Antenna Optimization in Long-Term Evolution Networks

Qichen Deng

Master of Science Thesis
Stockholm, Sweden 2013
Antenna Optimization in Long-Term Evolution Networks

Q I C H E N D E N G

Master’s Thesis in Optimization and Systems Theory (30 ECTS credits)
Master Programme in Mathematics (120 credits)
Royal Institute of Technology year 2013
Supervisor at KTH was Anders Forsgren
Examiner was Anders Forsgren

TRITA-MAT-E 2013:09
ISRN-KTH/MAT/E--13/09--SE
Abstract

The aim of this master thesis is to study algorithms for automatically tuning antenna parameters to improve the performance of the radio access part of a telecommunication network and user experience. There are four different optimization algorithms, Stepwise Minimization Algorithm, Random Search Algorithm, Modified Steepest Descent Algorithm and Multi-Objective Genetic Algorithm to be applied to a model of a radio access network. The performances of all algorithms will be evaluated in this thesis. Moreover, a graphical user interface which is developed to facilitate the antenna tuning simulations will also be presented in the appendix of the report.

Key words. Multi-Objective Optimization, Non Dominated (Pareto Optimal) Solutions, Pareto Front, Cost Function, Stepwise Minimization Algorithm, Random Search Algorithm, Modified Steepest Descent Algorithm, Multi-Objective Genetic Algorithm, Graphical User Interface (GUI).
## Contents

1. **Introduction**  
   1.1 Background ........................................ 7  
   1.2 Definitions and Terminologies ..................... 7  

2. **Problem Description and Model Formulation**  
   2.1 Main Simulation Parameters and Definitions ........ 9  
   2.2 Constraints and Objective Function ................. 15  
   2.3 Work Flow ........................................ 16  

3. **Stepwise Minimization Algorithm**  
   3.1 Cost Function ...................................... 17  
   3.2 Algorithm .......................................... 18  
   3.3 Algorithm Analysis and Result ...................... 20  

4. **Random Search Algorithm**  
   4.1 Algorithm .......................................... 23  
   4.2 Algorithm Analysis and Result ...................... 23  

5. **Modified Steepest Descent Algorithm** ......... 26  

6. **Genetic Algorithm for Multi-Objective Optimization**  
   6.1 Methodology ....................................... 30  
   6.2 Algorithm Procedure ................................ 30  

7. **Performance Evaluations of Algorithms** ....... 32  

8. **Discussion** ........................................... 39  

9. **Conclusion** ........................................... 42
1. Introduction

1.1. Background

In mobile networks it is vitally important to provide radio channels ensuring good quality voice or Internet services. Radio network tuning aims at optimizing various network parameters in order to improve user perceived quality. Modern antenna design allows electrical (non-mechanical) adaptation of the antenna pattern, enabling flexible orientation of the main antenna lobe in the vertical (tilt) and horizontal (azimuth) plane, as well as the width of the antenna lobe. With the introduction of these modern antennas in radio base stations, there are more controllable parameters which – when properly optimized – can increase the quality and capacity of the radio network. One way to handle this is to replace manual tuning methods by automated and ”self-optimizing” mechanisms enabling the radio network itself to find appropriate settings.

The aim of this thesis is to investigate different multi-objective optimization algorithms and propose one attractive solution. There are nine sections in this report. In Section 2, the radio network is introduced and an antenna tuning problem is formulated into a multi-objective optimization problem. From Section 3 to Section 6, four different algorithms will be applied to the multi-objective optimization problem. The performance evaluation of different algorithm is presented in Section 7. Section 8 contains a discussion and Section 9 is the conclusion of this thesis project.

1.2. Definitions and Terminologies

First of all, some definitions and terminologies [1] are introduced before deriving the optimization algorithms. Given a multi-objective optimization problem:

$$\min_{x \in X} F(x)$$ (1.1)

where $F(x) = \{f_1(x), f_2(x), ..., f_m(x)\}$ is a group of Conflicting Objective Functions, $X$ is the Feasible Region, the variable $x = \{x_1, x_2, ..., x_n\} \in X$ is called Decision Vector, each $x_i$ is called Decision Variable. Here Conflicting Objective Functions implies at least two of the the objective functions can not be minimized simultaneously. Furthermore, the image of the feasible region in the objective space is called a Feasible Objective Region $F(X)$.

A decision vector $a \in X$ is said to Dominate a decision vector $b \in X$ if and only if:

$$\forall i \in 1, ..., m : f_i(a) \leq f_i(b)$$ (1.2)

$$\exists j \in 1, ..., m : f_j(a) < f_j(b)$$ (1.3)

A decision vector $x \in X$ is called Pareto Optimal or Non Dominated if there does not exist another $x' \in X$ that dominates $x$. If $\{x^1, x^2, ..., x^n\} \subset X$ are the Pareto optimal solutions, $\{F(x^1), F(x^2), ..., F(x^n)\}$ is called the Pareto Front.
The definitions are illustrated by the following example:

$$\text{minimize } F(x) \quad (1.4)$$

where $$F(x) = \{x^2, \ (x - 3)^2\}$$. The Pareto optimal solution is $$0 \leq x \leq 3$$. The solution $$x = -1$$ is dominated by $$x = 1$$ since:

$$1^2 \leq (-1)^2$$

$$(1 - 3)^2 < (-1 - 3)^2 \quad (1.5)$$

In most of the cases, it is difficult or computationally expensive to find all the global Pareto optimal solutions. In this thesis report, the Pareto optimal solutions actually mean the non dominated solutions which have been investigated.
2. Problem Description and Model Formulation

In this section the radio network is introduced that will be studied in the rest of the report.

2.1. Main Simulation Parameters and Definitions

**Network**: Stockholm Center

In this thesis report, a cellular access network covering Stockholm Center is studied. It consists of 16 sites (base stations) and 44 cells.

![Figure 2: Landscape of Stockholm Center](image)

![Figure 3: Network Map of Stockholm Center](image)

In the Figure 3, each circle stands for one site and it can be seen that each site contains a 3-sector antenna. Different colors mean the height of buildings in the network map.
**Inter Site Distance**: 250 meters

The Inter Site Distance is the distance between two sites. In fact, if two sites are close to each other (less than 250 meters), there will be strong interference and the network performance will be deteriorated.

![Inter Site Distance](image)

**Network**: LTE

LTE (Long Term Evolution [2]) network, also marketed as 4G network, is a wireless broadband technology designed to support roaming Internet access via cell phones and handheld devices.

**Antenna**: An antenna is used to convert the guided waves in a feeder cable or transmission line into a radiating wave traveling in the free space. This thesis focuses on 3-sector antenna, each 3-sector antenna consists of 3 antennas.

![3-Sector Antenna](image)
There are three main parameters in one antenna, **Antenna Downtilt**, **Azimuth** and **Beam Width**:

- **Antenna Downtilt**: The Antenna Downtilt is the angle between the direction of antenna main lobe and horizon (the default setting is 8 degree):

![Antenna Downtilt Diagram](image)

**Figure 6: Antenna Downtilt**

- **Azimuth (Horizontal Direction)**: One of the antenna parameters. The azimuth angle is the compass bearing, relative to true (geographic) north, of a point on the horizon directly beneath an observed object.

![Default Setting of Horizontal Directions](image) ![New Horizontal Directions](image)

**Figure 7: Antenna Horizontal Directions**

- **BeamWidth**: The antenna BeamWidth along the main lobe axis in a specified plane is defined as the angle between points where the power density is one-half the power density at the peak:
SINR: Signal to Interference plus Noise Ratio (SINR) is commonly used in telecommunication as a way to measure the quality connections:

$$\text{SINR} = \frac{P_{\text{Signal}}}{P_{\text{Interference}} + P_{\text{Noise}}}$$  \hspace{1cm} (2.1)

Where $P_{\text{Signal}}$ is the received signal power which depends on the power of transmitter and path loss between sender and receiver. $P_{\text{Interference}}$ is the power of interference that comes from neighboring cells (other transmitters). $P_{\text{Noise}}$ is the thermal noise in the receiver which is consider fixed.
The SINR can be improved by strengthening transmitter power, or reducing the inter-cell interference. Changing antenna parameters properly will increase $P_{\text{Signal}}$ and decrease $P_{\text{Interference}}$, thus the SINR will be improved.

**Channel**: In telecommunications and networking, a (communication) channel, or radio channel is used to convey an information signal from one or several senders (or transmitters) to one or several receivers. A channel has a certain capacity for transmitting information, often measured by its bandwidth in Hz or its data rate in bits per second.

- **Downlink (Channel)**: A downlink channel, also called forward channel, has one transmitter sending signals to many receivers [3].

- **Uplink (Channel)**: An uplink channel, also called a multiple access channel or reverse channel, has many transmitters sending signals to one receiver, where each signal must be within the total system bandwidth $B$ [3].

![DownLink and UpLink Channels](image)

**Figure 10**: DownLink and UpLink Channels

**Number of Sampling Users**: 2,000
It means there are 2,000 users randomly sampled to probe the traffic of the network.
**User Throughput**: Measure of user perceived quality expressed as the data transfer rate of useful and non-redundant information. It depends on factors such as bandwidth, line congestion, error correction, etc and it is also a measure of users’ satisfaction. For example, if the 10 percentile downlink user throughput is 10 Mb/s, it means 10% of the users that have downlink throughput less or equal than 10 Mb/s. Take a look at the following figure:

![Figure 11: Example of User Throughput](image)

Each curve stands for a network performance with some network parameters (Electrical Downtilt, Azimuth and BeamWidth). One can easily figure out that the User Throughput 2 (the curve on the right) is better than User Throughput 1, since with the same percentile (y-value) the x-value in User Throughput 2 is greater than User Throughput 1. This also implies more users have better perceived quality (or simply data transfer rate) with the parameter setting of User Throughput 2.

The **User Throughput** is also the objective function of the antenna optimization problem. The aim is to shift the curve to the right as much as possible by changing some of the antenna parameters in the radio network.
2.2. Constraints and Objective Function

This subsection introduces constraints and objective function of the optimization problem:

**Constraints:** There are four kinds of decision variables in one antenna: **Electrical DownTilt, Horizontal Direction, Horizontal BeamWidth** and **Vertical BeamWidth.** To limit the search area, this thesis assigns the following lower bound and upper bound for each of the decision variable:

<table>
<thead>
<tr>
<th>Decision Variable</th>
<th>Lower Bound (deg)</th>
<th>Upper Bound (deg)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Electrical DownTilt (eT)</td>
<td>0</td>
<td>12</td>
</tr>
<tr>
<td>Horizontal Direction (hD)</td>
<td>-45</td>
<td>45</td>
</tr>
<tr>
<td>Horizontal BeamWidth (hBW)</td>
<td>0</td>
<td>13</td>
</tr>
<tr>
<td>Vertical BeamWidth (vBW)</td>
<td>0</td>
<td>130</td>
</tr>
</tbody>
</table>

Table 1: Lower Bounds and Upper Bounds of Decision Variables

**Objective Function:** The low percentile ($\leq$ 10 percentile) of the downlink and uplink user throughputs are considered as objective functions in this study. The **User Throughput** is a black-box function of **Electrical DownTilt, Horizontal Direction, Horizontal BeamWidth** and **Vertical BeamWidth** that generated by a static simulator (see Appendix B for more information about simulator). There is no information about the mathematical expression of user throughput.

The task is to maximize the 10 percentile downlink and uplink user throughput (denoted by $F(x)$). It can be converted to a minimization problem by multiplying the objective functions with -1. The optimization problem has the following form:

\[
\text{minimize } -F([eT_1, hD_1, hBW_1, vBW_1], ..., [eT_n, hD_n, hBW_n, vBW_n]) \quad (2.2)
\]

Subject to:

\[
0 \leq eT_i \leq 12, \quad i = 1, 2, ..., n \quad (2.3)
\]

\[
-45 \leq hD_i \leq 45, \quad i = 1, 2, ..., n \quad (2.4)
\]

\[
0 \leq hBW_i \leq 13, \quad i = 1, 2, ..., n \quad (2.5)
\]

\[
0 \leq vBW_i \leq 130, \quad i = 1, 2, ..., n \quad (2.6)
\]
2.3. Work Flow

![Work Flow Diagram]

Figure 12: Work Flow
3. Stepwise Minimization Algorithm

The Stepwise Minimization Algorithm tunes the antenna parameters (decision variables) one by one. That is, it minimizes the objective functions along each decision variable independently and consecutively. The algorithm begins with the first parameter and fixes the rest, then finds a good enough solution for the first parameter according to the one-dimensional minimal cost problem. After that the first parameter will be fixed to the selected value and the algorithm continues to tune the second parameter, then the third parameter and so on. Besides, all the non-dominated solutions will also be kept as alternatives by the algorithm. In order to run this algorithm, a one-dimensional cost function is introduced that is described in the following subsection.

3.1. Cost Function

Though the given problem is about multi-objective optimization, we convert it into a related single objective one by introducing a cost function which combines the objective functions of the original problem. First of all, define:

\[ x = (x_1, x_2, \ldots, x_n) \]  (3.1)
\[ f(x) = (f_1(x), f_2(x), \ldots, f_m(x)) \]  (3.2)
\[ y = f(x) = (y_1, y_2, \ldots, y_m) \]  (3.3)

Let \( S = \{x^1, x^2, \ldots, x^s\} \subseteq X \) be a set of feasible points and:
\[ y_{i,\min} = \min_{x \in S} \{ f_i(x) \} \]  (3.4)

Figure 13: \( y_{1,\min} \) and \( y_{2,\min} \) in 2 dimensional objective function
\[ y_{\text{min}} = (y_{1,\text{min}}, y_{2,\text{min}}, \ldots, y_{m,\text{min}}) \]  

Consider the following cost function:

\[ \tilde{f}(x) = \sqrt{\sum_{i=1}^{m} \left( \frac{y_i}{y_{i,\text{min}}} - 1 \right)^2} \]  

Here \( y_{i,\text{min}} = -\varepsilon \) if \( y_{i,\text{min}} = 0 \), where \( \varepsilon \) is a very small positive value. The aim of the related single objective optimization problem is to minimize the distance between the ratio \( \frac{y_i}{y_{i,\text{min}}} \) and 1 (See Figure 14).

![Figure 14: Best candidate according to cost function (3.6)](image)

3.2. Algorithm

The **Stepwise Minimization Algorithm** minimizes the cost function (3.6) along one decision variable at a time, which implies that the total steps are equal to the number of decision variables. In each step, it solves a subproblem with only one variable and a set of feasible solutions called the solution candidates are investigated, e.g., the subproblem has the following form in step \( i \):

\[
\text{minimize} \quad \tilde{f}(x) \\
\text{subject to:} \\
x_1 = \hat{x}_1, \quad x_2 = \hat{x}_2, \quad \ldots, \quad 0 \leq x_i \leq 1, \quad \ldots, \quad x_n = x_n^0
\]
where \( \hat{x}_1, \hat{x}_2, ..., \hat{x}_{i-1} \) are solutions to previous subproblems. \( x_{i+1}^0, x_{i+2}^0, ..., x_n^0 \) are the values from initial solution. Once the subproblem is solved (the algorithm finds the solution \( \hat{x}_i \) to \( x_i \)), the algorithm continues to solve a new subproblem in step \( i + 1 \):

\[
\text{minimize } \tilde{f}(x) \quad (3.9)
\]

subject to:
\[
x_1 = \hat{x}_1, ..., x_i = \hat{x}_i, \quad 0 \leq x_{i+1} \leq 1, ..., x_n = x_n^0 \quad (3.10)
\]

The algorithm stops when the solutions to all decision variables are found.

Let \( Z = \{z^1, z^2, ..., z^k\} \subseteq X \) and define the functions and variables:

- \( \text{minind}(X) - \text{find the index of minimal element of a vector } X. \quad (3.11) \)
- \( \text{randP}(N) - \text{generate } N \text{ uniformly distributed numbers in } [0, 1]. \quad (3.12) \)
- \( \text{prtopt}(Z) - \text{find the non dominated candidates in } Z. \quad (3.13) \)
- \( \hat{x}_i - \text{the solution to } i^{th} \text{ subproblem.} \quad (3.14) \)
- \( x_i^0 - \text{initial value of } x_i. \quad (3.15) \)

Together with (3.6), the algorithm is presented as follows:

**Algorithm 3.1. Stepwise Minimization Algorithm**

\[
\hat{x} \leftarrow x^0 \quad (\text{initial point});
\]
\[
Z \leftarrow x^0; 
\]
\[
N \leftarrow \text{number of solution candidates};
\]

for \( i = 1, ..., n \)

\[
(p_1, p_2, ..., p_{N-1}) \leftarrow \text{randP}(N - 1);
\]
\[
x^1 \leftarrow (\hat{x}_1, \hat{x}_2, ..., \hat{x}_{i-1}, x_i^0, x_{i+1}^0, ..., x_n^0);
\]
\[
Z \leftarrow x^1 \cup Z;
\]

for \( k = 2, ..., N \) do

\[
x^k \leftarrow (\hat{x}_1, \hat{x}_2, ..., \hat{x}_{i-1}, p_{k-1}, x_{i+1}^0, ..., x_n^0);
\]
\[
Z \leftarrow x^k \cup Z;
\]
end

for \( j = 1, ..., m \) do

\[
y_{j,\text{min}} = \min\{f_j(x^1), f_j(x^2), ..., f_j(x^N)\};
\]
end

\[
y_{\text{min}} \leftarrow (y_{1,\text{min}}, y_{2,\text{min}}, ..., y_{m,\text{min}});
\]
\[
cost \leftarrow (\tilde{f}(x^1), ..., \tilde{f}(x^N));
\]
\[
\alpha \leftarrow \text{minind}(\text{cost});
\]
\[
\hat{x}_i \leftarrow p_\alpha;
\]
\[
\hat{x} \leftarrow (\hat{x}_1, \hat{x}_2, ..., \hat{x}_{i-1}, \hat{x}_i, x_{i+1}^0, ..., x_n^0);
\]
\[
Z \leftarrow \text{prtopt}(Z);
\]
end
3.3. Algorithm Analysis and Result

For the purpose of illustration, we consider a simplified example, only 3 antennas of the network are tuned (Antenna 11, 40 and 24, selected by the simulator), each antenna has two decision variables, Electrical Downtilt and Horizontal Direction. Horizontal BeamWidth and Vertical BeamWidth are not considered. The rest of the antennas still have the default settings (See Figure 6 and Figure 7 on Page 11). The number of solution candidates is 10 and all the decision variables are normalized between 0 and 1.

<table>
<thead>
<tr>
<th>Initial Solution (x^0)</th>
<th>Objtv (Mb/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DL</td>
</tr>
<tr>
<td>A 11</td>
<td>A 40</td>
</tr>
<tr>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>A 11</td>
<td>A 40</td>
</tr>
<tr>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td></td>
<td>25.4132</td>
</tr>
</tbody>
</table>

Table 2: Stepwise Minimization Algorithm Initialization

Table 2 shows the initial solution and objective values generated by the simulator. A stands for antenna, DL and UL mean downlink and uplink user throughput respectively, Objtv are the objective values. Let us take a look at the first and second step of the solution procedure:

**Step 1**: The purpose of Step 1 is to find a good solution to decision variable 1. According to the number of solution candidates, 9 new random candidates of decision variable 1 are generated (the first column of solution table), while the rest of the decision variables (column 2 to column 6) are exactly the same as the initial value. The max downlink user throughput and uplink use throughput are **25.4132**Mb/s and **4.7225**Mb/s respectively. Cost function (3.6) implies that candidate 1 has minimal cost and x₁ is set to 0 (x̂₁ = 0). Table 3 shows the details of of Step 1 and Table 4 gives the non dominated solutions after Step 1:

<table>
<thead>
<tr>
<th>Candidates</th>
<th>Objtv (Mb/s)</th>
<th>Max (Mb/s)</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>E-Tilt</td>
<td>H-Direction</td>
<td>DL</td>
</tr>
<tr>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>25.4132</td>
</tr>
<tr>
<td>0.8756</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0.4392</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0.5265</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0.5201</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0.3512</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0.4749</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0.2855</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0.2318</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0.4792</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 3: Stepwise Minimization Algorithm Step 1
3. **Stepwise Minimization Algorithm**

<table>
<thead>
<tr>
<th>Non Dominated Candidates</th>
<th>Objtv (Mb/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>E-Tilt</td>
<td>H-Direction</td>
</tr>
<tr>
<td>DL</td>
<td>UL</td>
</tr>
<tr>
<td>0.0000</td>
<td>0.0000</td>
</tr>
</tbody>
</table>

Table 4: Non Dominated Solutions and Objective Values after Step 1

**Step 2**: The aim of Step 2 is to find a solution to decision variable 2. Since 0 is the solution to decision variable 1, then the first column of solution table is 0, 9 random feasible candidates of decision variable 2 are introduced in the column 2, while column 3 to column 6 remain the same:

<table>
<thead>
<tr>
<th>Candidates</th>
<th>Objtv (Mb/s)</th>
<th>Max (Mb/s)</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DL</td>
<td>UL</td>
<td></td>
</tr>
<tr>
<td>E-Tilt</td>
<td>H-Direction</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.0000</td>
<td>0.0000</td>
<td>25.4132</td>
<td>4.7225</td>
</tr>
<tr>
<td>0</td>
<td>0.9776</td>
<td>25.2686</td>
<td>4.4920</td>
</tr>
<tr>
<td>0</td>
<td>0.4912</td>
<td>25.1345</td>
<td>4.5287</td>
</tr>
<tr>
<td>0</td>
<td>0.8627</td>
<td>25.2698</td>
<td>4.4005</td>
</tr>
<tr>
<td>0</td>
<td>0.3059</td>
<td>25.2961</td>
<td>4.6340</td>
</tr>
<tr>
<td>0</td>
<td>0.2758</td>
<td>25.3490</td>
<td>4.6411</td>
</tr>
<tr>
<td>0</td>
<td>0.0654</td>
<td>25.3930</td>
<td>4.7132</td>
</tr>
<tr>
<td>0</td>
<td>0.2112</td>
<td>25.3261</td>
<td>4.5875</td>
</tr>
<tr>
<td>0</td>
<td>0.1233</td>
<td>25.3679</td>
<td>4.6016</td>
</tr>
<tr>
<td>0</td>
<td>0.0709</td>
<td>25.3886</td>
<td>4.6969</td>
</tr>
</tbody>
</table>

Table 5: Stepwise Minimization Algorithm Step 2

It can be seen that 0 is also the solution to decision variable 2 since its cost is minimum ($\hat{x}_1 = 0$, $\hat{x}_2 = 0$). For more details please refer to **Appendix C**. Table 7 and 8 give the result and non dominated solutions respectively after running the **Stepwise Minimization** algorithm.

<table>
<thead>
<tr>
<th>Best Candidate</th>
<th>Objtv (Mb/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DL</td>
</tr>
<tr>
<td>Electrical Downtilt</td>
<td>Horizontal Direction</td>
</tr>
<tr>
<td>A 11</td>
<td>A 40</td>
</tr>
<tr>
<td>0.0000</td>
<td>0.0000</td>
</tr>
</tbody>
</table>

Table 7: Stepwise Minimization Algorithm Best Candidate
Table 8: Non Dominated Solutions and Objective Values after Step 6

The best candidate is suggested as the final solution to the optimization problem, but the decision maker can also select one of the other non dominated candidates as decision.
4. Random Search Algorithm

The idea of Random Search algorithm is similar to Greedy Algorithm. It generates a set of random feasible solutions according to the number of solution candidates but only keeps the non dominated candidates.

4.1. Algorithm

Given a multi-objective optimization problem:

\[
\text{minimize} \quad f(x) \quad \text{subject to} \quad x \in X
\]  

(4.1)

where \( X \) is the feasible region. Let \( Z = \{z^1, z^2, ..., z^k\} \subseteq X \) and define the functions:

\[
\text{prtopt}(Z) - \text{find the non dominated candidates in } Z. \quad (4.2)
\]

\[
\text{randV}(N) - \text{generate } N \text{ random feasible decision vectors.} \quad (4.3)
\]

Algorithm 4.1. Random Search

\[
x^0 \leftarrow \text{Initial Point};
\]

\[
N \leftarrow \text{number of solution candidates};
\]

\[
Z \leftarrow \text{randV}(N);
\]

\[
Z \leftarrow \text{prtopt}(x^0 \cup Z);
\]

4.2. Algorithm Analysis and Result

The algorithm only tunes 3 antennas (Antenna 11, 24 and 40) for purpose of illustration, each antenna has two decision variables, Electrical Downtilt and Horizontal Direction. All the decision variables are normalized between 0 and 1. Solution candidates are generated by the random function (4.3). The following tables show the initial solution algorithm analysis and result for 10 solution candidates. Assume that the algorithm begins with Table 9:

<table>
<thead>
<tr>
<th>A 11</th>
<th>A 40</th>
<th>A 24</th>
<th>A 11</th>
<th>A 40</th>
<th>A 24</th>
<th>DL</th>
<th>UL</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>25.4132</td>
<td>4.7225</td>
</tr>
</tbody>
</table>

Table 9: Initial Solution and Objective Values
The highlighted candidates in Table 10 are non dominated:

Table 10: Random Search with 10 Solution Candidates

Large number of solution candidates theoretically generates better solutions, for example, Table 12 and Table 13 show the non dominated candidates with 500 and 5000 solution candidates respectively:

Table 11: Non Dominated Candidates - 10 Solution Candidates

Table 12: Non Dominated Candidates - 500 Solution Candidates
Table 13: Non Dominated Candidates - 5000 Solution Candidates

<table>
<thead>
<tr>
<th>E-Tilt</th>
<th>H-Direction</th>
<th>DL</th>
<th>UL</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0324</td>
<td>0.0969</td>
<td>25.8328</td>
<td>5.3754</td>
</tr>
<tr>
<td>0.0407</td>
<td>0.9092</td>
<td>25.4393</td>
<td>5.5448</td>
</tr>
</tbody>
</table>

Figure 15: Non dominated solutions by different number of solution candidates
5. Modified Steepest Descent Algorithm

In this section, we try to solve the given multi-objective optimization problem by a Modified Steepest Descent method which consists of Steepest Descent and Stepwise Minimization. The algorithm selects and minimizes one of the objective functions in each step, this implies the total steps are equal to the number of objective functions. For example, if \( f(x) = (f_1(x), f_2(x), ..., f_m(x)) \), the algorithm solves the following problem in step \( i \):

\[
\min_{0 \leq x \leq 1} f_i(x) \quad (5.1)
\]

Since the objective functions (10 percentile downlink and uplink user throughput) are black-box functions, the gradients of objective functions needs to be approximated.

Assume that there are \( n \) decision variables, the function is in \( m \)-dimensional, \( \Delta h \) is a very small positive scalar and \( x^i \) is the current point:

\[
x^i = (x^i_1, x^i_2, ..., x^i_n)
\]

\[
f(x^i) = (f_1(x^i), f_2(x^i), ..., f_m(x^i))
\]

\[
y_i = f_i(x^i)
\]

\[
\Delta h_1 = (\Delta h, 0, ..., 0) \quad \Delta h_2 = (0, \Delta h, 0, ..., 0) \quad ... \quad \Delta h_n = (0, 0, ..., 0, \Delta h)
\]

\[
y_{i1+} = f_i(x^i + \Delta h_1) \quad ..., \quad y_{i1-} = f_i(x^i - \Delta h_1)
\]

\[
y_{in+} = f_i(x^i + \Delta h_n) \quad ..., \quad y_{in-} = f_i(x^i - \Delta h_n)
\]

\[
\frac{\partial f_i}{\partial x_k^+}(x^i) = \frac{y_{ik+} - y_i}{\Delta h} \quad \text{if~} x_k - \Delta h < x_k, \text{Lower Bound}
\]

\[
\frac{\partial f_i}{\partial x_k^-}(x^i) = \frac{y_{ik-} - y_i}{\Delta h} \quad \text{if~} x_k + \Delta h > x_k, \text{Upper Bound}
\]

\[
0 \quad \text{otherwise}
\]

The approximated partial derivative is (See Figure 16 for more details):

\[
\frac{\partial f_i}{\partial x_k}(x^i) = \begin{cases} 
\frac{\partial f_i}{\partial x_k^+}(x^i) & \text{if~} \frac{\partial f_i}{\partial x_k^+}(x^i) < 0 \text{ and } \frac{\partial f_i}{\partial x_k^-}(x^i) < -\frac{\partial f_i}{\partial x_k^+}(x^i) \\
\frac{\partial f_i}{\partial x_k^-}(x^i) & \text{if~} \frac{\partial f_i}{\partial x_k^+}(x^i) > 0 \text{ and } -\frac{\partial f_i}{\partial x_k^-}(x^i) < \frac{\partial f_i}{\partial x_k^+}(x^i) \\
\frac{\partial f_i}{\partial x_k^+}(x^i) & \text{if~} x_k - \Delta h < x_k, \text{Lower Bound} \\
\frac{\partial f_i}{\partial x_k^-}(x^i) & \text{if~} x_k + \Delta h > x_k, \text{Upper Bound} \\
0 & \text{otherwise}
\end{cases}
\]

And the approximated gradient of the \( f_i(x) \) at \( x^i \) is:

\[
\nabla f_i(x^i) = \left( \frac{\partial f_i}{\partial x_1}(x^i), \frac{\partial f_i}{\partial x_2}(x^i), ..., \frac{\partial f_i}{\partial x_n}(x^i) \right) \quad (5.2)
\]
5. Modified Steepest Descent Algorithm

\[ \frac{\partial f_i}{\partial x_k}(x^i) = 0 \]

\[ \frac{\partial f_i}{\partial x_k}(x^i) = \frac{\partial f_i}{\partial x_k} + (x^i) \]

\[ \frac{\partial f_i}{\partial x_k}(x^i) = \frac{\partial f_i}{\partial x_k} - (x^i) \]

\[ \frac{\partial f_i}{\partial x_k}(x^i) = \frac{\partial f_i}{\partial x_k} + (x^i) \]

\[ \frac{\partial f_i}{\partial x_k}(x^i) = \frac{\partial f_i}{\partial x_k} - (x^i) \]

Figure 16: Partial Derivative Approximation
According to the approximated gradient and the idea of Steepest Descent, the corresponding search direction \( d \) for \( x^i \) is:

\[
d = (d_1, d_2, ..., d_n)
\]  

(5.3)

where [4]:

\[
d_k = \begin{cases} 
\frac{\partial f_i}{\partial x_k}(x^i) & \text{if } \frac{\partial f_i}{\partial x_k}(x^i) = \frac{\partial f_i}{\partial x_k}(x^i) < 0 \\
\frac{\partial f_i}{\partial x_k}(x^i) & \text{if } \frac{\partial f_i}{\partial x_k}(x^i) = \frac{\partial f_i}{\partial x_k}(x^i) > 0 \\
0 & \text{otherwise}
\end{cases}
\]

Notice that \( f_i(x) \) decreases along the search direction \( d \) from point \( x^i \) if \( d_k \neq 0 \) for some \( k \); it means in some neighborhoods of \( x^i \) there exists at least a point \( x^{i+1} \) such that \( f_i(x^{i+1}) < f_i(x^i) \). In order to see how big step \( f_i(x^i) \) can take and how much it will decrease, an step size \( \alpha \) needs to be determined by solving the subproblem:

\[
\min_{x^i + \alpha d \in X} f_i(x^i + \alpha d)
\]  

(5.4)

However, the subproblem is not possible to solve exactly. Iterative procedure such as Backtracking Line Search can be used to find an \( \alpha \) that sufficiently decreases \( f_i(x) \). Define:

\[
\alpha_k = \begin{cases} 
\frac{x_k, \text{UpperBound} - x_k}{d_k} & \text{if } d_k > 0 \\
\frac{x_k, \text{LowerBound} - x_k}{d_k} & \text{if } d_k < 0 \\
0 & \text{otherwise}
\end{cases}
\]  

(5.5)

The maximum step size is:

\[
\alpha_{\text{max}} = \min\{\alpha_1, \alpha_2, ..., \alpha_n\}
\]  

(5.6)

**Backtracking Line Search** starts with half of maximum step size and decreases successively until objective function reduction is obtained [5]:

**Algorithm 5.1.** Backtracking Line Search

\[
\alpha_{\text{max}} \leftarrow \text{maximum step size;}
\alpha \leftarrow \alpha_{\text{max}}; \quad \text{counter} \leftarrow 0; \quad x \leftarrow x^i;
\text{maxiter} \leftarrow \text{maximal number of iterations;}
\text{while } \text{counter} < \text{maxiter and } f_i(x) \geq f_i(x^i) \quad \text{counter} \leftarrow \text{counter} + 1;
\alpha \leftarrow \alpha/2;
\text{end}
\]

Since the Steepest Descent algorithm minimizes \( f_i(x) \) according to its (approximated) gradient, it converges to a local minimum if the gradient is 0. In this case, the **Algorithm 3.1** could be applied in order to improve the solution, where the cost function (3.6) is replaced by \( f_i(x) \). Besides, the non dominated solutions of Steepest Descent algorithm and the intermediate solution candidates generated by **Algorithm 3.1** will be put together and filtered in the end of the algorithm.
5. Modified Steepest Descent Algorithm

Algorithm 5.2. Modified Steepest Descent

\[\text{maxiter} \leftarrow \text{maximum iterations}; \quad \epsilon \leftarrow 0.00001;\]
\[x \leftarrow \text{Initial Point}; \quad Z \leftarrow x;\]
\[\text{for } i = 1, 2, \ldots, m\]
\[\quad \text{counter} \leftarrow 0;\]
\[\quad \text{while } \text{counter} < \text{maxiter}\]
\[\quad \quad \text{counter} \leftarrow \text{counter} + 1;\]
\[\quad \quad \text{Compute } \nabla f_i(x);\]
\[\quad \quad \text{if } \left\| \nabla f_i(x) \right\| < \epsilon\]
\[\quad \quad \quad (x, W) \leftarrow \text{minimize } f_i(x) \text{ by Algorithm 3.1};\]
\[\quad \quad \quad (x \text{ is the solution and } W \text{ is the set of non dominated candidates})\]
\[\quad \quad \quad Z \leftarrow W \cup Z;\]
\[\quad \quad \text{else}\]
\[\quad \quad \quad \text{Compute } d \text{ and } \alpha_{\text{max}};\]
\[\quad \quad \quad \text{Find } \alpha \text{ by Algorithm 5.1};\]
\[\quad \quad \quad x \leftarrow x + \alpha d; \quad Z \leftarrow x \cup Z;\]
\[\quad \text{end}\]
\[\text{end}\]
\[X \leftarrow \text{prtopt}(Z);\]

Figure 17 shows how the solutions evolve during the algorithm procedure:

![Figure 17: Evolution of Solutions in Modified Steepest Descent Algorithm](image-url)
6. Genetic Algorithm for Multi-Objective Optimization

One way of solving multi-objective optimization problems is to apply the genetic algorithm, which is a search heuristic that mimics the process of natural evolution. This section will give an introduction to Multi-Objective Genetic Algorithm.

6.1. Methodology

A genetic algorithm starts with a population of randomly generated individuals (feasible solutions). In each generation, the fitness (some kind of one-dimensional objective function) of every individual in the population is evaluated, multiple individuals are stochastically selected from the current population based on their fitness, and modified (recombined and possibly randomly mutated) to form a new population (offspring, see Figure 18). The new population is then used in the next iteration of the algorithm. The algorithm terminates when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached for the population. If the algorithm has terminated due to a maximum number of generations, a satisfactory solution may or may not have been reached. Therefore, the genetic algorithm theoretically does not guarantee (Pareto) optimal solution(s).

![Crossover and Mutate Diagram](image)

Figure 18: Offspring generated by Crossover or Mutation

6.2. Algorithm Procedure

There exists a genetic algorithm solver in Matlab 2012a for solving multi-objective optimization problems. The algorithm procedure is presented briefly in this subsection:

**Algorithm 6.1. Genetic Algorithm**

**Step 1**: Create the initial population of individuals;
**Step 2**: Evaluate the fitness of each individual;
**Step 3**: while time ≤ timelimit and fitness is not sufficient, do
  **Step 3.1**: Select the best-fit individuals for reproduction;
  **Step 3.2**: Give birth to offspring by crossover and mutation;
  **Step 3.3**: Evaluate the offspring;
  **Step 3.3**: Replace the least-fit population with new individuals;
Note that there are many ways to define the best-fit individuals for multi-objective optimization problems. For example, one can choose the individual with minimum sum of objectives. In order to obtain a satisfactory solutions, the algorithm may need to be run for long time and a large number of individuals will be evaluated in the process, this is computationally expensive for complex problems.
7. Performance Evaluations of Algorithms

Given two multi-objective optimization algorithms $A_1$ and $A_2$, with corresponding non dominated solution sets $S_1$ and $S_2$. Define the following relations:

1. $A_1$ has better performance than $A_2$ ($A_1 \succ A_2$) if $\forall s^2 \in S_2, \exists s^1 \in S_1$ that dominates $s^2$.

2. $A_1$ is equally good as $A_2$ ($A_1 \approx A_2$) if $\exists s^1 \in S_1$ such that no solution in $S_2$ can dominate $s^1$ and $\exists s^2 \in S_2$, no solution in $S_1$ can dominate $s^2$.

3. Let $A_3$ be an algorithm. We say the performance of $A_1$ is no worse than $A_3$ ($A_1 \succeq A_3$) if $A_1 \succ A_2$, $A_1 \approx A_3$ and $A_2 \approx A_3$

290 Objective Function Evaluations:

![Figure 19: Solutions Comparison - Roughly 290 Objective Function Evaluations](image)

Since the task is to maximize both 10 percentile downlink and uplink user throughputs, this implies the more outward the Pareto Front is in the figure, the better performance the algorithm has.
From Figure 19, it can be seen that:

\[
\begin{align*}
SM & \approx MGA & SM & \succ MSD & SM & \succ RS \\
MGA & \approx MSD & MGA & \succ RS & MSD & \approx RS
\end{align*}
\]

This implies:

\[
SM \succeq MGA, \quad SM \succ MSD \text{ and } SM \succ RS \tag{7.1}
\]

**Stepwise Minimization** has better performance than other algorithms when there are 290 objective function evaluations and it finds the greatest 10 percentile down-link and uplink user throughput.

860 Objective Function Evaluations:

![Graph showing performance evaluations](image)

Figure 20: Solutions Comparison - Roughly 860 Objective Function Evaluations
This gives:

\[
\begin{align*}
\text{SM} & \succeq \text{MGA}, \quad \text{SM} \succeq \text{MSD} \quad \text{and} \quad \text{SM} \succ \text{RS} \\
\end{align*}
\]  \hspace{1cm} (7.2)

**Stepwise Minimization** still has better performance than other algorithms concerning 860 objective function evaluations. Besides, Figure 20 shows that the Pareto front of **Stepwise Minimization Algorithm** spreads more evenly than other algorithms which would be helpful for decision maker to select a solution.

**1550 Objective Function Evaluations:**

![Figure 21: Solutions Comparison - Roughly 1550 Objective Function Evaluations](image)

In this case, we have:

\[
\begin{align*}
\text{MGA} & \succ \text{SM} \quad \text{MGA} \approx \text{MSD} \quad \text{MGA} \succ \text{RS} \\
\text{MSD} & \succ \text{SM} \quad \text{MSD} \succ \text{RS} \quad \text{SM} \succ \text{RS} \\
\end{align*}
\]
This gives:

\[ \text{MSD} \approx \text{MGA} \succ \text{SM} \succ \text{RS} \quad (7.3) \]

Figure 21 implies that with 1550 objective function evaluations, the performances of Multi-Objective Genetic Algorithm and Modified Steepest Descent Algorithm surpass the others and they are considered to be equally good (MGA is better at downlink user throughput while MSD can find greater uplink user throughput), then followed by Stepwise Minimization Algorithm. Random Search Algorithm has the worst performance.

2070 Objective Function Evaluations:

![Graph](image)

Figure 22: Solutions Comparison - Roughly 2070 Objective Function Evaluations

From Figure 22, we can obtain:

\[ \text{MGA} \approx \text{SM} \quad \text{MGA} \approx \text{MSD} \quad \text{MGA} \succ \text{RS} \]

\[ \text{MSD} \approx \text{SM} \quad \text{MSD} \succ \text{RS} \quad \text{SM} \succ \text{RS} \]
That is:

$$\text{MSD} \approx \text{MGA} \approx \text{SM} \succ \text{RS}$$  \hspace{1cm} (7.4)

The Stepwise Minimization, Multi-Objective Genetic Algorithm and Modified Steepest Descent have equally good performances, while Random Search has the worst performance. Both Stepwise Minimization and Modified Steepest Descent have more evenly spread Pareto front so that it would be convenient for decision maker to make the final decision.

2550 Objective Function Evaluations:

Figure 23: Solutions Comparison - Roughly 2550 Objective Function Evaluations

Figure 23 implies:

$$\text{MGA} \succ \text{SM} \quad \text{MGA} \approx \text{MSD} \quad \text{MGA} \succ \text{RS}$$
$$\text{MSD} \succ \text{SM} \quad \text{MSD} \succ \text{RS} \quad \text{SM} \succ \text{RS}$$
Thus:

\[
MSD \approx MGA > SM > RS \quad (7.5)
\]

With 2550 objective function evaluations, both Multi-Objective Genetic Algorithm and Modified Steepest Descent Algorithm have better performance than the others and they are equally good (MGA is better on downlink and MSD is better on uplink). Random Search Algorithm has the worst performance.

5600 Objective Function Evaluations:

![Figure 24: Solutions Comparison - Roughly 5600 Objective Function Evaluations](image)

Similarly, Figure 24 implies:

- \( MGA \approx SM \approx MSD > RS \)
- \( MSD > SM > RS > SM \)

Therefore:

\[
MSD > MGA \approx SM > RS \quad (7.6)
\]
Modified Steepest Descent Algorithm has better performance than other algorithms in this case, then followed by Multi-Objective Genetic Algorithm and Stepwise Minimization Algorithm. Random Search Algorithm still has the worst performance.
8. Discussion

The solution candidates are normalized between 0 and 1, Figure 25 and 26 illustrate the actual solutions and parameters settings:

<table>
<thead>
<tr>
<th></th>
<th>Antenna 11</th>
<th>Antenna 24</th>
<th>Antenna 11</th>
<th>Antenna 24</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normalized Solution</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Actual Solution (deg)</td>
<td>0</td>
<td>0</td>
<td>-45</td>
<td>-45</td>
</tr>
</tbody>
</table>

**Figure 25: Initial Solution and Parameter Setting**

- **Initial Objective Values**
  - DownLink: 25.4132 Mb/s
  - UpLink: 4.7225 Mb/s

**New Objective Values**
- DownLink: 25.9825 Mb/s
- UpLink: 5.5590 Mb/s

<table>
<thead>
<tr>
<th></th>
<th>Antenna 11</th>
<th>Antenna 40</th>
<th>Antenna 24</th>
<th>Antenna 11</th>
<th>Antenna 40</th>
<th>Antenna 24</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normalized Solution</td>
<td>0.0200</td>
<td>0.0010</td>
<td>0.1603</td>
<td>0.9407</td>
<td>0.9120</td>
<td>0.6765</td>
</tr>
<tr>
<td>Actual Solution (deg)</td>
<td>0.3096</td>
<td>0.012</td>
<td>1.9236</td>
<td>39.663</td>
<td>37.08</td>
<td>15.885</td>
</tr>
</tbody>
</table>

**Figure 26: New Solution and Parameter Setting**

- **Horizontal Direction**
  - $\theta_1 = -45^\circ$
  - $\theta_2 = -45^\circ$
  - $\theta_3 = -45^\circ$
  - $\theta_4 = 39.663^\circ$
  - $\theta_5 = 37.08^\circ$
  - $\theta_6 = 0.012^\circ$
  - $\theta_7 = 1.9236^\circ$
  - $\theta_8 = 15.885^\circ$
This thesis proposes three different multi-objective optimization algorithms, \textbf{Stepwise Minimization Algorithm}, \textbf{Random Search Algorithm} and \textbf{Modified Steepest Descent Algorithm}. The decision maker is not involved in the solution process.

There are only 3 antennas (6 decision variables) to be optimized for purpose of illustration, the algorithms solve the optimization problem in the same way for more antennas (decision variables). However, it is recommended to optimize 2 to 5 antennas (or 4 to 10 decision variables). Otherwise it will take longer time to evaluate a solution candidate as the number of decision variables increase.

It can be seen from the investigations that good solution can be obtained on cost of objective function evaluations. Though the \textbf{Random Search Algorithm} has the worst performance, the idea is that, it is not always necessary to get a good solution but sometimes more favorable to find a satisfactory solution instead, say improving the current objective values simultaneously. For example, if the we start with Table 14:

<table>
<thead>
<tr>
<th>Initial Solution</th>
<th>Objtv (Mb/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Electrical Downtilt</td>
<td>Horizontal Direction</td>
</tr>
<tr>
<td>0.9000</td>
<td>0.9000</td>
</tr>
</tbody>
</table>

Table 14: Initial Solution and Objective Values

<table>
<thead>
<tr>
<th>Solution Candidate</th>
<th>Objtv (Mb/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Electrical Downtilt</td>
<td>Horizontal Direction</td>
</tr>
<tr>
<td>0.6068</td>
<td>0.0185</td>
</tr>
<tr>
<td>0.8913</td>
<td>0.4447</td>
</tr>
</tbody>
</table>

Table 15: Algorithm Result after 6 Objective Function Evaluations

Table 15 shows that both uplink and downlink user throughputs can be improved with only 6 objective function evaluations by \textbf{Random Search Algorithm}.

The \textbf{Stepwise Minimization Algorithm} have better performance than other algorithms with \( \leq 860 \) objective function evaluations and the non dominated solutions spread evenly in the feasible objective region.

From previous sections, we can see that the performance of \textbf{Modified Steepest Descent Algorithm} is not very good for few objective functions evaluations (\( \leq 290 \)), this due to the fact that not enough function evaluations to find the minimum with respect to one objective function. As the objective function evaluations increased, it will surpass \textbf{Random Search Algorithm} and \textbf{Stepwise Minimization Algorithm}. Generally speaking, \textbf{Modified Steepest Descent Algorithm} and \textbf{Multi-Objective Genetic Algorithm} have equally good performances according to the investigations so far. Besides, as the number of objective function evaluations increase, the Pareto Fronts of both algorithms spread more evenly which facilitate the decision maker to make decision.
In Stepwise Minimization Algorithm, it is possible to make the algorithm start to optimize from the first decision variable to the last decision variable again after finding the solutions to all decision variables. And the Modified Steepest Descent Algorithm can also be changed to optimize each objective function as many times as possible. But they are not considered for simplicity reason.
9. Conclusion

First of all, it is not recommended to draw too strong conclusions due to the fact that in this thesis work, the prepared algorithms have only been applied to a limited number of deployment and traffic scenarios and a selected set of antenna parameters. Moreover, the task is just to find the non dominated solution candidates as many as possible for decision makers. In practice, a final decision should be made by decision maker, that is, only one solution will be selected among all the non dominated solution candidates.

With the particular antenna parameters (Electrical Downtilt and Horizontal Direction of Antenna 11, 24 and 40), it can be concluded that Stepwise Minimization Algorithm is more advisable if the computation time is crucial. If the improvement of objective values is of great importance, either Modified Steepest Descent or Multi-Objective Genetic Algorithm is more preferable.

It is suggested that given a multi-objective optimization problem, one can first try Random Search Algorithm for a short period of time. If no improvement is obtained, just go to Stepwise Minimization Algorithm, Modified Steepest Descent or Multi-Objective Genetic Algorithm. The alternative way is to start with Random Search Algorithm to find a good initial solution, then continue with Stepwise Minimization Algorithm or Modified Steepest Descent Algorithm.
Appendix A: Graphical User Interface

A graphical user interface (GUI) has been developed to facilitate and solve antenna tuning problem. Users can set different network parameters and choose different algorithms to solve the antenna parameter optimization problem. There are functions for adding/deleting traffic hot zones, disabling/activating some certain antennas and adding/removing base stations. This section will give an overview of the GUI. First of all, take a look at the GUI before starting the simulation (Figure 27):

![Graphical User Interface Initialization](image)

Figure 27: Graphical User Interface Initialization

Some basic parameters need be set to define the optimization problem, Location, Number of (Probing) Users, User Throughput Percentile and Number of Antennas to be tuned in the simulation:

![Setting Network Parameters](image)

Figure 28: Setting Network Parameters
In order to solve the antenna parameters optimization problem, an algorithm needs to be selected. As can be seen from Figure 29, the default algorithm in the GUI is Random Search Algorithm, the other alternatives are Stepwise Minimization Algorithm, Modified Steepest Descent Algorithm and Multi-Objective Genetic Algorithm:

![Select Algorithm](image)

**Figure 29: Selecting Optimization Algorithm**

After that, the user simply clicks the Start button. The algorithm will now search for non-dominated solutions along the way. Figure 30 shows the result when an algorithm has terminated:

![Graphical User Interface after Algorithm Terminates](image)

**Figure 30: Graphical User Interface after Algorithm Terminates**
Once the algorithm finishes, one can modify the scenario, e.g., add two traffic hot zones (Figure 31) in the current network and disable two antennas (Figure 32):

![Figure 31: Adding Traffic Hot Zones](image)

The two numbers with multiple circles in Figure 31 are the new added traffic hot zones and the antennas with red marks in Figure 32 are disabled. Now the network map is different from before, users need to restart the algorithm to get a new solution for the new situation. Whenever the traffic or network has changed, one always needs to restart the algorithm so that the GUI can updates the network data and solve the new optimization problem.

