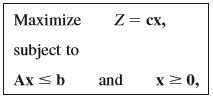
**Operations Draft**

**Short intro**

The following report concerns the description and analysis of the application of a linear model to optimize scheduling for stand and gate assignment at the Kenyatta airport, based on the data present at “Robust Scheduling for the Bay and Gate Assignment” by Jakko Deken. Throughout the report various scenarios, from simpler cases consisting on few aircrafts to a small number of gates and stands to more complex ones will be considered, testing the model’s accuracy on each of them and matching optimized solutions to predictions. Additionally, sensitivity analysis will be conducted on each situation considered, together with impacts of model complexity on computational time.

**Model description And Software used for simple examples**

Being considered as a linear optimization problem, the simplex method was used, recurring to the CPlex toolbox on MATLAB, importing data from csv excel files. Consequently, alike any other problem using the simplex method, it follows the following general format in its matrix formulation:



**Variables Xi**

In the implemented model, a Binary Integer Programming (BIP) approach was taken, in which every decision variable xijk where i stands for flight number, j for the stand used and k for the corresponding gate number. Thus, having Xijk = 1 means flight i should be directed to stand j and the passengers should then be moved to gate k.

**A and B Matrices**

Regarding the implementation of the A and B matrices, four major kinds of constraints were implemented: the first block of constraints, corresponding to the first I lines of the A matrix, are summarized below:

Write equation on LaTeX (for every i, the sum for all j and k of all variables must be equal to 1).

Which imposes that each flight must be assigned one and one only stand and gate combination.

The second major block of constraints in the A matrix is relative concurrent flights i.e. flights which will be at the airport simultaneously. In order to implement these restrictions, the algorithm developed, when reading the flight schedule from a csv excel file, identifies concurrent flights, grouping them in set It (times concurrent at time slot t). Using this It set, the constraints summed below make sure that concurring flights are not assigned to the same gate (2a)) nor the same Stand (2b).

Equation 2) a)

Equation 2) b)

The third block of constraints is relative to domestic and international flights. Once again, in the code developed, this information is accessible in the csv excel file which enables all domestic flights to be grouped together in one set, I\_dom, and all international flights to be grouped in I\_int. Thus, the following restraints impose that no international flight is to land on a domestic gate and vice-versa:

3)

Lastly, the fourth block of constraints is relative to aircraft type. In the Model developed, 17 models of aircrafts are considered (SHOW LIST SOMEWHERE IN THE ANNEX), making it so that for the bigger models, only a few stands can be selected. The aircraft type can be viewed in the flight schedule, while the models each stand can accommodate are defined in the *stand\_limit* file. Consequently this constraint can be written down as

4)

Where J\_ati represents the set of stands in which the aircraft from flight I can be allocated to and J the complete set of stands

**Objective Function**

Regarding the choice of the objective function, Z, in the model at hand, two different implementations were considered. In the first one, used in the first and simplest example, the objective function consisted on minimizing the total distance from stand to gate travelled by the passengers. This simpler version of the objective function, mathematically summarized below, was conceived in order to test the functionality and optimality of the solution obtained in easy to predict scenarios, associated with a small number of incoming flights, gates, stands. Concerning the costs associated with each decision variable, cijk, arbitrary values were used (INCLUDE A TABLE WITH THE COST COEFFICIENTS IN THE FIRST EXAMPLE)

When analysing more complex problems, with a significant increase in the number of variables due to the high number of incoming flights, stands and gates, four different solutions relative to four different objective functions are considered.

The first solution obtained is relative to an objective function analogue to the one used in simpler examples: minimizing the total distance travelled by all flights between the assigned stands and gates. However, since minimizing this specific distance may not always be the main priority in gate assignment, three other scenarios were considered.

In the first alternative scenario the objective is set to minimize the total distance (travelled by all flights) from the runway to the assigned gate. Due to the inexistence of accurate data regarding the distance from the runway to each of the stands, once again, arbitrary values were selected. On the second alternative scenario, the objective function consists on minimizing the total distance travelled by passengers from the assigned gate to the airport exit. This objective function, unlike the previous ones, takes into account not only the distance from each gate to the airport exit but also the number of passengers on each flight.

Lastly, the fourth and final objective function considers all three distances mentioned above, recurring to normalization and adding weight factors in order to establish the degree of preference in minimizing each one of the three distances. Its general template is summed below:

Unshit yourself, Tommy

Thus, in each example considered, four possible solutions will be presented, each corresponding to one of the objective functions mentioned.

**Example 1**

In the first, simplest, example, the model was tested to land four flights, with three available gates and three gates.

The schedule for the flights and the distance between each gate and terminal is presented in the following tables: