined. However, as it could be intuit, those constraints won't be always fulfilled (e.g unfeasible solutions), but the best solution will be the one that presents the best results.

A general optimization problem will be described as follows:

$$\begin{array}{lll} \text{minimize / maximize} & f(x) & \\ \text{subject to} & G_i(x) = 0, & i = 1, \ldots, m_e. \\ & G_i(x) \leq 0, & i = m_e + 1, \ldots, m. \end{array} \tag{2.1}$$

where x is the vector of length n design parameters, $f(x)$ is the objective function, and the vector function $G(x)$ returns a vector of length m containing the values of the equality and inequality constraints evaluated at x.

Both the size of the problems in terms of number of constraints and design variables and the characteristics of the objective function and constrains are really important due to their dependency with the problem's efficiency and accuracy. When the objective function and the constraints are linear functions of the design variable, the problem is known as a linear programming (LP) problem.

In order to define our problem as an optimization problem we will design it as a MILP, which is a problem where we can find the linear objective function f_x, the bounds and linear constraints and some restrictions on some components of x. Our problem is defined as a MILP due to the fact that some variables are restricted to the set of positive integers. In an LP problem, by contrast, every variable is from the set of Real numbers.

2.3. Heuristic

As is defined in [10], a heuristic aims at studying the methods and rules of discovery in problem solving, which is a process of systematically trying to attain a preconceived but not immediately attainable aim.

To understand the use of the heuristic model along this project, one needs to distinguish between exact methods and heuristic methods. Its principal difference is that the first ones find the optimal solution while the second ones (the ones this work is keen on) try to identify the best solution even though it is not the optimal one. As may be appreciated, the time that an exact method requires to find the optimal solution will be an order of magnitude much higher than the one that is involved for a heuristic method.

The feature that all the heuristic optimization (HO) methods have in common is that they start off with a more or less arbitrary initial solution, then they iteratively produce new solutions using generation rules, they evaluate these new solutions, and eventually they report the best solution found during the search process. This iterated search is usually halted when there is no improvements over a given number of iterations.

In a planning problem there are some categories of solutions:

- A *possible solution* is any solution, regardless if it breaks or not any constraint. Depending on the planning problem the number of possible solutions is huge but many of them are worthless.
- A *feasible solution* is a solution which does not break any hard constraint. Every feasible solution is a possible solution, but sometimes there are no feasible solutions.
- An *optimal solution* is the solution with the highest score. There is always at least one optimal solution even though it is non feasible.
- The *best solution* is the feasible solution with the highest score found by an implementation in some time.

In HO methods, the generation of a new solution can be done by producing a new solution based on the previous results or by modifying the current solution (neighborhood search). The latter will be the one used along the development of this work. Besides, in order to reach the optimal solution, HO methods have a way to treat every solution. That way, each solution will be classified with a ”score”.

Usually HO methods consider not only the solution that reaches an immediate improvement but also some of those that are inferior to the previous solution found so far.

2.3.1. Local Search Algorithms

To solve our optimization problem there are two local search (LS) algorithms that are going to be simultaneously combined. Those are, tabu search (TS) and late acceptance hill-climbing (LAHC).

The first one, TS, is a metaheuristic search algorithm employing local search methods described by [11].

Local search takes a potential solution to a problem and checks its immediate neighbors. That is, it checks those solutions that are similar except for very few details trying to find a better solution. Local search methods have a tendency to become stuck in suboptimal regions where many solutions in the same way fit. Therefore, TS improves the performance of local search by relaxing the basic rule; that way, it can accept some worse move if no improving move is available on the search (i.e., with a strict local minimum), but also, TS avoids with some prohibitions the coming back to previous solutions.

In this way, TS uses a local or neighborhood search procedure to iteratively move from one solution x to an improved solution x' in the neighborhood of x, until some stopping criterion has been satisfied. Those restrict solutions are called the ”Tabu List” and they avoid the system from a cyclic behavior. At the end, in its easiest way, the ”tabu” list is a sort memory that contains the previously visited solutions. That way, TS avoids getting stuck in local optima.

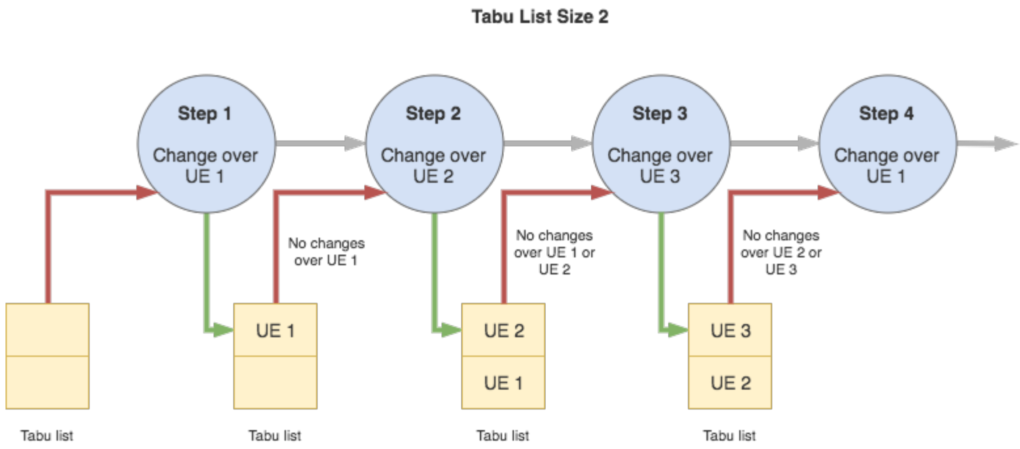

Figure 2.2: Tabu Search graphical representation

In Fig. 2.2 an example of the methodology of the TS algorithm has been represented, with tabu list size equal to two. In *Step 1*, we can see an empty tabu list, so any change is allowed over any UE and we make a change over UE1. At that moment, the UE1 is included in the tabu list. In *Step 2*, movements over users that are in the tabu list are not allowed (in this case, UE1). However, there is a change over UE2, and this UE is now included in the tabu list. Now, in *Step 3*, movements over UE2 and UE1 are not allowed, and the heuristic makes a change over UE3. At this point the tabu list is full so the UE1 (who was included first) is deleted from the list in order to let the UE3 to be included. Therefore, at *Step 4* movements over UE1 are allowed again, but not over UE2 and UE3. This process will continue untill the time is over. The heuristic will propose the obtained solution at the step with the best score.

By contrast, the second algorithm, LAHC [12], implements a similar process to TS: it saves data in a list with information about the previous steps; in contrast, in this case a move is just accepted if it does not decrease the previous score, or if it leads to a score that is at least the last score.

In that way, LAHC accepts any move that has a score higher than the best score of a number of steps ago. Thus, LAHC is more restrictive when the size of the list is small due to that forces the heuristic to find a new best solution between a lower number of steps.

The methodology of LAHC is represented in Fig. 2.3. In *Step 4*, the heuristic proposes a move over the current solution, that is going to be compared with the obtained solution in *Step 2* (as the fact that the list in this example has a length of 2).

As mentioned previously, in this project the HO method considered will be the combination of TS and LAHC, where a potential solution 1) is added to the ”tabu” list preventing the algorithm from visiting again that possible solution, and 2) has to get a higher score of a given fixed number of steps ago.

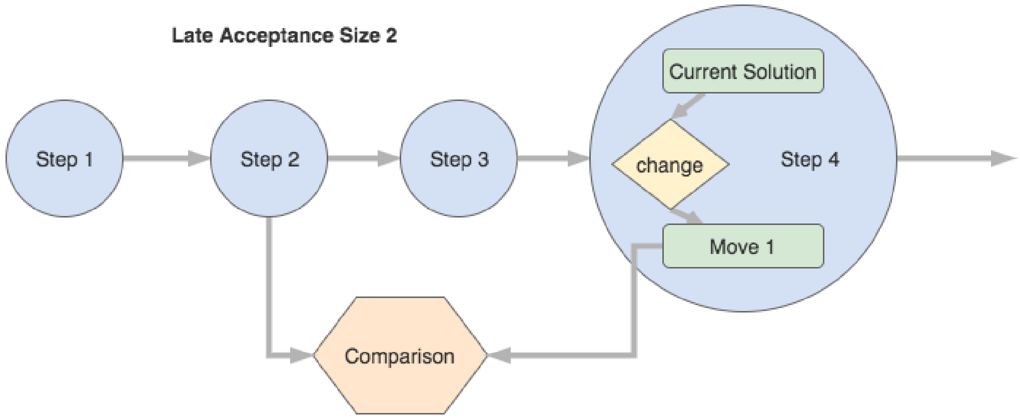

Figure 2.3: Late Acceptance graphical representation

By combining both algorithms it is possible to reach a good solution in a short period of time.

2.4. Robustness

As it is known from Optimization Theory [13], uncertainty in input parameters to an optimization problem may render an optimal solution to a deterministic problem to be an infeasible one if some of the parameters are allowed to deviate from their nominal values. Consequently, most of the optimization models found in literature for user association may lead to infeasible solutions if e.g UE demands are not known precisely. The consequences may be blocked UE due to the non-availability of resources in the macros or SCs to cope with the new demands.

Robust optimization (RO) is known to deal with optimization problems where uncertainty of data is present [13]. RO takes data uncertainty into account already at the modeling stage and tries to protect solutions against uncertainty. In RO the uncertainty model is not stochastic, but rather deterministic and set-based. Instead of solving the problem in some probabilistic sense to stochastic uncertainty, the decision-maker constructs a solution that is optimal for any realization of the uncertainty in a given set.

The general Robust Optimization formulation is:

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x, u_i) \leq 0, \quad \forall\, u_i \in U_i,\, i = 1, \ldots, m. \end{array} \tag{2.2}$$

Where $x \in R_n$ is a vector of decision variables, $u_i \in R_k$ are disturbance vectors or parameter uncertainties, and $U_i \subseteq R_k$ are uncertainty sets. Therefore, this problem offers some measure of feasibility protection for optimization problems containing parameters which are not exactly known.

Γ-Robustness was proposed by Bertismas and Sim [14], and the role of the parameter Γ, which is called protect deviation, is used to adjust the robustness of the proposed method against the level of conservatism of the solution. In that way, the number of coefficients that are allowed to deviate is restricted, leading to the concept of *cardinality constrained robustness*. The idea behind is that it is very rare that all the parameters take the same "negative" deviation at the same time.

As mentioned in the previous section, this work suggests the implementation of a HO method. However, it needs to be adapted in order to search for a solution that not only ensures quality but also increases robustness. Therefore in order to make it robust, the modification of the evaluation function is required in order to investigate those aspects rather than altering any specific feature. Robust HO changes the evaluation function $f(x)$ to $f_r(x)$, following some principles:

Principle 1. Before evaluating the current solution, some noise is added to it. That way, robust HO evaluates $x^* = x + \delta$ instead of x^*. The added noise should be dependent on the expected noise and it should reflect the expected changes in input data.

Principle 2. The updated function is called the robust evaluation function $f_r(x)$. This function does not evaluate a single point, but evaluates several derived solutions which are combined into a single function.

The robust evaluation function may be written like,

$$f_r(x) = \frac{1}{n} \sum_{i=}^{n} w_i f(x + \delta_i) \tag{2.3}$$

where n indicates the number of evaluated functions, and $f(x+\delta_i)$ denotes the function value of each derived solution and it is weighted by w_i.

The allocation strategy which is pursued along this work needs to be robust to UE demands variations which consequently may lead to higher energy consumption. This is because spare capacity needs to be reserved on the cells to face the uncertainty in demand variations. Otherwise, if several UEs increase their demand simultaneously they will be blocked due to lack of available resources. Therefore, the proposed robustness will be defined using Γ as the number of UEs whose demand suffers a certain deviation at each instance. Thus, it is important to remind here that Γ will be at most the number of UEs in the scenario. Besides, the worst case scenario is considered, which represents the case where all the maximum "negative" deviation takes place at the same time for all the UEs.

Consequently, those scenarios with higher Γ values represent a more conservative scenario where the solution is protected agains demand deviations of a higher number of UEs. However, the higher the Γ, the more power consumption, as more UEs may suffer an increase in the demanded rate.

2.5. OptaPlanner

OptaPlanner [1] is a constraint satisfaction solver, it is a lightweight, embeddable planning engine that optimizes planning problems. It enables Java programmers to solve optimization problems efficiently. OptaPlanner will be the tool that this work will use in order to define the pursued planning problem.

Solving a planning problem with OptaPlanner consists out of 5 steps as Fig. 2.4 depicts:

- Firstly, modeling the planning problem as a class that implements the interface Solution.
- Secondly, configuring a Solver (that it will be explained below).
- Third, loading a problem data set which will be the planning problem.
- Then, solving it.
- And finally, getting the best solution found by the Solver.

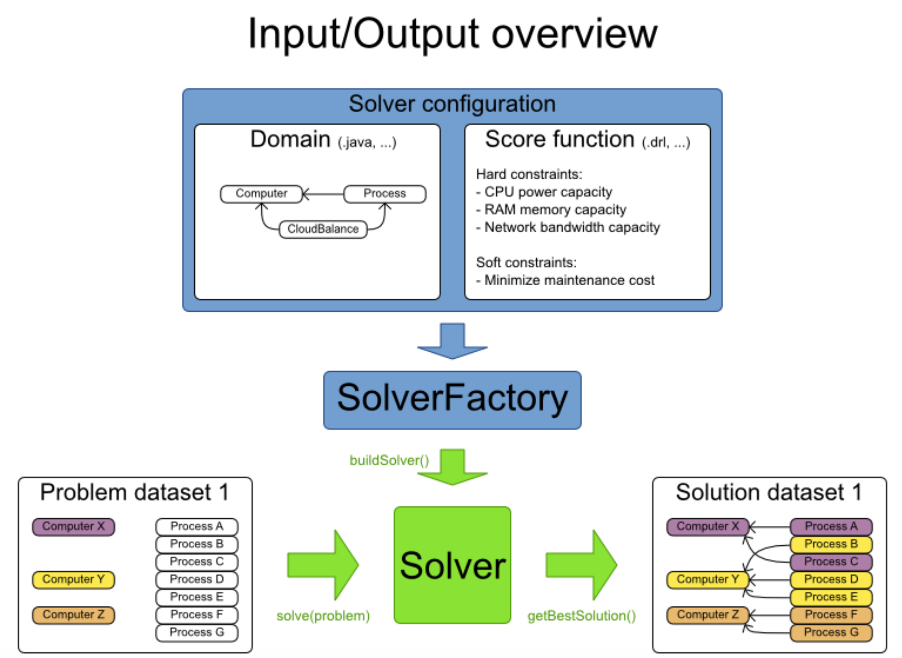

Figure 2.4: Scheme of solving a planning problem with OptaPlanner [1]

2.5.1. Modeling the planning problem

Apart from the solver, the modeling of the planning problem is really important for OptaPlanner. The model will identify several key classes.

- Unrelated class which is not used by any of the score constraints.
- A problem fact class which is used by the score constraints, but does NOT change during planning (as long as the problem doesn't change).
- A planning entity class which is used by the score constraints and DOES change during planning.

It is also important to identify the planning variables. A planning variable is a property on a planning entity that changes during solving. Usually, in a many to one relationship, the "many" side is the planning entity class with the planning variable. In order to understand these concepts, the following chapter will describe the discussed problem in detail.

2.5.2. Identifying the best solution

In order to identify the best solution, the concept of score is required. The score is the objective way to compare two different solutions. At the end, the solution with the highest score will be better. OptaPlanner defines a solver that aims to find the solution with the highest score among all the possible solutions. The solver gives a score to each solution during solving process. That way, the solver identifies the best solution, which is the one with the highest score during the running time, which could also be the optimal solution.

To help the solver to give each solution a score, OptaPlanner needs to know how to calculate it. This is where the function of the constraints emerges.

Usually, a planning problem has at least 2 levels of constraints, hard and soft constraints. A hard constraint is the one that *must* not be broken while the soft one *should* not be broken. Moreover, the constraints could be positive too, then the constraint must/should be fulfilled. These constraints define the score calculation of a planning problem. Each solution is graded with a score. Therefore, in case of negative constraints, the constraints will produce a negative score and vice versa in case of positive constraints.

For example, in order to maximize or minimize a constraint, it is required to tell to OptaPlanner the signum of each constraints. The signum will depend on if the planning problem tries to maximize or minimize that constraint. That way, if a constraint has a positive signum, OptaPlanner tries to maximize it, and vice versa.

Another significant parameter to specify in the solver configuration will be the termination process. Due to a metaheuristic algorithm generally doesn't know when it finds the optimal solution, the only thing you can't depend on, is on finding the optimal solution (unless you know the optimal score). Therefore, especially metaheuristic phases will need to be told when to stop solving.

2.5.3. Solving process

The solving process the solver in the OptaPlanner implements has several parts, which are graphically represented in Fig.2.5.

Figure 2.5: Overview OptaPlanner solving process [1]

There are two implemented phases: *Heuristic Construction*, which defines the starting point of our heuristic, and *Local Search*, which tries to find the best solution for the problem from the starting point. In Sec. 4 this solver implementation is explained in detail using a basic problem in order to increase its understanding.

As it can be appreciated, the solver will run both phases. Each of those will usually iteratively run steps. And each step, usually iteratively runs moves.

New solutions are created by making moves in the current solution. A move is a change which is done in solution *A* to create solution *B*. The difference between steps and moves is that move is every change, independently how good or bad it is, whereas a step is the best found move. That way, the new solution is called a neighbor of the original solution, because it can be reached in a single move. Those movements in our work can be made using different techniques:

- Selecting 1 planning entity and 1 planning value and assigning the entity's variable to that value. At the end, select one UE and assigning a random path.
- Selecting 2 different planning entities and swapping the planning values of their planning variables. This is, exchanging paths of 2 different UEs.
- Changing all the planning entities that have the same planing variable. In this particular case, changing all the UEs with the same path to another path. This change is identical for all of them.

- Swapping all the planning entities that have the same planing variable. At the end, swapping all the UEs with the same path with another set of UEs with the same path.

The Solver will randomly determine which kind of change is going to be selected to supply the next move. All of them have the same probability of being selected.

However, as to evaluate all moves at every step becomes inefficient. Configuring the number of accepted moves is required, that way the solver only evaluates a random subset of all the possible moves.

2.6. State of the art

User association problem is one of the key aspects in dense wireless networks, which decides which UE will be connected to which base station. This problem becomes even more complex in a massive scenario where a huge amount of SC is deployed. Therefore, the UE association impacts indirectly the BH traffic and consequently, the BH utilization and power consumption.

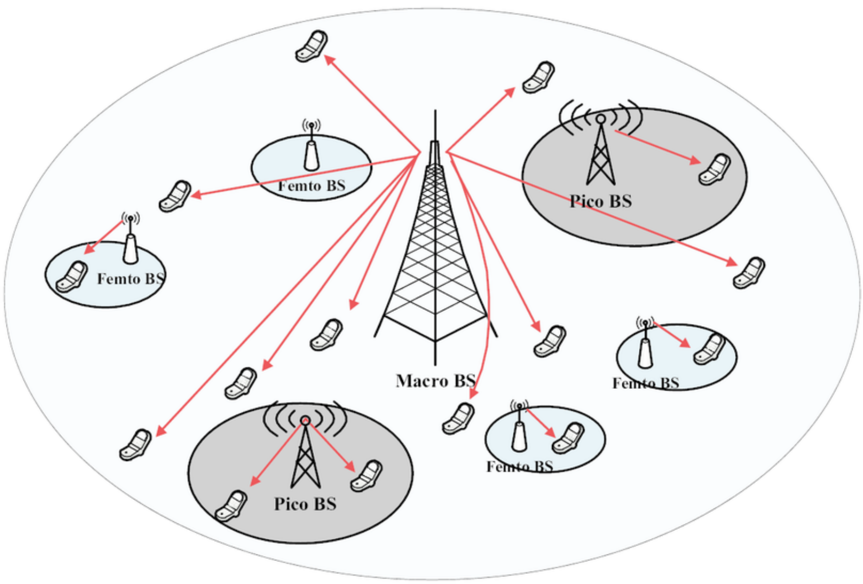

Figure 2.6: Illustration of a heterogeneous network user association. Only one macro is depicted for simplicity [2]

When a new UE joins to the network, both the UE association and the BH routing approaches are required to be optimized in order to reduce as much as possible all the damages that can be derived. These pursued solutions should target a high network energy efficiency, at the same time that achieving low computational complexity.

Most proposed approaches try to optimize the user association without taking into account the BH in some cases, but in other cases they try to solve the routing problem without considering the association options.

Commonly researchers assume that the BH is not the bottleneck neither in terms of capacity nor in terms of energy. However in the case of 5G multi-hop mesh BH may not be completely true. Therefore, researchers have recently studied BH-aware association strategies. Some of those strategies followed are encompassing both the BH and the AL, or in other cases, the joint association problem is studied at the same time of the BH capacity and the energy budget of base stations is considered, as it is proposed in [15] and [16] respectively. However, all those approaches do not consider the total energy efficiency maximization of the network, which is expected to be the aim of the next generation networks.

The closest propose to this work is [17] where an energy-efficient association algorithm is described. However, this work does not optimize the user association and BH routing due to the fact that it only considers a single available path for each connection between SC and macro.

In [3] the authors developed the exact MILP and heuristic for the problem that was the starting point of this project. However, in this work, we go a step further proposing a robust heuristic to the problem.

A similar robust approach was proposed in [18], but that project addresses a different problem on Virtual Machine consolidation.

Similar to [19] where the existence of an optimal setting for the TS list is studied, in this project we have also studied, but not in every detail, the impact on these parameters on our scenarios (see Ch. 5.7.).

CHAPTER 3. MODEL AND HEURISTIC

In this chapter, the system model under study will be described. The user association and BH traffic routing problem was formulated in [3] as a MILP targeting at minimizing the total power consumption of both the BH and the ALs, while trying to satisfy both the BH and AL capacity constraints and the rate demands of UEs. In Sec. 3.1.1. the MILP proposed in [3] is summarized. Sec. 3.2. will introduce the input data file, while the developed heuristic is explained in Sec. 3.3.1.

3.1. System Model

The contemplated topology in this work consists of a set of macros. Besides, a set of SCs is also deployed in the same area. The SCs are connected between them and to the macros through a set of mmWave BH links establishing a mesh BH network. On the other hand, the topology presents a set of UEs trying to access to the network. Each UE requires a specific demand d_u based on its guaranteed bit rate (GBR) service, in other words, each one demands a specific number of physical resource block (PRB). For the access network, a set of microwave ALs will be considered between the UEs and their serving cells (i.e., macros or SCs).

There are a few considerations to take into account about the topology such as, the UE can be associated with just one cell at a time. Furthermore, flat slow fading channels are also considered and constant power allocation is employed, therefore, the maximum transmitted power of each cell is divided equally in its PRBs.

Only downlink transmissions are considered, where the source nodes are located in the Internet (i.e., the cell provide a fixed connection to the Internet), while the sinks are located at the UEs. Hence, a UE may download data from a macro or from a SC. In the latter case, the traffic is routed from the macro to the SC through the BH mesh and then to UE. As clarification, in the path from one macro to one specific UE, one BH link can not be used twice. The flows cannot be splitted over several links, but they have to follow the same set of links.

3.1.1. Analytical Model

As it was already defined, the aim of this project is to minimize the power consumption in both, the AL and BH. This is, the consumption that the macros and the SCs require.

The proposed analytical model is not created along this work, but it is defined in [3]. However, to make it easier to the reader to understand the heuristic model proposed along this project, it has been included hereafter as it constitutes the basis over which our heuristic is built. Therefore, the described MILP may be written as

$$argmin \quad \sum_{i} p_i \tag{3.1}$$

where p_i represents the power consumption of each cell. As it was described before, our MILP searches for the optimal solution not only in the power that is consumed on the AL but also the consumed power on the BH links. That way the total power consumption of the network will be minimized.

To do so, the system requires some constrains related with both the flow conservation and the power.

Before going into detail it is required to indicate that all the needed analytical notation will include in Table 3.1 for a better comprehension of the equations.

Table 3.1: Analytical model notation table

Symbol	Description
$\mathcal{E}$	Set of eNodeBs
$\mathcal{S}$	Set of small cells
$\mathcal{U}$	Set of user equipments
$\mathcal{L}_{\mathcal{AL}}$	Set of access links (ALs)
$\mathcal{L}_{\mathcal{BL}}$	Set of backhaul (BH) links
p_i	Total power of cell i, sum of the power in the ALs of cell i (p_i^{AL}) and in the BH links exiting i (p_i^{BH})
$x^u_{(i,j)}$	Binary indicator of the use of link (i,j) by UE u
$BW_{(i,j)}$	Bandwidth of BH link (i,j)
BW_{PRB}	Bandwidth of a physical resource block (PRB)
$c_{(i,u)}$	Number of PRBs used for the access link (AL) (i,u)
$c_{i_{max}}$	Max. number of PRBs at cell i
d_u	Download rate demand of user u
$SE_{(i,u)}$	Maximum achievable spectrum efficiency for $SINR_{(i,j)}$
$N_{TX_{(i,u)}}$, $N_{RX_{(i,u)}}$	Number of transmit and receiving antennas of AL (i,u)
$\Delta^{AL}_{p_i}$	Slope of the load-dependent AL power of cell i
Δ^{BH}_{p}	Slope of the load-dependent BH power
$N^{AL}_{TRX_i}$	Number of AL transceiver chains at cell i
$N^{BH}_{TRX_{(i,j)}}$	Number of BH transceiver chains for link (i,j)
$p^{AL}_{max_i}$	Max. transmit power of the AL transceiver of cell i
$p^{BH}_{max_{(i,j)}}$	Max. transmit power of the BH transceiver of link (i,j)
$p^{AL}_{out_i}$	RF output power for the AL at cell i
$p^{BH}_{out_{(i,j)}}$	RF output power for the BH link (i,j)
$p^{AL}_{0_i}$	Min. non-zero power consumption of cell i
$p^{BH}_{0_{(i,j)}}$	Min. non-zero power consumption of BH link (i,j)
$\beta_{(i,j)}$	Path loss and gain dependent parameter of BH link (i,j)
$L_{TX_{(i,j)}}$, $L_{RX_{(i,j)}}$	Losses of the transmitter and receiver of BH link (i,j)
$G_{TX_{(i,j)}}$, $G_{RX_{(i,j)}}$	Gain of the transmitter and receiver of BH link (i,j)
$PL_{(i,j)}$	Path loss of BH link (i,j)
LM	BH link margin
N_{TH}	Thermal noise
N_F	Receiver noise figure

Flow Conservation Constraints

The first constraint about the flow conservation describes the prohibition of splitting the flows along the path. It means, the flow can not be divided, and that constraint is shown as,

$$\sum_j x^u_{(i,j)} - \sum_j x^u_{(j,i)} = \begin{cases} 1, & \text{if } i = \text{source} \\ -1, & \text{if } i = u \text{ (sink)} \\ 0, & \text{otherwise} \end{cases} \tag{3.2}$$
$$\forall\, u \in \mathcal{U}, \quad \forall\, i \text{ and } j \in \mathcal{E} \cup \mathcal{S} \cup \mathcal{U}$$

where $x^u_{(i,j)}$ is a binary link vector that is 1 when the link (i,j) is used by an UE and 0 otherwise.

Besides, there is another restriction in terms of UE connections, because a UE only can to connect with one cell at a time, therefore.

$$\sum_{(i,u)\in\mathcal{L}_{\mathcal{AL}}} x^u_{(i,u)} = 1, \quad \forall\, u \in \mathcal{U} \tag{3.3}$$

As mentioned before, UEs in the same cell share all its available resources. Therefore, the following equation shows the number of PRB assigned to each UE according the cell availability and the demand of each one. However, even the number of this parameter is described, it is defined directly in the input file.

$$c_{(i,u)} = \left\lceil \frac{d_u}{BW_{PRB} SE_{(i,u)}} \right\rceil \tag{3.4}$$

where the BW_{PRB} identifies the bandwidth of a PRB while the $\lceil\ \rceil$ denotes the ceiling operator. On the other hand, it is required to identify the parameter $SE_{(i,u)}$ which is defined by [20] as

$$SE_{(i,u)} = min(N_{TX_{(i,u)}}, N_{RX_{(i,u)}}) \log_2\left(1 + \frac{N_{RX_{(i,u)}} SINR(i,u)}{min(N_{TX_{(i,u)}}, N_{RX_{(i,u)}})}\right) \tag{3.5}$$

However, even though both parameters, $c_{(i,u)}$ and $SE_{(i,u)}$ are defined, the input file already shows the needed value for the creation and resolution of the heuristic. As it will be presented in the following chapters.

On the other hand, on the AL the SC's PRB capacity is not allowed to be lower than the number of the assigned PRB to each UE. This restriction is defined as follow,

$$\sum_u x^u_{(i,u)} c_{(i,u)} \le c_{i_max}, \quad \forall\, i \in \mathcal{E} \cup \mathcal{S} \tag{3.6}$$

where c_{i_max} identifies the maximum capacity of the cell i. In this work, this value is defined in the input file as the maximum number of PRB of each one.

Power Constraints

As the previous chapter mentions, this work propose the aim of minimizing the power consumption either in BH and AL. Therefore, to calculate the power consumption of each cell, two terms will be considered. First the power consumption in its ALs (p_i^{AL}), and the second one is the power consumed by all the BH links existing in it (p_i^{BH}).

$$p_i = p_i^{BH} + p_i^{AL}, \quad \forall\, i \in \mathcal{E} \cup \mathcal{S} \tag{3.7}$$

To understand where p_i^{BH} and p_i^{AL} come from, the following lines will explain how to calculate them.

Regarding **the power model in the access link** is needed to remark that the linear approximation [21] is considered.

$$p_i^{AL} = N_{TRX_i}^{AL}(p_{0_i}^{AL} + \Delta_{p_i}^{AL} p_{out_i}^{AL}), \quad \forall\, i \in \mathcal{E} \cup \mathcal{S} \tag{3.8}$$

In the p_i^{AL} the power is calculate using the number of transceiver chains of cell ($N_{TRX_i}^{AL}$), the minimum non-zero output power of the AL transceiver at cell i ($p_{0_i}^{AL}$) which is static, the slope of the load-dependent power consumption, which takes different values based on the used type of the antenna, is also considered ($\Delta_{p_i}^{AL}$), and finally the power consumption of the transceiver for the ALs between each cell and its associated UEs ($p_{out_i}^{AL}$). This last parameter is given by

$$p_{out_i}^{AL} = \frac{p_{max_i}^{AL}}{c_{i_max}} \sum_{u \in U} (x_{(u,i)} c_{(u,i)}) \tag{3.9}$$

where $p_{max_i}^{AL}$ is the maximum transmit power of all the ALs of each cell.

On the other hand, **the power model in the backhaul** is quite similar to the adopted approach on the AL. As it happened with p_i^{AL} before, now the p_i^{BH} consists on a static part $(p_{0_i}^{BH})$ and a variable one that scales with the aggregate rate flowing over each BH link. Therefore, the power of all the BH links existing at each cell is calculated as

$$p_i^{BH} = \sum_{(i,j)} N_{TRX_{(i,j)}}^{BH} (p_{0_{(i,j)}}^{BH} + \Delta_{p_{(i,j)}}^{BH} p_{out_{(i,j)}}^{BH}) \tag{3.10}$$

where again $N_{TRX_{(i,j)}}^{BH}$ is the number of transceivers chains of BH link (i,j), the minimum non-zero output power of the AL transceiver at cell i is represented by $p^{BH}_{0_{(i,j)}}$, the parameter $\Delta^{BH}_{p_{(i,j)}}$ denotes the slope of the load-dependent BH power consumption and finally $p^{BH}_{out_{(i,j)}}$ represents the RF transceiver output power of BH link (i, j) at each cell [22].

A new restriction according with the maximum power transmission is the one related with the $p^{BH}_{max_{(i,j)}}$, therefore,

$$0 \leq p^{BH}_{out_{(i,j)}} \leq p^{BH}_{max_{(i,j)}} \tag{3.11}$$

Besides, $p^{BH}_{out_{(i,j)}}$ is given by

$$p^{BH}_{out_{(i,j)}} = \left(2^{\frac{\sum_u x^u_{(i,j)} d_u}{BW_{(i,j)}}} - 1 \right) \beta_(i, j) \tag{3.12}$$

where $\sum_u x^u_{(i,j)} d_u$ can be identified as the aggregated traffic that passes through each link, the $BW_{(i,j)}$ is the bandwidth of the BH link (i, j), and finally the parameter $\beta_(i, j)$ results by subtracting from the total losses the gains of the transmitter and the receiver of the BH link (i, j).

This other parameter will be shown in the input file, so it will be already calculated.

It is needed to clarify that as the equation (3.12) is non-linear, an approximation will be needed. In order to do so, a piecewise linear interpolation function will be used. Depending on the scenario, 6 or 7 segments will be required to linearize and approximate the power consumed by the transceiver of a BH link. This function will be explained in detail in Sec. 3.2. where the input data file is described.

3.2. Input Data

The input data file, as it was advanced before, contains some parameters that are strictly necessary for the correct system operation. These input data files have not been created along this work, but they were provided. They have been generated from simulations for a specific scenario (i.e., number of SCs, UEs, possible connections between UEs and macros or SCs, BH links, UE demands..)

Firstly, the file shows a section where the number of VNs, macros, SCs, UEs are denoted, as well as the number of antennas used.

After that, a section with the information of the set of BH edges is described. This information specifies for each link, the id of source and sink, its maximum capacity in Mbps, the value of $\beta_{(i,j)}$ required in Eq. 3.12 and its bandwidth in Hertz. Once the info of the BH is detailed, there is a new section which shows the information of the set of AL edges. In this case, the given information for each link consists on again the id of source and sink, the number of PRB which are associated to each edge (the value of Eq. 3.4, the power PRB in watt required in Eq. 3.9 (specifically it shows the value of $\frac{p^{AL}_{max_i}}{c_{imax}}$), and finally the value required in the denominator for the calculation of the parameter $c_{(i,u)}$ in Eq. 3.4 (particular case of $BW_{PRB}SE_{(i,u)}$).

The two following sections show the set of UE and their demands. Finally, and in order to approximate the first part of the Eq.3.12 (remembering the value of $\beta_{(i,j)}$ is given) using a piecewise linear interpolation function, the values needed for this function are also detailed. These are specified by giving a set of slopes and a set of breakpoint at which the slopes change.

In order to increase the understanding, a graphical resource is included in Fig. 3.1 where the reader can see an example of the piecewise-linear function.

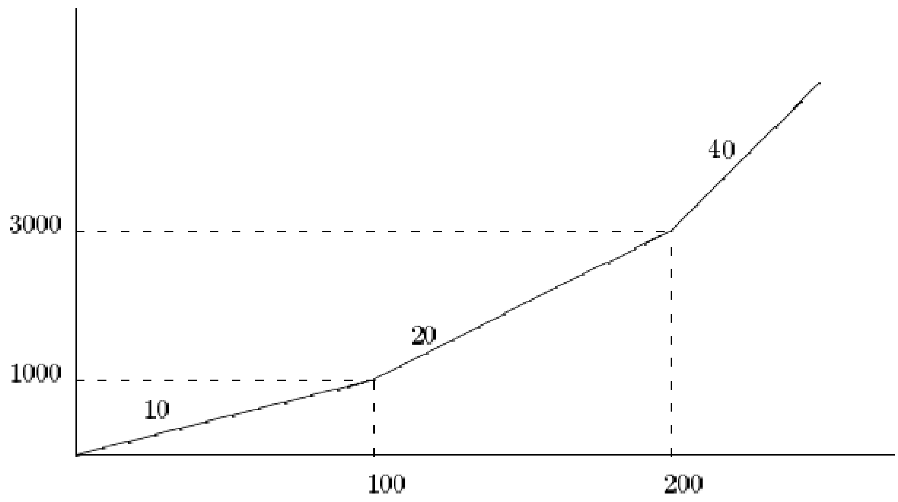

Figure 3.1: Piecewise-linear function

In this function we have defined the value of the slopes (10, 20, 40) and the value of the breakpoints (100, 200), that way to calculate the value of a point in the y-axis we have to use the final point in x-axis and the slope for each section.

3.3. Proposed Heuristic

The aim of this work is, again, to solve the joint association between UEs and the correspondent macro, which in this work could be either a macro cell, an SC or a virtual node. To do that and according to the concept of heuristic explained in Sec. 2, this work aims at developing a heuristic model, which is based on the proposed analytical model described in Sec. 3.1.1.. This heuristic tries to solve the same problem as before, which was provided as a basis for the creation of this heuristic model.

The achieved process is represented in Fig. 3.2. This diagram shows that the association (access) and routing (BH) problems are solved at the same time in the first step. Then, the heuristic gives a score to that solution and swaps in order to make changes trying to obtain a new solution. Once either the time is over or some time has elapsed without improvements, the heuristic stops. At that time, it selects the best found solution, and after that it decides which cells can or cannot be switched off (those cells will be the ones that remain unused once the user association is completed).

We want to recall here that the aim of the heuristic is to find the best approaches regarding the power consumption, that is, the best solution will be the one that associates all the UEs with their macro, and at the same time reduces the power consumption in both the access and the BH links.

The proposed heuristic along this work is developed using Java as a programming language, and the used open source software to solve it is OptaPlanner [1].

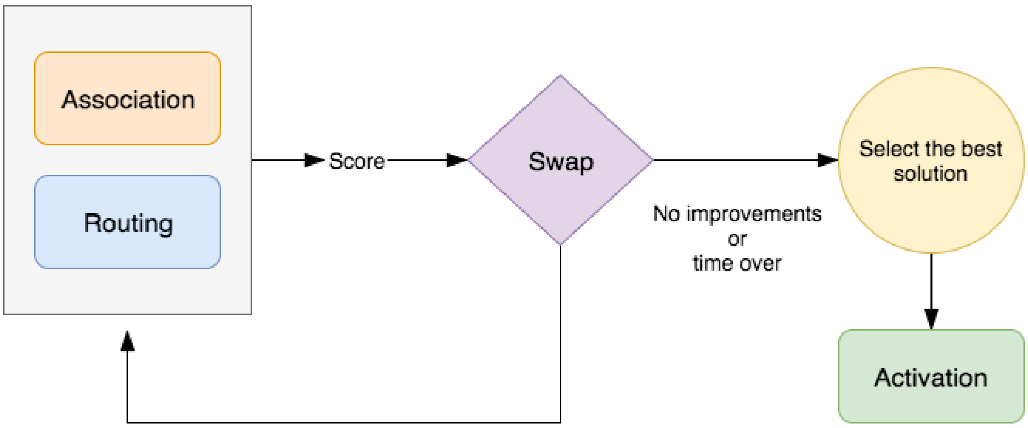

Figure 3.2: Diagram of the heuristic model execution

3.3.1. Described problem

As it was introduced in the previous chapters, the suggested scenario along this project consists on a HetNet with one or more VNs (representing the anchor to the Internet), one or more macro cells and several SCs. Additionally, all of them are interconnected among them as it is described on the data input file (Section 3.2.). Moreover, even though UEs have the possibility to connect with some of those base stations (i.e., macro cell or SC), they must connect just with one.

As there is just one UE-base station choice, all the information belonging to the same UE will follow the same path, understanding the path as the set of hops the information, of a given UE, will pass through. Thus, before solving the UE association problem, all the possible combinations of hops have to be created at first in order to build the set of possible paths.

These paths will have as an end each UE and the other end will be always the virtual node. These embrace since the path with just the direct connection between each UE and the virtual node, and the path between each UE and each macro/SC through different SCs. Besides, a path can not include twice or more times the same base station. To gain an understanding, Fig. 3.3 is included, where path 1 and 2 are applicable, but path 3 is not as it pass twice by the same SC (SC1).

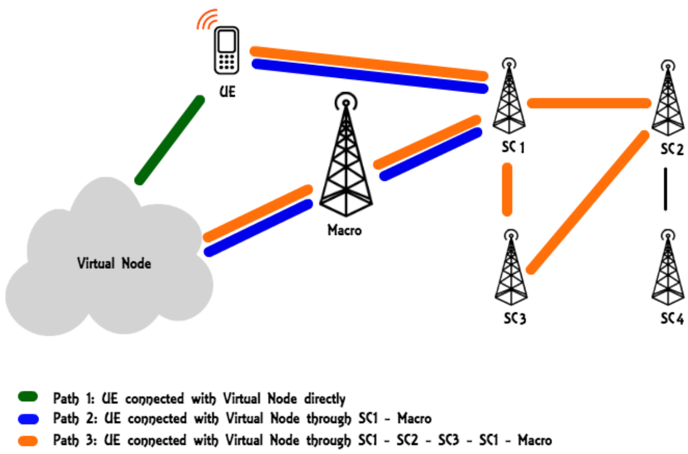

Figure 3.3: Example of two possible paths for the same UE

Once the path concept for the correct development of this work is clarified, it is needed to identify into our problem the planning entities, planning variables, etc. described in the previous section, which are the key concept to the solver configuration.

In the defined class model there are three classes that are indispensable in the solving. Those are the class that is related with the UEs and their parameters, the one that refers to the set of links, the one with the parameters of each path, and finally the associated one with the solver.

User class:

This class is the one which is linked with the UE requirements. Firstly, this class will have a parameter to identify the requested demand of each UE in terms of Mbps. Moreover, the UE class will include related information with all the possible connections a user may use.

Thus, this class will incorporate a list with the needed PRBs for the connection with each cell. That way, this list will identify with a number of PRBs if there is any available connection with the cell, and with a larger number (greater than the maximum amount of PRB by cell) if the UE has no connection with one cell.

Similarly, the UE class will show a list of required power which is again different for the connection with each cell. In the same way, a high number will identify no connection with a cell while a number lower than 1 will tag the available connections. Besides, the information about the denominator of Eq. 3.4 is also included as a list in order to use it in the constraints calculation. This information will be depicted in the input data file (Sec. 3.2.) depending on the AL characteristics.

This class will also include a parameter to identify the followed path, that will connect the UE with the virtual node using the denoted SC.

Link class:

This class will identify all characteristics of each link that will be involved in the path. That way, this class will present the capacity (this time in Mbps, as we are talking about the BH) related with each edge. Once again, this class will present the power consumption of each link.

Furthermore, this class requires also information about the required parameters for the calculation of the power consumption such as β value, bandwidth (BW), maximum capacity... of each link.

Path class:

The path class will be the one that collect all the information about the set of links that the BH follows between each cell and the virtual node. That way, it comprises a list that identifies the involved cells on each hop of the path, at the end this list identifies the existing set of BH links in each path. This is, this class will have as parameters one list of the link objects created with the previous class.

Solver class:

This is the last class, and it concerns to the identification of the score. This class will be the one that is responsible to give an score to each found solution. To do that, this class has to identify the previous classes as lists to play with. That way, the solver will know which parameters has to move to find new solutions and which parameters has to include into the score calculation.

According to the explanation made in Sec. 2.5. *User class* will be identified as the planning entity since it will change during planning and it will be used by the constraints. It means, this will be the class that we want to change its parameters (planning variable), in this case the parameter that will change will be the *Path* variable.

Thusly, remembering that the planning variable will identify that/those planning entity properties that change during solving, the planning variable will be the *Path* in this case. That way, the solver will try to find the best path that each UE has to follow to reduce the AL and BH power consumption.

Fig. 3.4 depicts the class diagram of the described problem. Where we can identify each class with its parameters as well as the tag for each planning variable and planning entity which are required in the solver configuration as it was introduced before.

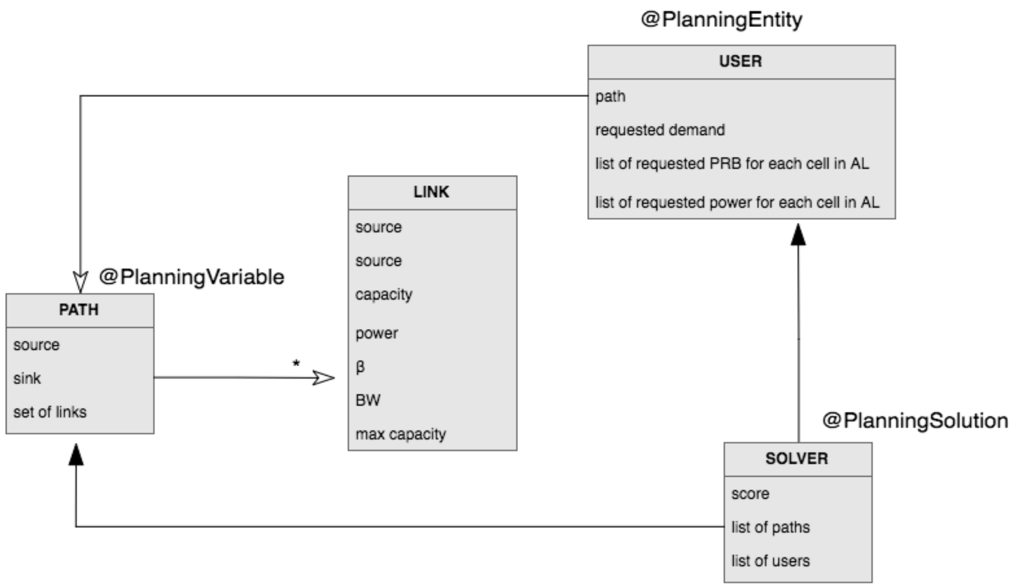

Figure 3.4: Model class diagram

CHAPTER 4. BASIC EXAMPLE PROBLEM

In order to clarify the understanding of the solving process, this section will try to represent all the middle steps that the proposed heuristic does.

