

# Empirical Modelling, Linear programming and optimization

1160398

## Abstract

This paper discusses the use of Empirical Modelling in Linear programming and optimization. We will go through the design of this model and look at how the methods link the variables to be a model together with the advantages and disadvantages of this model. Finally, we will conclude with the problems faced during programming and the further work could be done in both this particular model and the EM.

## 1. Model Design

### 1.1 Display

The display is designed as a dynamic board with X and Y coordinates, and the display will update when user made any changes in the constraints or goals, and some simple functions, for example enlarge and reduce, is also provide to make a better understanding of the display.

### 1.2 User interface

The user interface is designed to let the user can easily modify any variables in constraints and the goal, the model will update the changes and automatically find the best result if exists. And the user will have some options such as turn on/off the shadow to show the available part of each constraint, get a simple sample or restart the model (back to the beginning with nothing).

### 1.3 Algorithm

To solve a general 2D linear optimization program, we know that the best result can only be found on the cross point of two constraints or the line between two cross points if there exists any best result. So I just focus on all the cross points. Every cross point will be given an result by function and compare by each other to find a best result (the precondition will check if the point matches all the constraints). And then the result will be check by some other possible points that also match all the constraints to check if the previous result is the best answer or the best answer is infinity.

## 2. Model Description

### 2.1 Layout

The layout of the whole model is using %angel, which compare with using simple points and positions in %donald it have a easier way to make sure the relationship with components. Moreover, the button in %angel looks much better and every component is easier to create and make changes, as well as the mouseclick action.

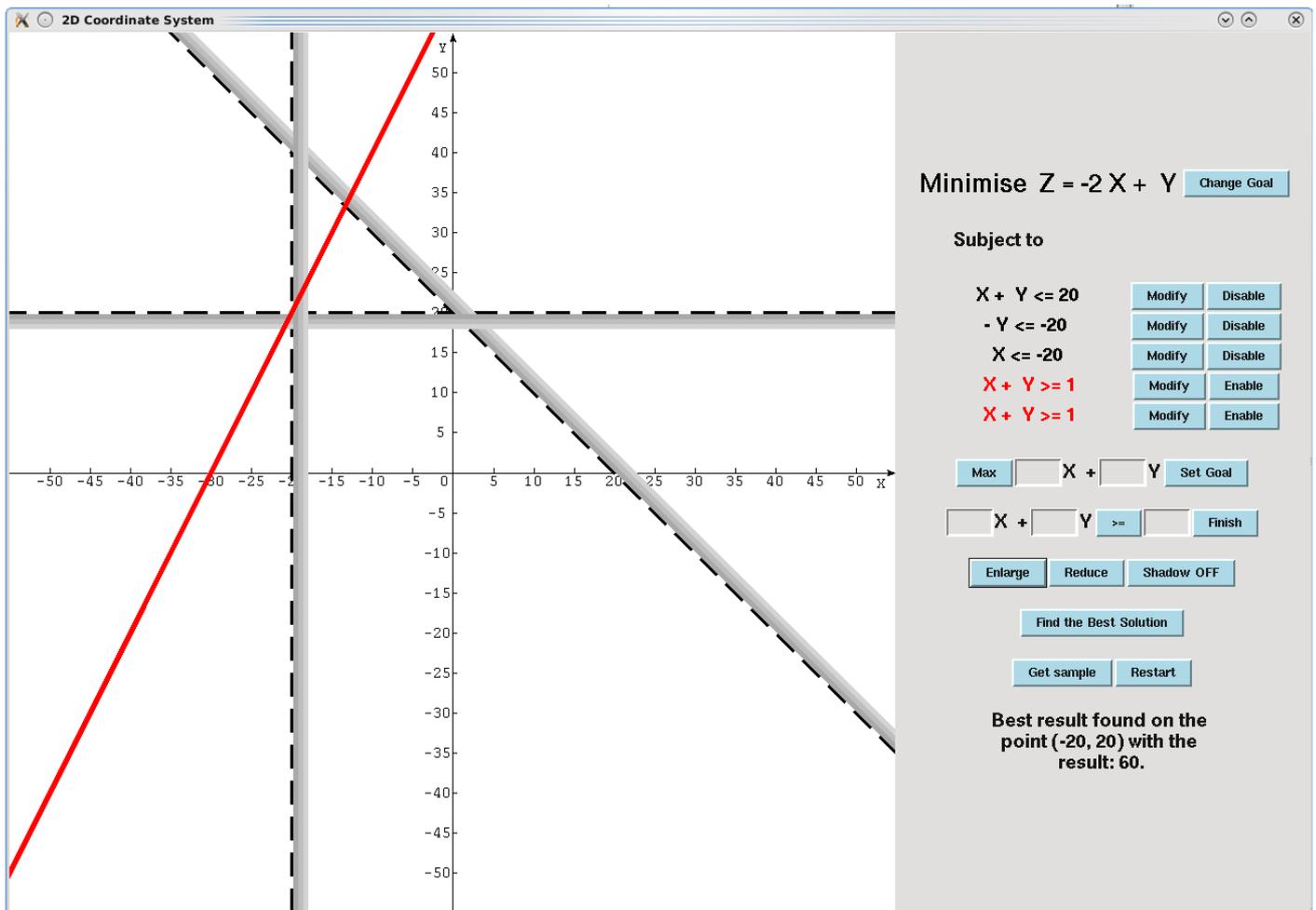
In my layout, the left part is a big coordinate system and the right part is the user interface. They are using the same variables to make sure they will update the information together.