![Figure 32: Disabling Antenna](image)
One can also add new base stations to the network, Figures 33 and 34 shows the network map before and after adding 2 base stations respectively, Figure 35 gives the location of the two new base stations:

![Figure 33: Original Network Map](image1)

![Figure 34: Network Map with 2 New Base Stations](image2)

![Figure 35: Locations of 2 New Base Stations](image3)
### Appendix B: Simulation Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Indoor Users Percentage</td>
<td>80%</td>
</tr>
<tr>
<td>Outdoor Users Percentage</td>
<td>20%</td>
</tr>
<tr>
<td>Number of Base Station</td>
<td>16</td>
</tr>
<tr>
<td>Sectors per Site</td>
<td>1-3</td>
</tr>
<tr>
<td>Total Resource Blocks</td>
<td>20MHz</td>
</tr>
<tr>
<td>Path Loss Model</td>
<td>BEZT</td>
</tr>
<tr>
<td>Max Transmit Power, UpLink</td>
<td>0.2W</td>
</tr>
<tr>
<td>Antenna Model</td>
<td>hv 742215 fitted</td>
</tr>
<tr>
<td>Antenna Height, User Equipment</td>
<td>Ground/Every Floor + 1.5m</td>
</tr>
<tr>
<td>Antenna Gain, User Equipment</td>
<td>Omni Directional</td>
</tr>
<tr>
<td>Max Transmit Power, DownLink</td>
<td>60W</td>
</tr>
<tr>
<td>Antenna Height, Base Station</td>
<td>2.5m above roof top</td>
</tr>
<tr>
<td>Max Antenna Gain, Base Station</td>
<td>18dbi</td>
</tr>
</tbody>
</table>

Table 16: Simulation Parameters
Appendix C: Stepwise Minimization Algorithm

For the purpose of illustration, we consider a simplified example, only 3 antennas of the network are tuned (Antenna 11, 40 and 24), each antenna has two decision variables, Electrical Downtilt and Horizontal Direction. Horizontal BeamWidth and Vertical BeamWidth are not considered. The number of solution candidates is 10 and all the decision variables are normalized between 0 and 1. Table 17 shows the initial solution and objective values (Objtv - Objective Values, DL - DownLink User Throughput, UL - UpLink User Throughput, A - Antenna).

<table>
<thead>
<tr>
<th>Initial Solution (x^0)</th>
<th>Objtv (Mb/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Electrical Downtilt</td>
<td>Horizontal Direction</td>
</tr>
<tr>
<td>A 11</td>
<td>A 40</td>
</tr>
<tr>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>25.4132</td>
<td>4.7225</td>
</tr>
</tbody>
</table>

Table 17: Stepwise Minimization Algorithm Initialization

<table>
<thead>
<tr>
<th>Candidates</th>
<th>Objtv (Mb/s)</th>
<th>Max (Mb/s)</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>E-Tilt</td>
<td>H-Direction</td>
<td>DL</td>
<td>UL</td>
</tr>
<tr>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>0.8756</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0.4392</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0.5265</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0.5201</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0.3512</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0.4749</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0.2855</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0.2318</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0.4792</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 18: Stepwise Minimization Algorithm Step 1

<table>
<thead>
<tr>
<th>Non Dominated Candidates</th>
<th>Objtv (Mb/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>E-Tilt</td>
<td>H-Direction</td>
</tr>
<tr>
<td>0.0000</td>
<td>0.0000</td>
</tr>
</tbody>
</table>

Table 19: Non Dominated Solutions and Objective Values after Step 1
### Table 20: Stepwise Minimization Algorithm Step 2

<table>
<thead>
<tr>
<th>Candidates</th>
<th>Objtv (Mb/s)</th>
<th>Max (Mb/s)</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>E-Tilt</td>
<td>H-Direction</td>
<td>DL</td>
<td>UL</td>
</tr>
<tr>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>0</td>
<td>0.9776</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0.4912</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0.8627</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0.3059</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0.2758</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0.0654</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0.2112</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0.1233</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0.0709</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

### Table 21: Non Dominated Solutions and Objective Values after Step 2

<table>
<thead>
<tr>
<th>Non Dominated Candidates</th>
<th>Objtv (Mb/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>E-Tilt</td>
<td>H-Direction</td>
</tr>
<tr>
<td>0.0000</td>
<td>0.0000</td>
</tr>
</tbody>
</table>

### Table 22: Stepwise Minimization Algorithm Step 3

<table>
<thead>
<tr>
<th>Candidates</th>
<th>Objtv (Mb/s)</th>
<th>Max (Mb/s)</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>E-Tilt</td>
<td>H-Direction</td>
<td>DL</td>
<td>UL</td>
</tr>
<tr>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>0</td>
<td>0.7272</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0.5129</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0.6309</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0.0529</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0.9861</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0.7432</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0.4726</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0.8652</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0.0339</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

### Table 23: Non Dominated Solutions and Objective Values after Step 3
<table>
<thead>
<tr>
<th>Candidates</th>
<th>Objtv (Mb/s)</th>
<th>Max (Mb/s)</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>E-Tilt</td>
<td>H-Direction</td>
<td>DL</td>
<td>UL</td>
</tr>
<tr>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0529</td>
<td>0.2747</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0.0529</td>
<td>0.6193</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0.0529</td>
<td>0.2681</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0.0529</td>
<td>0.1462</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0.0529</td>
<td>0.0909</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0.0529</td>
<td>0.1937</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0.0529</td>
<td>0.6631</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0.0529</td>
<td>0.6916</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0.0529</td>
<td>0.8148</td>
</tr>
</tbody>
</table>

Table 24: Stepwise Minimization Algorithm Step 4

<table>
<thead>
<tr>
<th>Non Dominated Candidates</th>
<th>Objtv (Mb/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>E-Tilt</td>
<td>H-Direction</td>
</tr>
<tr>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>0.0529</td>
<td>0.6193</td>
</tr>
<tr>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>25.6507</td>
<td>4.8476</td>
</tr>
</tbody>
</table>

Table 25: Non Dominated Solutions and Objective Values after Step 4

<table>
<thead>
<tr>
<th>Candidates</th>
<th>Objtv (Mb/s)</th>
<th>Max (Mb/s)</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>E-Tilt</td>
<td>H-Direction</td>
<td>DL</td>
<td>UL</td>
</tr>
<tr>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0529</td>
<td>0.6631</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0.0529</td>
<td>0.6631</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0.0529</td>
<td>0.6631</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0.0529</td>
<td>0.6631</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0.0529</td>
<td>0.6631</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0.0529</td>
<td>0.6631</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0.0529</td>
<td>0.6631</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0.0529</td>
<td>0.6631</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0.0529</td>
<td>0.6631</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0.0529</td>
<td>0.6631</td>
</tr>
</tbody>
</table>

Table 26: Stepwise Minimization Algorithm Step 5
### Appendix 51

#### Non Dominated Candidates

<table>
<thead>
<tr>
<th>E-Tilt</th>
<th>H-Direction</th>
<th>Objtv (Mb/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0529</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0.6631</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0.3967</td>
</tr>
</tbody>
</table>

#### Table 27: Non Dominated Solutions and Objective Values after Step 5

<table>
<thead>
<tr>
<th>Candidates</th>
<th>Objtv (Mb/s)</th>
<th>Max (Mb/s)</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>E-Tilt</td>
<td>H-Direction</td>
<td>DL</td>
<td>UL</td>
</tr>
<tr>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0529</td>
<td>0.6631</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0.0529</td>
<td>0.6631</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0.0529</td>
<td>0.6631</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0.0529</td>
<td>0.6631</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0.0529</td>
<td>0.6631</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0.0529</td>
<td>0.6631</td>
</tr>
</tbody>
</table>

#### Table 28: Stepwise Minimization Algorithm Step 6

<table>
<thead>
<tr>
<th>E-Tilt</th>
<th>H-Direction</th>
<th>Objtv (Mb/s)</th>
<th>Max (Mb/s)</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0529</td>
<td>0.6193</td>
<td>0.0000</td>
</tr>
</tbody>
</table>

#### Table 29: Non Dominated Solutions and Objective Values after Step 6
References