A simple scenario is depicted in Fig. 4.1 representing the initial system model with one VN, one macro, six SCs, and three UEs. The BH connections between cells are represented as a gray straight line, whereas the AL are shown with a dotted green line.

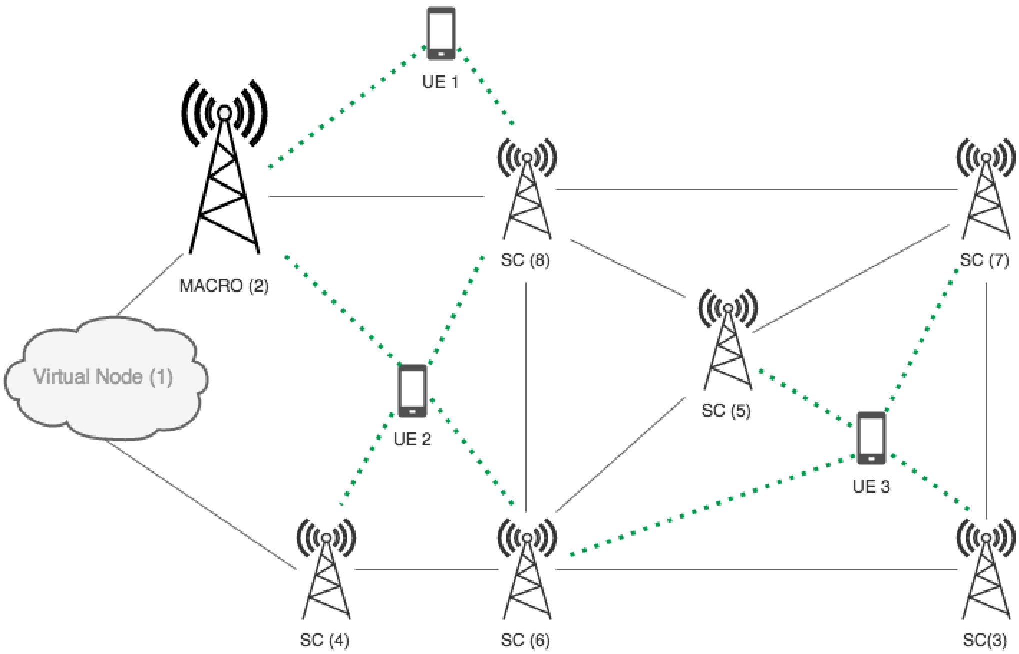

Figure 4.1: Basic example problem scenario

Once the scenario is defined, previously to the solving process it is needed to identify all the different paths in the BH network. That way, two different association will appear, the one related with the access network and the one related with the path the user's information will follow in the BH.

This way, the heuristic model tries to find the best connection between UE and cells following two steps:

- Select the best AL to connect with in order to reduce the power consumption, at the same time fulfilling the capacity requirements of the link.
- Select the best BH path with one end as the selected cell in the access network. Similarly, it could be different ways to connect the virtual node with the UE so the heuristic tries to find the best option reducing the power consumption while fulfills all the link's requirements.

As it can be appreciated, there are several combination of both, and not always the best connection for the access network will result on the best path for the BH. That way, it is required to find a balance between them, which will result on the best found solution.

The generation of the set of possible paths is required. In this example, by using the restriction of maximum 4 hops, an amount of 45 different paths are available.

In order to understand the whole progress of the heuristic, we are going to represent several steps to find the best solution. However, Fig. 4.2 will represent all the steps that OptaPlanner does in order to solve this basic problem. This picture shows that the heuristic creation phase consists of three steps, as well as the LS phase does.

```
DEBUG     CH step (0), time spent (47), score (-2init/0hard/-127soft), selected move count (45), picked move (RobustUE-1 {null -> Path-3}).
DEBUG     CH step (1), time spent (56), score (-1init/0hard/-212soft), selected move count (45), picked move (RobustUE-2 {null -> Path-2}).
DEBUG     CH step (2), time spent (70), score (0hard/-297soft), selected move count (45), picked move (RobustUE-3 {null -> Path-4}).
INFO  Construction Heuristic phase (0) ended: time spent (72), best score (0hard/-297soft), score calculation speed (3317/sec), step total (3).
DEBUG     LS step (0), time spent (350), score (0hard/-253soft), new best score (0hard/-253soft), accepted/selected move count (100/3637),
          picked move (RobustUE-2 {Path-2 -> Path-4}).
DEBUG     LS step (1), time spent (674), score (0hard/-222soft), new best score (0hard/-222soft), accepted/selected move count (100/6450),
          picked move ([RobustUE-1] {Path-3 -> Path-10}).
DEBUG     LS step (2), time spent (30674), score (0hard/-253soft),     best score (0hard/-222soft), accepted/selected move count (0/2930043),
          picked move (RobustUE-1 {Path-10 -> Path-20}).
INFO  Local Search phase (1) ended: time spent (30729), best score (0hard/-222soft), score calculation speed (95907/sec), step total (3).
INFO  Solving ended: time spent (30730), best score (0hard/-222soft), score calculation speed (95680/sec), phase total (2), environment mode
```

Figure 4.2: Involved steps in solving the basic problem

4.1. Heuristic Creation Phase

This phase is called heuristic creation and is the starting point of the solving process. As mentioned before, this phase will have several steps and several moves. During this phase, there are just 3 steps, one for each UE, and as many moves as the number of paths (as in this case there are less paths than moves accepted at each step).

Even though there are several ways to do it, this heuristic creation phase uses the *first fit decreasing (FFD)* algorithm. It cycles through all the planning entities but it assigns the more difficult planning entities first. That is, this algorithm will try to associate each UE sorting it using some sorting method.

The developed sorting method consists on assign at the beginning those UEs who have less chances to associate in the access (i.e., those with less options to connect with cells). Specifically, in this basic example, the one with less opportunities is UE1 (it has just 2 chances, with SC8 and Macro2), while the other two UEs have more options.

Fig. 4.3 shows the last moves of the first step of the heuristic creation. Those are the last movements of the association of the most difficult UE. This is, it represents the tests of association between UE1 and some paths, specifically from path 27 to path 44. At the end of this step, we can see the given score to this step, *score (-2init/0hard/-127soft)*. Where, the first part *-2init* means there are other 2 UEs without association (as this is the first step of this phase), *0hard* means the hard constraints are not broken, and finally the *-127soft* represents the power consumption at this moment where just one UE is associated.

```
TRACE       Move index (27), score (-2init/-900hard/-32210soft), move (RobustUE-1 {null -> Path-27}).
TRACE       Move index (28), score (-2init/0hard/-221soft), move (RobustUE-1 {null -> Path-28}).
TRACE       Move index (29), score (-2init/-900hard/-32210soft), move (RobustUE-1 {null -> Path-29}).
TRACE       Move index (30), score (-2init/-900hard/-32210soft), move (RobustUE-1 {null -> Path-30}).
TRACE       Move index (31), score (-2init/0hard/-221soft), move (RobustUE-1 {null -> Path-31}).
TRACE       Move index (32), score (-2init/-900hard/-32210soft), move (RobustUE-1 {null -> Path-32}).
TRACE       Move index (33), score (-2init/-900hard/-32210soft), move (RobustUE-1 {null -> Path-33}).
TRACE       Move index (34), score (-2init/0hard/-221soft), move (RobustUE-1 {null -> Path-34}).
TRACE       Move index (35), score (-2init/-900hard/-32210soft), move (RobustUE-1 {null -> Path-35}).
TRACE       Move index (36), score (-2init/0hard/-221soft), move (RobustUE-1 {null -> Path-36}).
TRACE       Move index (37), score (-2init/-900hard/-32210soft), move (RobustUE-1 {null -> Path-37}).
TRACE       Move index (38), score (-2init/0hard/-221soft), move (RobustUE-1 {null -> Path-38}).
TRACE       Move index (39), score (-2init/0hard/-1196soft), move (RobustUE-1 {null -> Path-39}).
TRACE       Move index (40), score (-2init/-900hard/-32210soft), move (RobustUE-1 {null -> Path-40}).
TRACE       Move index (41), score (-2init/-900hard/-32210soft), move (RobustUE-1 {null -> Path-41}).
TRACE       Move index (42), score (-2init/-900hard/-32210soft), move (RobustUE-1 {null -> Path-42}).
TRACE       Move index (43), score (-2init/-900hard/-32210soft), move (RobustUE-1 {null -> Path-43}).
TRACE       Move index (44), score (-2init/-900hard/-32210soft), move (RobustUE-1 {null -> Path-44}).
DEBUG     CH step (0), time spent (69), score (-2init/0hard/-127soft), selected move count (45), picked move (RobustUE-1 {null -> Path-3}).
```

Figure 4.3: Example of one step and some of its different moves in the heuristic creation phase

This process will be repeated until the last step in the heuristic phase ends, this is, until all the UEs are associated (see Fig. 4.4). At that moment, the given score is *(0hard/-297soft)* which means that the hard constrains, which are straightly related with the flows of each link, are not broken (otherwise this value will be different to 0), while the soft ones, the related ones with the power consumption of each link, have a value of *-297*.

```
INFO  Construction Heuristic phase (0) ended: time spent (132), best score (0hard/-297soft), score calculation speed (1259/sec), step total (3).
```

Figure 4.4: Chosen solution at the end of the heuristic creation phase

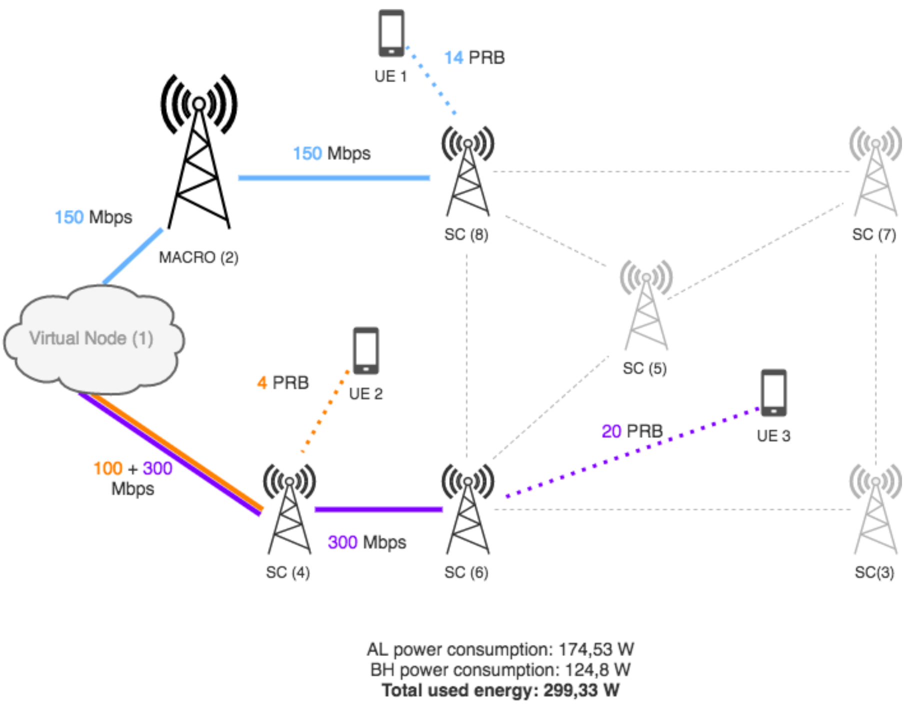

Figure 4.5: Graphical solution at the end of the heuristic creation phase

For a better understanding, Fig. 4.5 shows a graphical representation of this first association between cells and UEs. This picture also depicts the different consumed resources of each link (PRB for the access and Mbps for the BH). Besides, the figure shows the power consumption for each network and also the global one. On the other hand, the picture represents the unused cells and links using soft gray color, such SC3, SC5 and SC7.

It is important to highlight that the score given by OptaPlanner (*-297soft*) should correspond with the exact value in absolute value of the power consumption in the association (this time *299.33 W*). However, and as a way to simplify the model, the calculation of the constraints does not consider decimals therefore the OptaPlanner gives a score very similar, but not identical, to the final power consumption value (this can be applied to the rest of commented solutions).

4.2. Local Search Phase

Hereafter, local search phase begins. The process consists on generating a new solution from the previous one. This is, during the different steps of this local search, the aim is to select the solution with the lowest values of hard and soft constraints. However, those values have to be considered using just their absolute value, thus the minus sign just identifies the minimizing end.

Once local search phase is started, it is needed to remember how this new solution (and consequently the following ones) is created. As it was mentioned earlier, each new solution comes from a change or movement in the previous one. The OptaPlanner in the local search phase tries every move on the current solution and picks the best accepted move as the step according to some parameters. Those parameters are the number of accepted movements by each step, as well as the termination conditions. For every test, the parameters chosen are 100 accepted movements and the heuristic will be terminated after 120 secs or if after 30 secs there are no further improvements.

```
TRACE        Move index (4412) not doable, ignoring move ([RobustUE-2] {Path-2} <-> [RobustUE-2] {Path-2}).
TRACE        Move index (4413), score (-900hard/-32300soft), accepted (false), move (RobustUE-3 {Path-4} <-> RobustUE-2 {Path-2}).
TRACE        Move index (4414) not doable, ignoring move (RobustUE-3 {Path-4} <-> RobustUE-3 {Path-4}).
TRACE        Move index (4415), score (-900hard/-32300soft), accepted (false), move (RobustUE-3 {Path-4} <-> RobustUE-2 {Path-2}).
TRACE        Move index (4416), score (-900hard/-32300soft), accepted (false), move ([RobustUE-2] {Path-2} <-> [RobustUE-3] {Path-4}).
TRACE        Move index (4417), score (-900hard/-32300soft), accepted (false), move ([RobustUE-2] {Path-2} <-> [RobustUE-3] {Path-4}).
TRACE        Move index (4418) not doable, ignoring move (RobustUE-3 {Path-4 -> Path-4}).
TRACE        Move index (4419) not doable, ignoring move (RobustUE-3 {Path-4} <-> RobustUE-3 {Path-4}).
TRACE        Move index (4420), score (-914hard/-33585soft), accepted (false), move (RobustUE-3 {Path-4 -> Path-38}).
TRACE        Move index (4421), score (-1800hard/-64286soft), accepted (false), move (RobustUE-1 {Path-3} <-> RobustUE-3 {Path-4}).
TRACE        Move index (4422) not doable, ignoring move ([RobustUE-2] {Path-2 -> Path-2}).
TRACE        Move index (4423), score (0hard/-1241soft), accepted (false), move ([RobustUE-1] {Path-3 -> Path-1}).
TRACE        Move index (4424), score (-1800hard/-64286soft), accepted (false), move (RobustUE-1 {Path-3} <-> RobustUE-3 {Path-4}).
TRACE        Move index (4425), score (-900hard/-32349soft), accepted (false), move (RobustUE-1 {Path-3 -> Path-12}).
TRACE        Move index (4426), score (-900hard/-32574soft), accepted (false), move (RobustUE-1 {Path-3} <-> RobustUE-2 {Path-2}).
TRACE        Move index (4427), score (0hard/-253soft), accepted (true), move (RobustUE-2 {Path-2 -> Path-4}).
TRACE        Move index (4428), score (-900hard/-32349soft), accepted (false), move (RobustUE-1 {Path-3 -> Path-16}).
TRACE        Move index (4429), score (-900hard/-32300soft), accepted (false), move (RobustUE-2 {Path-2} <-> RobustUE-3 {Path-4}).
TRACE        Move index (4430), score (0hard/-297soft), accepted (true), move (RobustUE-3 {Path-4 -> Path-6}).
DEBUG     LS step (0), time spent (1944), score (0hard/-253soft), new best score (0hard/-253soft), accepted/selected move count (100/3637), picked move
             (RobustUE-2 {Path-2 -> Path-4}).
```

Figure 4.6: Example of the step 0 and some of its different moves in the local search phase

As Fig. 4.6 depicts LS has the same execution way than the previous phase. However, there is a slight difference, now the movements consists on making changes on the current paths. Now, the moves are built selecting randomly among the 4 algorithms which were explained in Sec. 2.5. Those changes are for example, changing the path of the UE2 from path 2 to path 4.

In Fig. 4.6 it can also be appreciated the existence of some moves with the *"not doable, ignoring move"* (for example, the first move shown which proposes to change UE2 from path 2 to path 2). This message means that the move keeps the same path, so at the end it is not making any change from the previous solution.

That way, the first step in the LS phase obtains the represented solution in Fig.4.7 where we can see the same existing connections for UE1 and UE3 while the UE2 shows a different path for the BH and the AL comparing with the obtained solution in the heuristic creation phase (see Fig. 4.5).

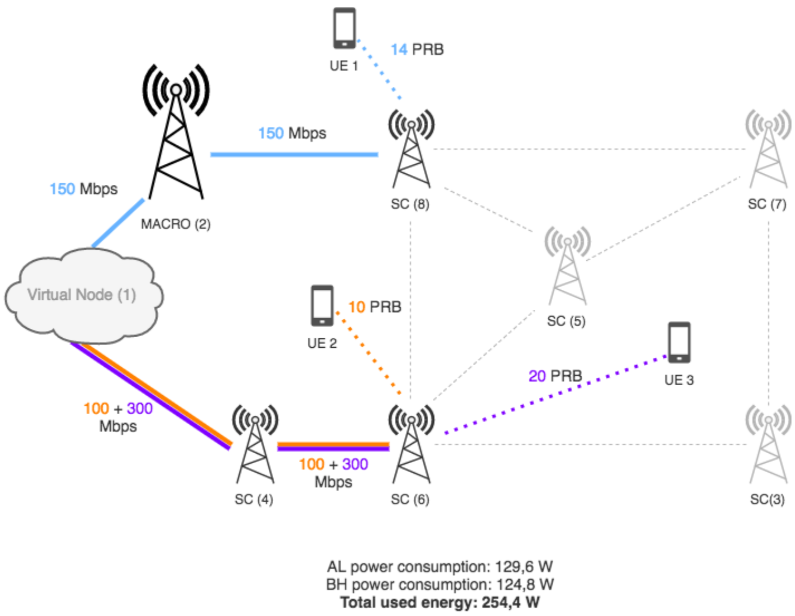

Figure 4.7: Checked solution at the first step done in local search phase

This solution has an score equal to *(0hard/-253soft)* where again this solution doesn't break any hard constraint, and shows a lower value (in absolute value) than the previous solution, thereby this solution is better than the previous one. As it can be appreciated, the power consumed in the BH remains identical whereas in the access a decreasing of power occurs as the new selected path for the UE2.

As it was shown, this phase, in this basic example, consists on 3 different steps. That way, once the first step ends, the second step starts, and Fig. 4.8 depicts some of those movements.

It is important to remember the used local search algorithms are TS and LAHC. Therefore, and as Fig. 4.8 shows, a new difference appears between step 0 and the following steps in LS phase. This difference is the message *"Proposed move is tabu and is therefore not accepted"* (for instance, this figure shows that all the movements that involves the UE2 are not allowed due to that UE was moved in the previous step). This message introduces the concept of tabu list which was previously explained in Sec. 2.3.1.

In this example, the tabu list only contains 1 element, therefore, in step 1 the UE that have to remain without changes is the one associated in step 0. In step 2, the UE that is not

```
TRACE      Proposed move ([RobustUE-2] {Path-39 -> Path-39}) is tabu and is therefore not accepted.
TRACE      Move index (8734), score (0hard/-1348soft), accepted (false), move ([RobustUE-2] {Path-4 -> Path-39}).
TRACE      Proposed move ([RobustUE-2] {Path-3} <-> [RobustUE-1] {Path-4}) is tabu and is therefore not accepted.
TRACE      Move index (8735), score (-920hard/-33160soft), accepted (false), move ([RobustUE-2] {Path-4} <-> [RobustUE-1] {Path-3}).
TRACE      Move index (8736), score (-914hard/-33588soft), accepted (false), move (RobustUE-3 {Path-4 -> Path-34}).
TRACE      Move index (8737) not doable, ignoring move (RobustUE-3 {Path-4} <-> RobustUE-2 {Path-4}).
TRACE      Proposed move ([RobustUE-2, RobustUE-3] {Path-14 -> Path-14}) is tabu and is therefore not accepted.
TRACE      Move index (8738), score (-904hard/-32403soft), accepted (false), move ([RobustUE-2, RobustUE-3] {Path-4 -> Path-14}).
TRACE      Move index (8739), score (-900hard/-32242soft), accepted (false), move (RobustUE-1 {Path-3 -> Path-23}).
TRACE      Move index (8740), score (-900hard/-32274soft), accepted (false), move (RobustUE-1 {Path-3 -> Path-14}).
TRACE      Proposed move ([RobustUE-2, RobustUE-3] {Path-41 -> Path-41}) is tabu and is therefore not accepted.
TRACE      Move index (8741), score (-906hard/-32561soft), accepted (false), move ([RobustUE-2, RobustUE-3] {Path-4 -> Path-41}).
TRACE      Move index (8742), score (-900hard/-32242soft), accepted (false), move ([RobustUE-1] {Path-3 -> Path-23}).
TRACE      Move index (8743), score (0hard/-347soft), accepted (false), move (RobustUE-3 {Path-4 -> Path-33}).
TRACE      Move index (8744) not doable, ignoring move ([RobustUE-1] {Path-3} <-> [RobustUE-1] {Path-3}).
TRACE      Move index (8745), score (0hard/-253soft), accepted (true), move (RobustUE-1 {Path-3 -> Path-20}).
DEBUG    LS step (1), time spent (5989), score (0hard/-222soft), new best score (0hard/-222soft), accepted/selected move count (100/6450), picked move
           ([RobustUE-1] {Path-3 -> Path-10}).
```

Figure 4.8: Step two and some of its different moves in the local search phase

able to change its path will be the one associated in step 1, and so on. In this case, the UE who can not be change is *UE2*.

On the other hand, the presence of the LAHC algorithm is reflected when the move is not accepted (*accepted (false)* in both, Fig. 4.6 and 4.8). This non acceptance is due to the fact of these moves are worst than the previous. Another evidence of the use of this algorithm is the number of the selected move (*accepted/selected move count (100/6450)* in Fig. 4.8), where it can be appreciated the selected move is the move number 100 of the 6450 proposed moves (taking into account that from all of them just 100 are accepted). This time, the last accepted movement is the selected one for this step.

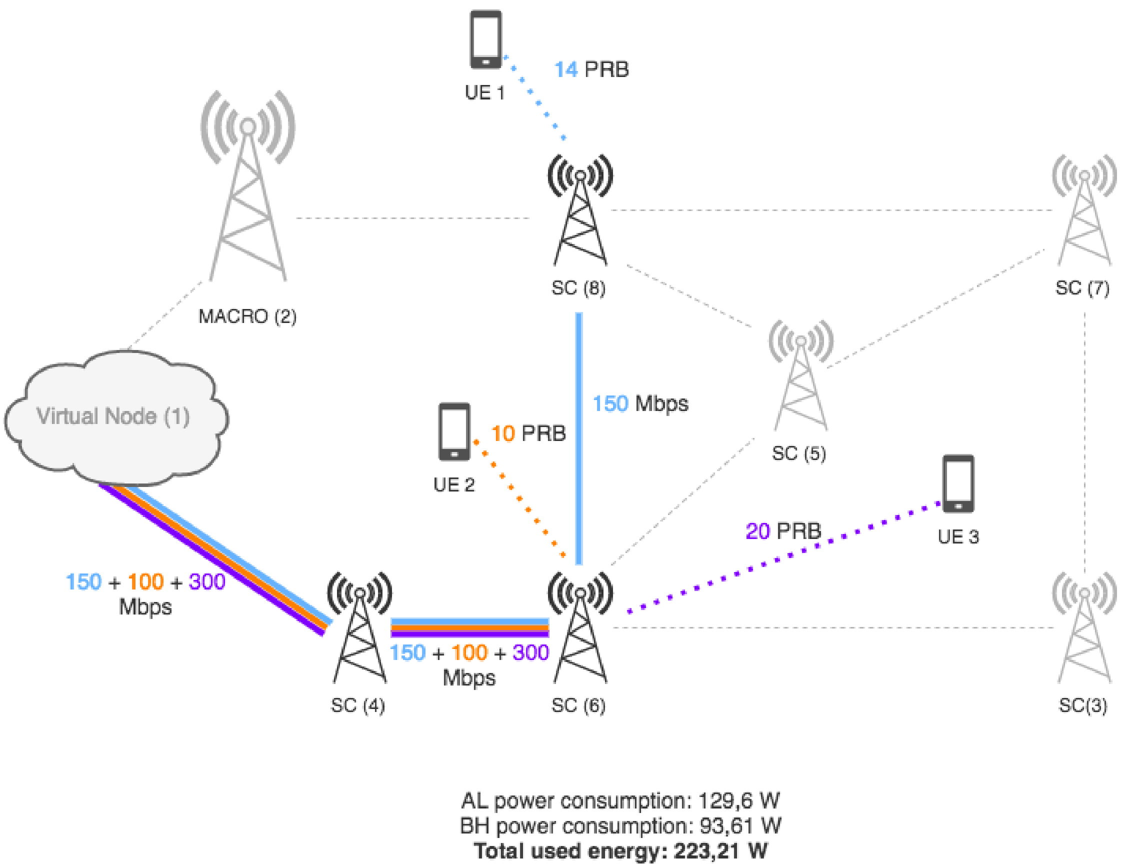

Figure 4.9: Graphical solution at the second step done in local search phase

The graphical solution for this second step is depicted on Fig. 4.9, where we can appreciate the new solution shows the same existing connections for UE2 and UE3 while the UE1 shows a different path for the BH but the same AL comparing with the previous step in LS, this is, Fig. 4.7.

This step 2 will end and the last step will start following the same commented process. Finally, the proposed solution for the third step of the LS phase is represented in Fig. 4.10.

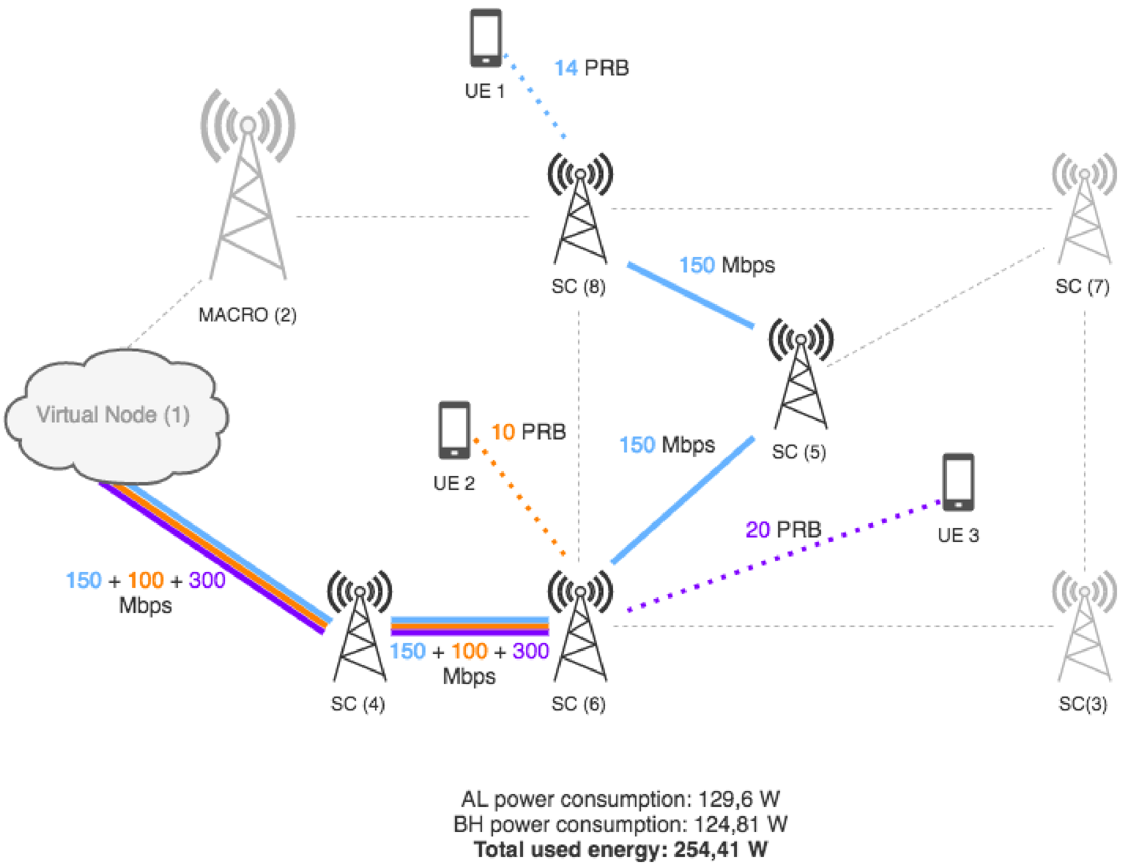

Figure 4.10: Graphical solution at the third step done in local search phase

We can check that the obtained total power consumption is identical to the given solution for the step 0 in this phase (LS). However, both solutions (step 0 and step 2) are not good solutions, as they propose a solution with more power consumption than the one in step 1. Therefore, the best solution for this basic problem will be the solution obtained in step 1, which is represented in Fig. 4.9.

To sum up, this chapter has explained with lots of details the followed process by OptaPlanner in order to find the best solution. This problem is very simple in order to increase the understanding of the process, and to act as a proof of the proper operation of the developed heuristic. That way, the reader can understand how the heuristic works even for bigger scenarios.

CHAPTER 5. RESULTS

This chapter will gather information about the tested scenarios as well as some tests and results that have been performed along this work. Those results may help the reader to check the accuracy of the heuristic model against the exact one.

5.1. Scenarios and Parameter Selection

There are several tested scenarios along this project, which help with the verification of the correct operation of the proposed heuristic. Those scenarios consist on different scenarios with 1 macro, 16 SCs and different number of UEs (20, 40, 60, 80, 100). Those scenarios have been provided from external simulations using several parameters that can be found in the Appendix A.

In order to avoid repetition, and due to the fact that all the scenarios behave similarly, the results will be demonstrated using a single scenario, the case of 100 UEs. It is important to remark that not all the scenarios behave identically, therefore even though the values are in absolute value it does not mean the rest of the scenarios behave identically.

Besides, other scenarios with different distribution of cells, such as with one VN, have been tested. Those proofs and extra results are given in Appendix A. This allows the reader to understand with a global view the results obtained, independently of which scenario is used in each moment since the impact on the power saving will be different for each scenario.

According to the selected parameters, similar to the basic proposed example (Sec. 4) there are two termination conditions. Firstly, there are a initial global limitation of 300 seconds, however there is also another limitation since the heuristic will finish if after 30 seconds there are no improvements. Besides, for every experiment the tabu list and late acceptance list may have different values since each scenario requires different inputs. For the exposed results, tabu list has a length of three users, while the late acceptance list has a size of seven solutions.

Observations: Determining the length of those list is a very complex process due to the size affects to the performance of the algorithms, as they shown in [19] for the Traveling Salesman Problem. Therefore, it may occur that the chosen length is not the optimal since the process for their determination was no exhaustive. This interesting aspect needs further study and may be proposed as an extension to the current work.

5.2. Impact of using different optimization algorithms

In order to define which algorithms will be used in the configuration of the heuristic, some tests have been performed. Regarding the explained algorithms in Sec. 2.3.1., it was required to select between them. In order to make the decision before testing anything else, we tried to solve the heuristic problem using, separately and jointly, both algorithms. The result of this test is, as it was advanced in preceding sections, the combination of both heuristic search algorithms.

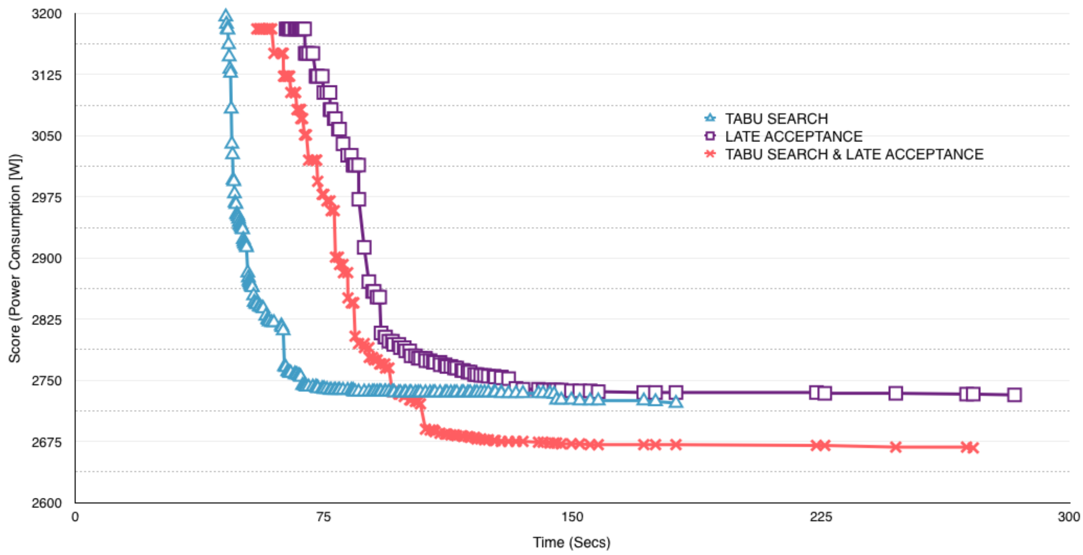

Figure 5.1: Power consumption over time for different local search algorithms

Fig. 5.1 depicts the power consumption over time for the different used algorithms (TS, LAHC and both jointly), thus showing that the LAHC algorithm provides the worst result while the best is achieved using TS and LAHC jointly.

Moreover, from the graph it can be seen all of them find new best solutions very fast at the beginning while over time the number of new solutions is lower and they appear with lower frequency. [1].

On the other hand, it can be appreciated that, with the same termination conditions, the first algorithm to find its best solution is TS, so it might be said that it is the fastest one. However, even though regarding to the time, the combination of both is a worst algorithm, due to it finds a good solution in a longer period of time, it can be appreciated that for a time around 100 secs it reaches a pretty good solution (much better than the algorithms separately). This is the reason why along this project the used algorithms are TS and LAHC jointly for all the test and scenarios.

[1] This time, the termination condition about the improvements was disable in order to see how the algorithms behave over time.

5.3. Impact of using different Γ and Maximum Deviation values

Fig. 5.2 illustrates the power consumption after user joint association for different protection levels (varying Γ from 5 to 55).

The power consumption is shown as power increment over the case with Γ equal to 0 (i.e., no deviation is considered before solving the problem). It is also important to remark that these tests have been executed assuming a 30% demand uncertainty (δ in 2.3), where the uncertainty will be considered as it was explained in Sec.robustness.

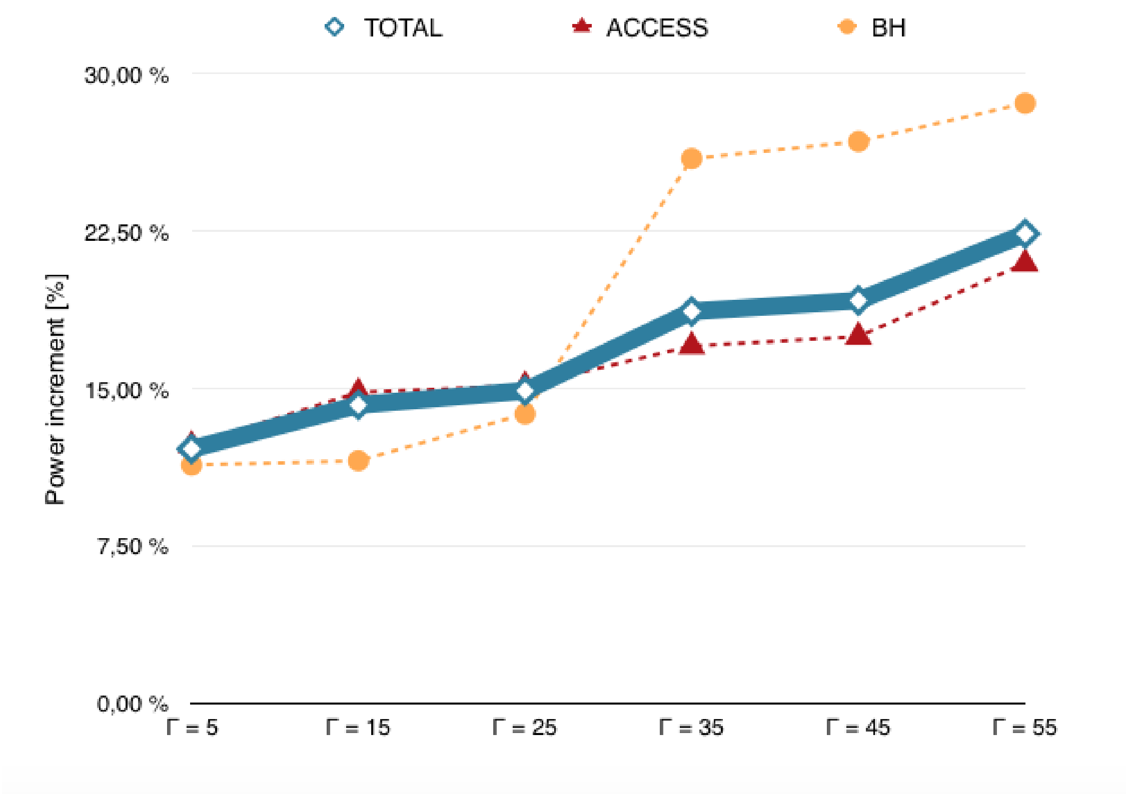

Figure 5.2: Power consumption using different Γ values in a fixed number of iterations, assuming 50% demand uncertainty

From the graph we can extract several aspects. For instance, the higher the value of protection deviation (Γ), the greater the power consumption. This variation is consistent due to the closer the scenario is to the real case (where UEs change randomly their demands), the higher will be the consumed power in the whole network.

Moreover, and as it may be expected, the energy consumption affects to the network but also it affects separately to the AL and to the BH. It is appreciated the effect of the power increment is similar to the total in AL when in the BH the increment is more abrupt. Due to the impact of the demand deviation is made over the number of PRB in the AL but not in the BH, it is difficult to understand how can be the increment in the BH so abrupt and so soft in the AL.

This is because for this specific scenario the UEs connect to the macro (it consumes more power) for the nominal value ($Gamma = 0$). In other scenarios where the solution with the nominal value does not use the macro in the AL, the power increment will be higher in the AL than in the BH.

That way, the increment in the BH reflects a higher number of required links. It can be appreciated for Γ 35 and 45.

As mentioned, these results may change along different scenarios. However, the idea is to represent the increment power consumption while the different values of Γ are incremented.

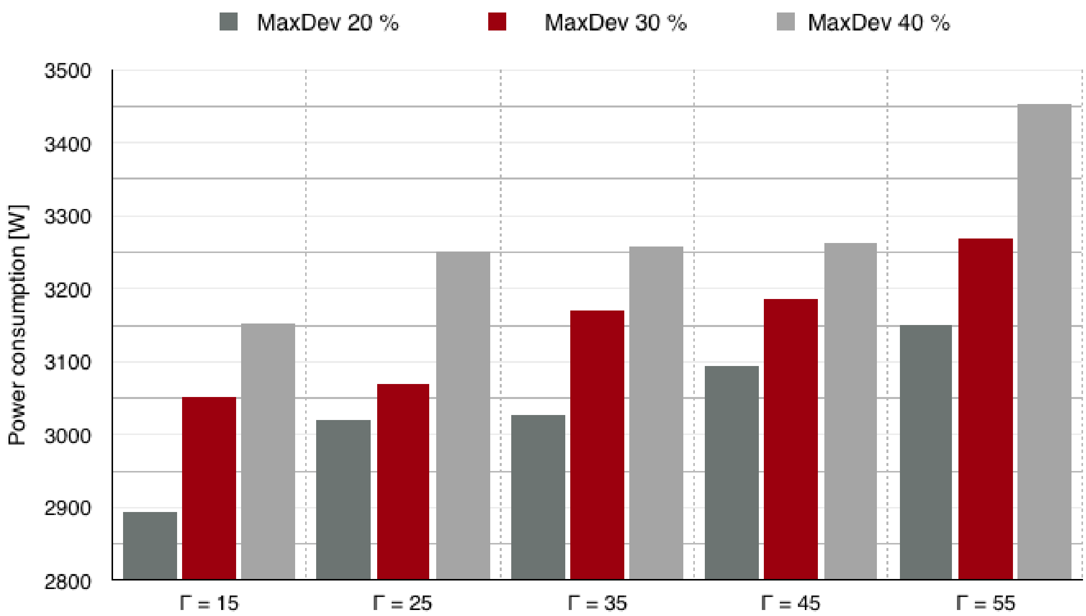

Figure 5.3: Power consumption using different Γ values in a fixed number of iterations, over different demands uncertainty

On the other hand, and as it can be appreciated in Fig. 5.3, if we increment the maximum deviation value from 20% to 40%, the power consumption is bigger for the same protection level (Γ). Besides, for the same maximum deviation, the power consumption also increases with Γ.

We can observe that, with 40% of maximum deviation, the power consumption roughly remains constant for Γ between 25 and 45. This is because the number of ALs used is similar in the three cases. As the power consumption in AL follows Eq. 3.8, it is more notable to use more ALs than increasing the number of PRB (using the same number AL). By contrast, for $\Gamma = 15$ and $\Gamma = 55$ the power is very different as the number of used ALs is bigger for $\Gamma = 55$.

5.4. Impact of different time termination values

In order to check the accuracy of the developed robust heuristic over time, Fig. 5.4 represents different values of power consumption for different Γ values (Γ = 0 and 4) and setting different termination conditions (60 sec, 90 sec, ..., 360 sec).

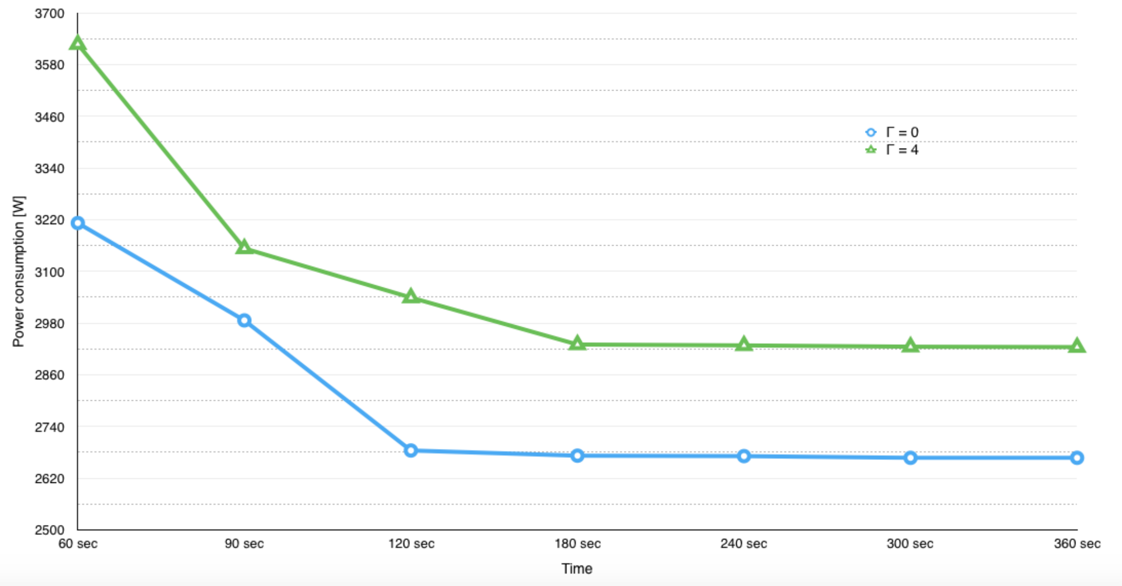

Figure 5.4: Power consumption over time (termination condition) and for different Γ

Fig. 5.4 have been made using the same scenario of 100 UEs. Besides, the maximum deviation allowed for each UE is 30%. From the graph it can be appreciated that the heuristic, when using more time for calculations and making changes to find new possible solutions, is able to find better solutions. This is, the more time the process is solving the lower will be the power consumption of the network. On the other hand, it can be also appreciated the influence of Γ values. In other words, when the value of Γ increases, the value of the power consumption is higher even though having more time execution produces better solutions.

In both cases, as of some time (different for each Γ values and it increases with Γ values) the heuristic tends to not find better solutions. Therefore, from now on we will use 360 seconds as termination value. However, as it depends on Γ values, finding the best termination value, for every Γ values, may be part of future work.

5.5. Impact on the energy consumption

Fig. 5.5 shows the increment of the global power consumption betweeen the utilization or not of the proposed algorithms.

The graph depicts the difference power consumption between using just the algorithms that OptaPlanner uses by default (First Fit and Late Acceptance with a size list of 400) and using the proposed ones. This graph highlights the importance of the use of the selected algorithms, independently of the value of Γ, since the power consumption is incremented when not using the proposed algorithms.

These tests have been made using $\Gamma = 0, 5, 10, 15$. Even though the difference can not seem really notable, it is important to remind the scenario is not big enough, after all it has only 100 UEs. However, this effect may increase for a bigger scenario where the set of solutions to explore is also bigger.

From the figure we can also deduce the produced increment does not depend on the Γ value.

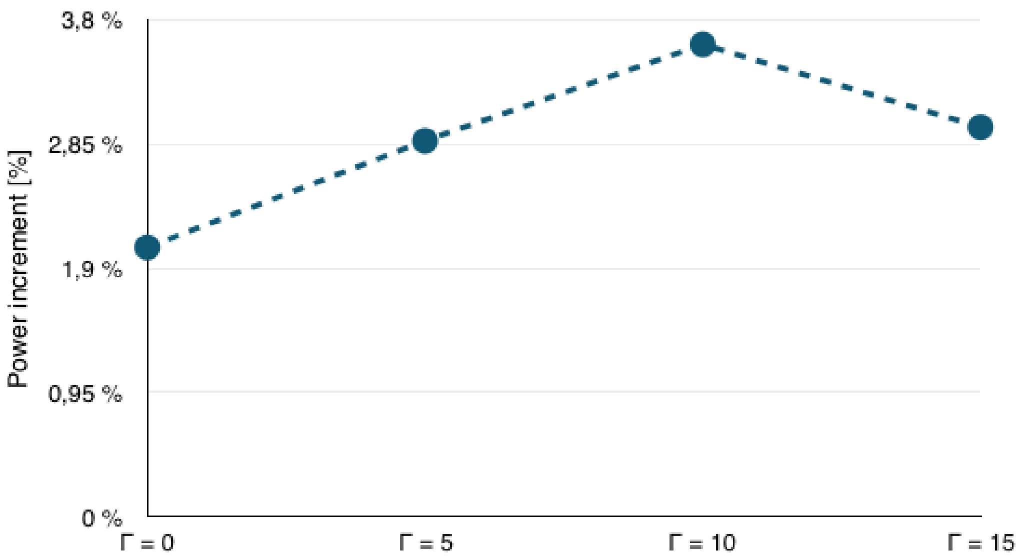

Figure 5.5: Comparison power increment over different values of Γ and sorting

5.6. Grade of service (UE satisfaction)

One interesting exploration is how uncertain resource demand of UEs can lead to overload situation. This increases the possibilities of deriving into service-level agreement (SLA) violations. In order to produce the results, we resolve the problem using the designed heuristic. And after that we generate 10,000 random iterations of UE demands. Those demands are located within the interval [-50%, +50%]. For each generated iteration we check if the UE allocation, obtained by the heuristic, is feasible or not. This decision is made checking the capacity constraint for every SC measured in PRBs.

We use the concept of Realized robustness (RR) as a metric to evaluate the robustness of the final allocation with random demand deviations. This is, using the solution obtained that contemplates the worst case where a Γ number of users deviate their demands using a maximum deviation, we try to check using a random demand (bounded for the same interval) for each user in the same scenario if the obtained solution remains feasible. That way, the RR concept is measured as the ratio of feasible instances over the total number of instances (10,000) [2].

A representation of this concept for scenarios with 20, 40, 60, 80 and 100 UEs, with a deviation of $\geq$ 50%, is shown in Fig. 5.6. From this figure it can be extracted the number of feasible solutions will increase when the number of UEs in the proposed allocation decreases. It depicts that for $\Gamma = 0$, the allocations suffer heavily (i.e., 75% of RR) for UEs demand deviations. As there is no protection applied, many capacity constraints can be violated.

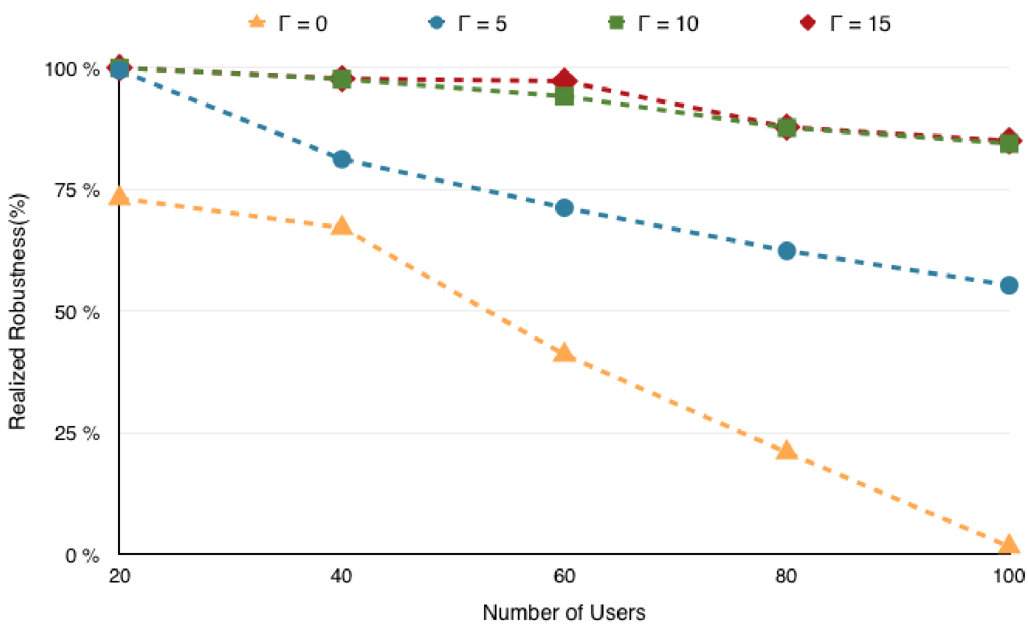

Figure 5.6: Realized Robustness (RR) for different scenarios with *MaxDev* 50%

In Table. 5.1, where further insights on the scenario of 60 UEs are displayed, it can be also appreciated that the higher the value of Γ is, the more difficult to violate some constraints is. Moreover, it is reflected that the demand deviation also impacts on the same solution for a given Γ. This is, there are more feasible solutions when the demand is deviated a 10% rather than a 50%. To sum up, the RR increases rapidly with the protection level (Γ) and we can say that, with Γ=15 we are almost fully protected for the scenarios with up to sixty users.

On the other hand, we use the concept of Price of robustness (PoR) to quantify the extra power consumption required for the robust solution. PoR is calculated as the difference between the obtained power consumption with the nominal solution (which is obtained for $\Gamma = 0$) and the power consumption obtained for $\Gamma \geq 0$ normalized by the power consump-

[2]This method is better explained in [18]

Table 5.1: Realized Robustness (RR) for case with 60 UEs

Γ	*MaxDev(%)*	*RR(%)*
0	10	82,95%
	30	50,82%
	50	41,03%
5	10	90,16%
	30	77,23%
	50	41,25%
10	10	99,82%
	30	98,17%
	50	94,16%
15	10	99,96%
	30	99,49%
	50	97,26%
30	10	100%
	30	99,98%
	50	99,73%

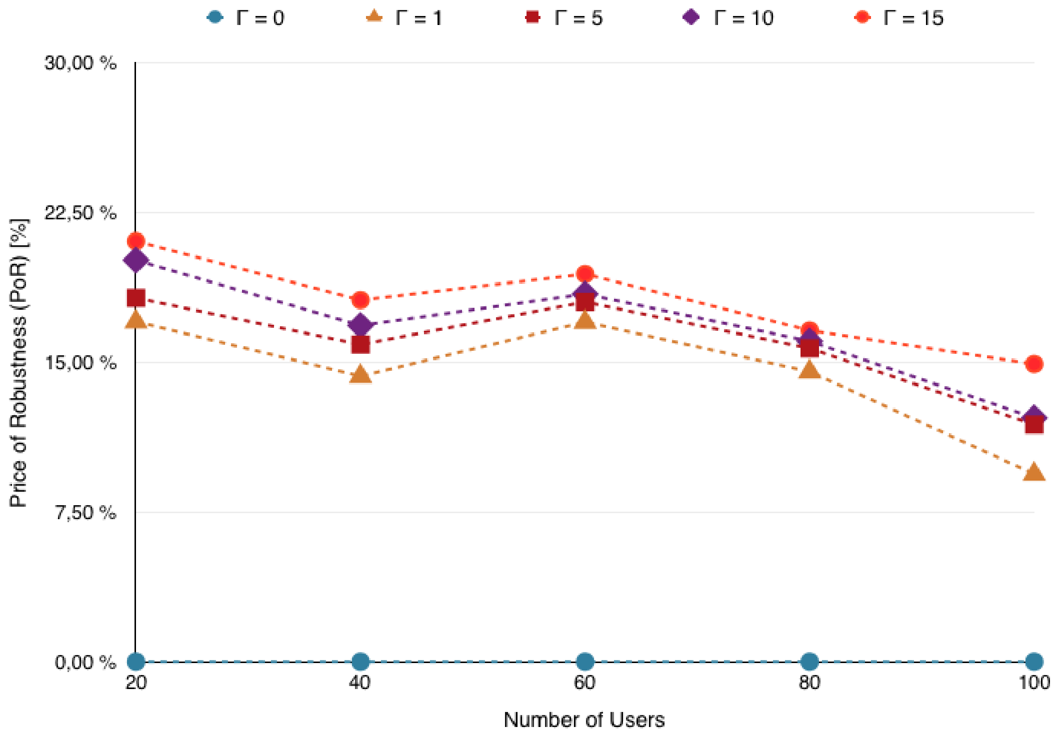

Figure 5.7: Price of Robustness (PoR) for different scenarios with different *MaxDev* values

tion of the nominal solution.

As shown in Fig. 5.7, the PoR is higher for a lower number of UEs and tends to decrease as the number of UEs increases. This is due to the fact that it does not affect the same to deviate a user in a scenario of 20 UEs than in a scenario of 80 UEs.

However, some differences are observed in the power consumption as the power model for different scenarios is selected randomly.

5.7. Impact of heuristic parameters

There exists a compromise between the size of the scenario and the size of the Tabu and Late Acceptance lists. As it was mentioned, both act as local search algorithms that can work separately or simultaneously, but they do not have the same performance.

That way, starting from the Tabu List size, we have to remember that in this list we include UEs, those UEs that have been modified in previous steps, as many as the list size. This avoids OptaPlanner to contemplate any solution that involves changing those UEs. Therefore, the list size has to be a ”correct” number depending on each scenario size. One important point to note is that this list can not be greater than the number of UEs, due to that will allow the heuristic to change every UE just once, so this will be more restricted. However, the shorter the list the less restrictive the local search will be.

Something similar occurs with the Late Acceptance List size. However, this time will include into the list the score of solutions in previous steps, as many steps as the list size. That way, the longer the list is, the less impact the algorithm will have on the solution. This occurs due to if the list is bigger than the made number of steps, it will translate into not using any algorithm. While the shorter this list is, the more restrictive will be the local search. For instance, if this list will be 1 length size, we will force to the heuristic to found a new best score solution each step.

CHAPTER 6. CONCLUSIONS

This chapter will include some conclusions that can be extracted from the development of this project. Moreover some next steps will be enumerated as possible future work. Finally, there is a personal opinion after the work is finished.

6.1. Conclusions

The main objective of the thesis consists on developing a fast and robust heuristic model for the association user problem in a 5G HetNet. In the end, this work has developed an heuristic using OptaPlanner based on TS and LAHC.

To do so, initially we have studied the OptaPlanner tool using some of its examples. After that, we have designed the best approach (UML diagram, class definitions, etc.) to solving our optimization problem with the OptaPlanner environment. In order to do that, some basic scenarios, that could be manually calculated, where we have designed and implemented, in such a way that the proper functioning of the system could be tested. Once the system has been validated, we have proceeded to evaluate a set of different scenarios with a varying amount of users.

Finally, a robust approach to the problem was also implemented by the incorporation of the aforementioned Γ parameter and the corresponding maximum deviation percentage. That allows us to obtain more realistic scenarios where variations in the UE demands are considered.

Regarding the obtained result we can extract several conclusions. On the one hand, the use of different LS algorithms is important, as they may change the performance of the heuristic. The effect of Γ parameter is also important, which allows the heuristic to be close to more realistic scenarios, however we have to assume the global power consumption will increase. On the other hand, different execution times have an impact on the performance of the algorithm, as it have been proven that they may end into worst approaches.

Besides, the proposed heuristic can be applied to any scenario independently of how many VNs, macros, SCs or UE presents. The results obtained behave similar for all the scenarios (see Appendix A). Moreover, this approach can be easily applied to any local search algorithms such as Simulated Annealing, Hill climbing, etc.

Furthermore, our heuristic can be tuned according to the level of robustness required for the network. This provides to our heuristic plenty of functionalities, because varying some parameters the results may change completely.

6.2. Future work

As future work, the following extensions are proposed, which are considered interesting for complementing and improving the developed heuristic.

Firstly, the development of the robust exact model in order to compare the obtained results between the heuristic and the exact models. The non disposition of that model has not made possible to compare the results with the exact model.

Secondly, there is a long path in order to find the best combination of size list of the two different LS algorithms. As it was described earlier, it has to exist a balance between them in order to find the most accurate solution. It does not mean that greater values will produce better solutions.

Regarding the environmental impact of this project it is required to say that the development of software always needs hardware that consumes electricity, with the associated economic impact and CO2 emissions. However, the proposed heuristic tries to find a solution of the association user problem trying to reduce the global power consumption, we may say, in that sense, this project tries to colaborate with the reduction of the environmental impact.

Moreover, in terms of ethical considerations of the project, we may highlight they may not be negative as this project proposes an energy-efficient heuristic that allows to reduce the environmental and economic impact.

6.3. Personal Opinion

As a student of a Master's degree this project was a challenge for me in several ways. On the one hand, it supposed not only to learn new concepts such as robustness or heuristic problems, but also to elaborate on some concepts such as the optimization problems. The latter was a topic which was present along the Master's courses.

On the other hand, the approach assumes also to program using tools like Java, so I could learn more about this language program, which I already knew, but even though sometimes it is hard to translate some ideas into programming language.

Furthermore, and after the development of the whole heuristic, I am very happy with the pursued objectives, not only because the heuristic works correctly, but also because the interesting results obtained so far. In fact, we are going to submit these results to a conference in the near future and we are currently working on the draft of the paper. Also, this study is part of the CICYT project TEC2013- 48099- C2-1-P and the SOCRA project.

BIBLIOGRAPHY

[1] OptaPlanner - Constraint satisfaction solver. https://www.optaplanner.org/. [Online; accessed October-2017]. ix, 12, 14, 23

[2] Ye, Qiaoyang, Rong, Beiyu, Chen, Yudong, Al-Shalash, Mazin, Caramanis, Constantine, and Jeffrey Andrews. User association for load balancing in heterogeneous cellular networks. 12, 05 2012. ix, 15

[3] A. Mesodiakaki, Enrica Zola, and Andreas Kassler. Joint user association and backhaul routing for green 5g mesh millimeter wave backhaul networks. *to appear in ACM MSWiM '17 (under publication)*, 2017. 1, 2, 16, 17, 51

[4] NGMN Alliance. NGMN 5G white paper. Vol. 1, Feb. 2015. 5, 6, 51

[5] Agapi Mesodiakaki and Enrica Zola. Software defined small cell RAN optimization. *SOCRA Project*, Jan. 2016. 6

[6] L. Wei, R. Hu, Y. Qian, and G. Wu. Key elements to enable millimeter wave communications for 5G wireless systems. *IEEE Wireless Communicaitions*, Vol. 21:pp. 136.143, Jan. 2015. 6

[7] Small Cell Forum. Backhaul technologies for small cells: use cases, requirements and solutions. *tech. rep*, 049.04.02, Feb. 2013. 6

[8] NGMN Alliance. Small cell backhaul requirements. *tech. rep*, Jun. 2012. 6

[9] T. S Rappaport *et al*. Millimeter wave mobile communications for 5G celular: It will work! *IEEE. Access*, Vol. 1:pp. 335–349, May. 2013. 6

[10] Stelios H. Zanakis and James R. Evans. Heuristic optimization: Why, when, and how to use it. *Interfaces*, Vol. 11(No. 5):pp. 84–91, (Oct., 1981). 7

[11] F. Glover and M. Laguna. Tabu search. *Norwell, MA, USA: Kluwer Academic Publishers*, 1997. 8

[12] Edmund K. Burke and Yuri Bykov. The late acceptance hill-climbing heuristic. *Technical Report CSM-192, ISSN 1460-9673*, June 2012. 9

[13] D. Bertsimas and M. Sim. The price of robustness. *Operations Research*, Vol. 52(No. 1):pp. 35–54, 2004. 10

[14] D. Bertsimas, David B. Brown, and Constantine Caramanis. Theory and applications of robust optimization. 2007. 11

[15] N. Sapountzis, T. Spyropoulos, N. Nikaein, and U. Salim. User association in over and under-provisioned backhaul hetnets. *Eurocom Research Report*, pages 16–318, Apr. 2016. 16

[16] Q.Han, B. Yang, C. Chen, and X.Guan. Backhaul-aware joint user association and resource allocation for energy-constrained hetnets. *IEEE Trans. Veh. Technnol*, Vol. 66(No. 1):pp. 580–593, Jan. 2017. 16

[17] A. Mesodiakaki et al. Energy and spectrum efficient user association in millimeter wave backhaul small cell networks. *IEEE Trans. Veh. Technol*, Vol. 66(No. 2):pp. 1810–1821, Feb. 2017. 16

[18] Robayet Nasim and Andreas J. Kassler. A robust tabu search heuristic for VM consolidation under demand uncertainity in virtualized datacenters. 16, 41

[19] Shingeru Tsubakitani and James R. Evans. Optimizing tabu list size for the travelling salesman problem. *Computers Ops Res.*, Vol. 25(No. 2):pp. 91–97, 1998. 16, 35

[20] E. Dahlman, S. Parkvall, and J. SkÃüld. 4g: Lte/lte-advanced for mobile broad-band. *Academic Press, ISBN: 012385489X*, 2011. 19

[21] G. Auer et al. How much energy is needed to run a wireless network? *IEEE Wireless Commun*, Vol. 18(No. 5):pp. 40–49, Oct. 2011. 20, 51

[22] A. Mesodiakaki et al. Energy efficient line-of-sight millimeter wave small cell backhaul: 60, 70, 80 or 140 ghz? *Proc. 17th IEEE WoWMoN*, Jun. 2016. 21, 51

[23] T.S Rappaport. E-UTRA & E-UTRAN: Overall description; stage 2. *3GPP TS 36.300*, Vol. 11.5.0:Rel. 11, Mar. 2013. 51

[24] Small cell enhancements for E-UTRA & E-UTRAN-physical later aspects. *3GGP TR 26.872*, Vol. 1.0.0:Rel. 12, Aug. 2013. 51

[25] `https://www.gotmic.se/`. Documents: gTSC0020, gAPZ0039, gRSC0016, gRSC0015, gTSC0023, gAPZ0042. 51

[26] T.S Rappaport. Wireless communications: Principles and practice. *Prentice Hall*, 2nd ed.:pp.153–154. 51

APPENDIXS

APPENDIX A. RESULTS

In this Appendix we try to show some of the results that have been done along the development of this Master thesis. In order to demonstrate the heuristic behaves correctly independent of the used scenario, in this Apendix we will try to summarize some results with different scenarios.

Firstly, in Table A.1 we can find the parameters used in the performance evaluation. They have been also used in the exact model [3].

Table A.1: Parameters used in the performance evaluation

Parameter	AL: eNB	AL: SC	BH link
Frequency f (GHz)	2		73
Available BW (MHz)	20 (100 PRBs)		500
N_{TRX}	8 (MIMO 8x8) [23]		variable
p_0(W)	130 [21]	6.8 [21]	3.9 [21]
p_{max}(W)	39.8107 [24]	1 [24]	1.9953 [25]
Δ_p	4.7 [21]	4.0 [21]	variable
Path Loss	69.55+26.16 log$f$ - 13.82 log$h$-C_H+(44.9-6.55log$h$)log($d_{iu_{km}}$) [26]		Eq. 6-11 in [22]
C_H	0.8 + (1.1 log$f$ - 0.7) h_{UE} - 1.56 log f [26]	0	-
h(m)	25 [24]	2.5 [4]	-
	h_{UE} = 1.5 [24]		
NF(dB)	9 [24]		6 [25]
G_{TX}, G_{RX} (dB_i)	17 [24]	5 [24]	43 [22]

Firstly, in order to guarantee the use of both local search algorithms simultaneously (TS and LAHC), in Fig. A.1 we can see that for all of the cases, the combination of both reaches better solutions (lower power consumption) in a lower period of time.

Moreover, as we said in 5 there are several scenarios with different eNodeB (eNB) distributions, such as:

- Scen A: 5 UEs, 1 VN, 1 macro and 16 SCs.
- Scen B: 100 UEs, 1 VN, 1 macro and 15 SCs.

As a summary, in Fig. A.2 has been represented the different power consumption for different Γ values and maximum deviation values.

Both, Scen A and Scen B, show a similar behavior for different values of Γ and maximum deviation. The heuristic proposes solutions with higher power consumption when the value of Γ increases. Moreover, the same occurs when the maximum deviation increases.

However, despite of the fact they have been represented in the same figure, they are completely different scenarios. Moreover, they have been tested using different parameters. Nevertheless, it allows to represent the correct performance of the developed heuristic, independently of the used scenario and/or the used protection level and maximum deviation values.

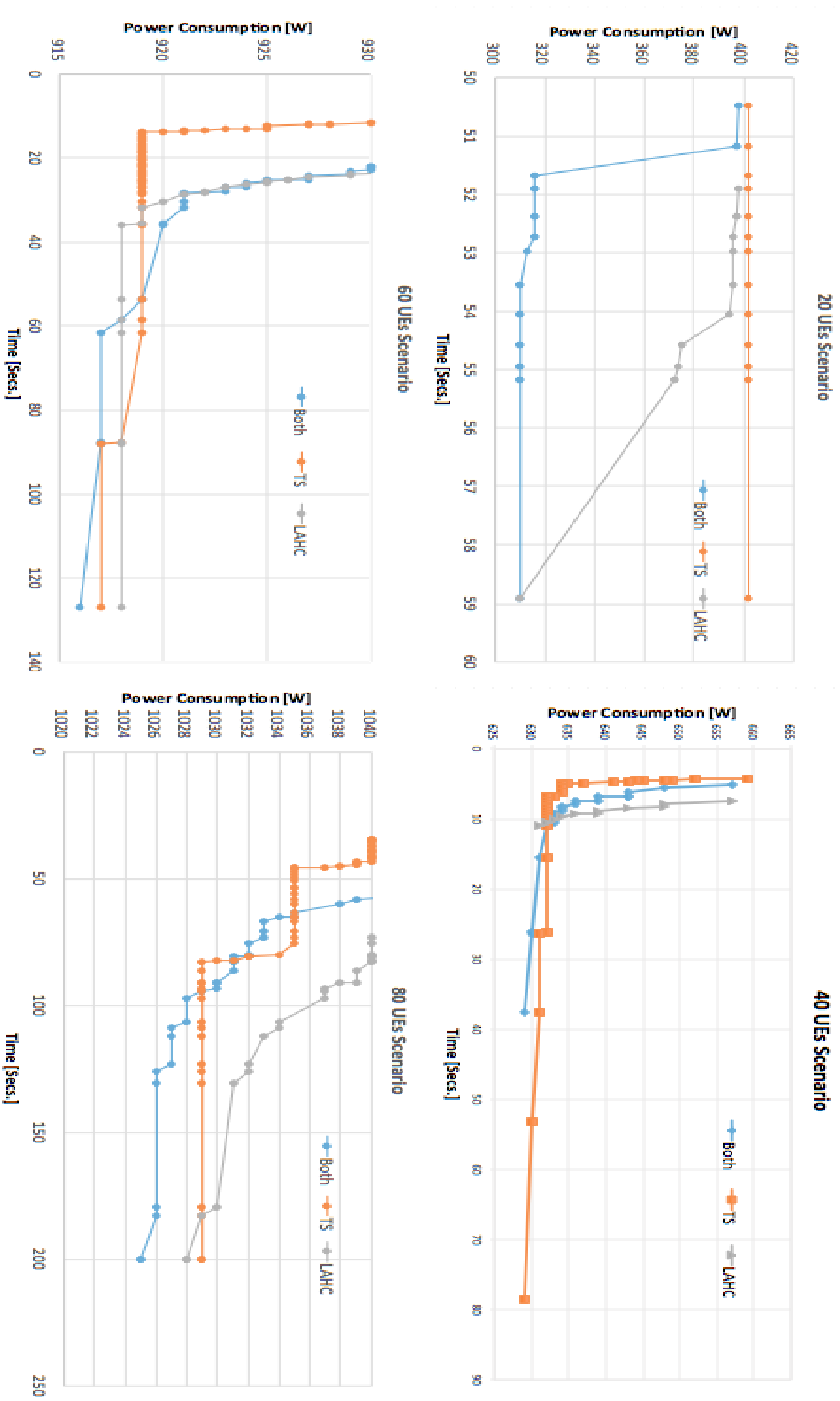

Figure A.1: Comparison of different scenarios (20, 40, 60 and 80 UEs using TS, LAHC or both algorithms.

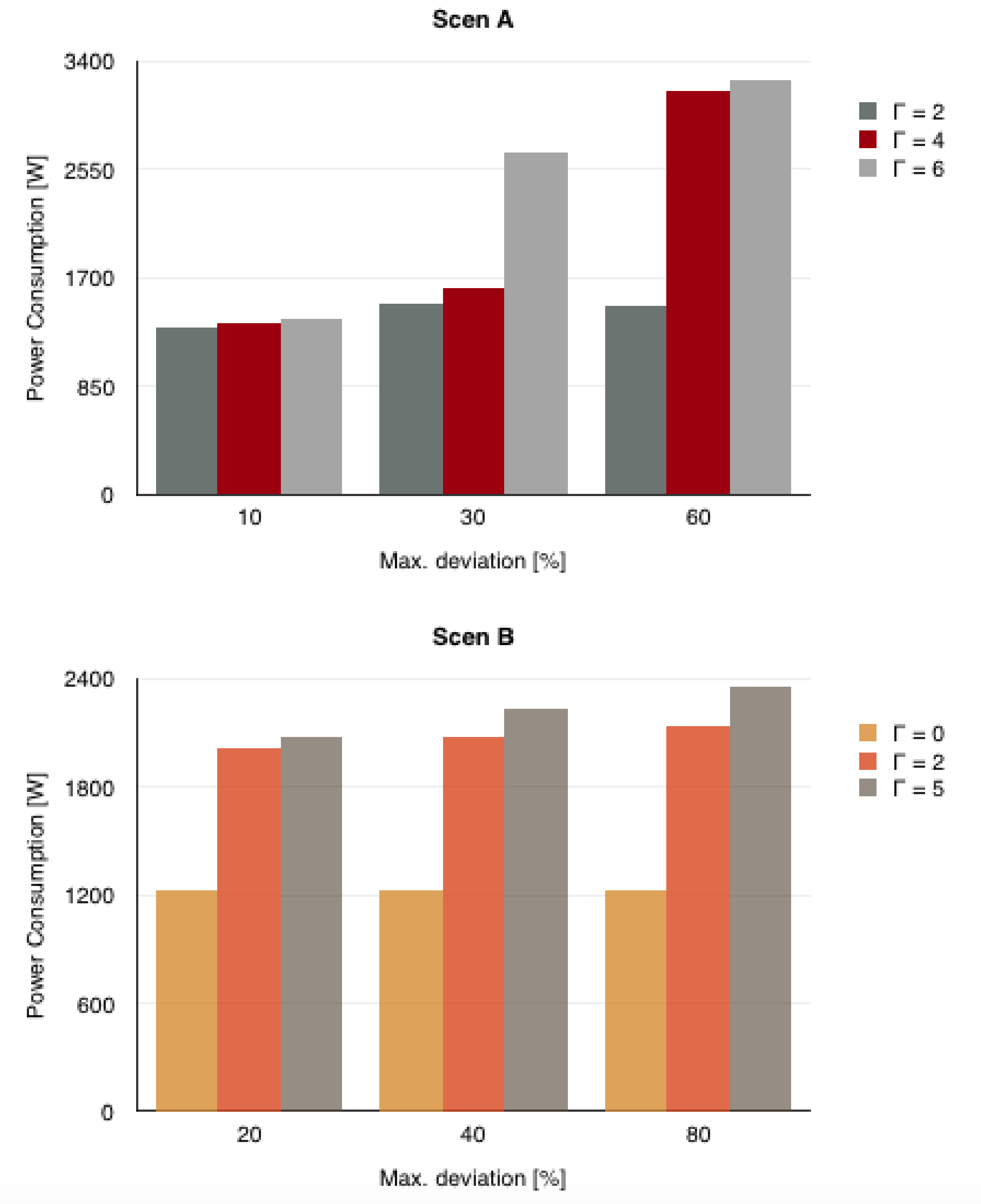

Figure A.2: Power consumption for different Γ and maximum deviation values for Scen A and B