### 2.2 Coordinate System

The coordinate system in the left is simply a picture built in %scout and the whole picture is added to the layout. The coordinate system is built with a basic point, which is also the zero point on the display, all other points will change is the zero point is changed. Unit length is signed out the let the user can easily find the position of each point. The unit length is dynamic and the user can change it and all the points and constraints will update itself to match the new coordinate system. For each constraint, there are three shadow lines mark the part that doesn't match this constraint (it can be turned off).

### 2.3 User interface

On the top of the user interface, the goal and all the constraints is showed to the user and if they want to make any changes, just click the button on the right side so that the information of this goal/constraint will be copied to the text box below and the user can change any variables then click the finish button to let the model



to update the information. There are also other buttons to let the user easily choose to enable/disable any constraint and the coordinate system on the left will update it immediately. Able/disable/modified constraint is marked with different color to let the user can easily know what they are doing. Some simple sample is also provide to the user and generally a restart button will delete all the input information. The final result will be show on bottom with the best result (if exists) together with when the result can be found.

### 3. Problems Faced

#### 3.1 Layout

The layout in %angel is better than what in %donald, but I still have some problems while using them.

The position of each part should be very clear and simple, which means I can't finish all the components in one level, they should be designed into different levels and make sure in each level there doesn't have too many directions, or some components may be missing.

By the way, if the whole board is empty and I create some component with "top" position, it still appear in the middle not the top.

### 3.2 Complexity

As I don't know how to make changes to points and lines in %eden until I have finish most of the model, so that all the points is stored with two real number for each of them, for example x and y. And the function to link the model is updating them separately not together, which means I will have one function to update x and another for y with the same input information. This problem becomes serious when I want to make a maximum of 10 constraints for my model. There are too many links with the functions and variables the tkeden gives me an overflow error so I have to reduce the constraints to 5.

### 4. Advantages/Disadvantages

#### 4.1 Advantages

The user interface is very easy to understand and use. The user can learn how the linear programming optimization is doing when constraints and goal changes, by different input information, they will learn the different situation that the best result could be found or not.

## 4.2 Disadvantages

The maximum of 5 constraints are too few, and it's very difficult to make progress because they are not stored as list, each of them has a variety of variables.

The algorithm is all about the cross points, but there are some situation that no cross point exists, for example no constraint ,only one constraint exists or all the constraints have the same slope.

## 5. Further Working

By using the list, the complexity of the model can be reduced easily, so that more constraints can be added to this model.

But the complexity of this algorithm will increase rapidly when the number of constraints increases. Because the number of cross points. When there are 5 constraints, only 10 cross points exist, for 10 constraints, it increases to 45, and 120 cross points for 15 constraints. So the algorithm itself should be changed.

Other thing could be done is to develop this model into a 3D model, but it is difficult to show each constraint (as I use shadow in 2D model), and the calculate function is also difficult to work out.

## 6. Reference

1. <http://empublic.dcs.warwick.ac.uk/projects/bubblesortBeynon1998>
2. <http://empublic.dcs.warwick.ac.uk/projects/empeHarfield2007>
3. [syminfoBeynon2011](http://empublic.dcs.warwick.ac.uk/projects/syminfoBeynon2011)