Introduction (Programming)

Operations research is a scientific approach to decision making that seeks to best design and operate a system, under conditions requiring the allocation of scarce resources. The scientific approach to decision making usually involves the use of one or more mathematical models.

If, whenever decision variables appear in the objective function and in the constraints of an optimisation model, the decision variables are always multiplied by constants and added together, such a model is a ***linear model***.

If one or more decision variables must be integer, then we say that an optimisation model is a ***discrete model*** or an ***integer model***.

Outline

Tip: do the assignment first.

For the project, create a program that solves Linear Programming and Integer Programming Models and then analyses how the changes in an LP's parameters change the optimal solution.

The source code must be a visual studio project. **Any** .NET programming language may be used. The project should build an executable (solve.exe) that is menu driven with the following:

The program should be able to accept an input text file with the mathematical model and export all results to an output text file.

Minimum Requirements Criteria

- Program should accept a random amount of decision variables.
- Program should accept a random amount of constraints.
- Use comments with programming.
- Programming Best Practices should be implemented.

Input Text File Criteria

The first line contains the following, seperated by spaces:

- The word max or min, to indicate whether it is a maximization or a minimization problem.
- For each decision variable, a operator to represent wheter the objective function coefficient is a negative or positive.
- For each decision variable, a number to represent its objective function coefficient.

A line for each constraint:

- The operator of the technological coefficients for the decision variables, in the same order as in the specification of the objective function in line 1, that represents whether the technological coefficient is negative or positive.
- The technological coefficients for the decision variables, in the same order as in the specification of the objective function in line 1.

- The relation used in the constraint, with =,<=, or >=, to indicate respectively, an inequality to constraint the constaint right-hand-side.
- The right-hand-side of the constraint.

Sign Restrictions

- Sign restriction to be below all the constraints, seperated by a space, +, -, urs, int, bin, in the same order as in the specification of the objective function in line 1.

Note: The Linear Programming Model or the Integer Programming Model should be entered into the file. Not the canonical forms of the different algorithms respectivly or the Relaxed Linear Programming Model.

Example of Input File

max +2 +3 +3 +5 +2 +4

+11 +8 +6 +14 +10 +10 <=40

bin bin bin bin bin bin

Tip: Use the above IP in the video and then play around with the values to show all criteria.

Processing

Your program should provide you with the option to select which algorithm to use in order to solve the programming model.

Your program should provide you with options to perform sensitivy analysis operations after the programming model has been solved.

Programming Model Criteria

- Ability to solve normal max Linear Programming Models (specifically the given Knapsack IP).
- Ability to solve binary Integer Programming Models (specifically the given Knapsack IP).

Algorithms to be available

- Primal Simplex Algorithm **and** Revised Primal Simplex Algorithm.
- Branch and Bound Simplex Algorithm **or** Revised Branch and Bound Simplex Algorithm.
- Cutting Plane Algorithm **or** Revised Cutting Plane Algorithm.
- Branch and Bound Knapsack algorithm.

Algorithm Criteria

- Display the Canonical Form and solve using the Primal Simplex Algorithm. Display all tableua iterations.
- Display the Canonical Form and solve using the Revised Primal Simplex Algorithm. Display all Product Form and Price Out iterations.
- Display the Canonical Form and solve using the Branch & Bound Simplex Algorithm.
    - Backtracking should be implemented.
    - Program should create all possible sub-problems to branch on.
    - Program should fathom all possible nodes of sub-problems.
    - Display all the table iterations of the above mentioned sub-problems.
    - Best candidate must be displayed.
- Display the Canonical Form and solve using the Cutting Plane Algorithm. Display all Product Form and Price Out iterations.
- Solve using Branch and Bound Knapsack algorithm.

- Backtracking should be implemented.
- Program should create all possible sub-problems to branch on.
- Program should fathom all possible nodes of sub-problems.
- Display all the table iterations of the above mentioned sub-problems.
- Best candidate must be displayed.

Output file format

The output file should contain the Canonical form and all the tablue iterations of the algorithm that was selected to solve the Programming Model.

All decimal values should be rounded to three points.

Sensitivity Analysis Criteria

Your program should have the options to do the following sensitivity analysis operations:

- Display the range of a selected Non-Basic Variable.
- Apply and display a change of a selected Non-Basic Variable.
- Display the range of a selected Basic Variable.
- Apply and display a change of a selected Basic Variable.
- Display the range of a selected constraint right-hand-side value.
- Apply and display a change of a selected constraint right-hand-side value.
- Display the range of a selected variable in a Non-Basic Variable column.
- Apply and display a change of a selected variable in a Non-Basic Variable column.
- Add a new activity to an optimal solution.
- Add a new constraint to an optimal solution.
- Display the shadow prices.
- Duality:
    - Apply Duality to the programming model.
    - Solve the Dual Programming Model.
    - Verify whether the Programming Model has Strong or Weak Duality.

Special Case Requirements

Program should be able identify and resolve infeasible or unbounded programming models.

Bonus Criteria

The maximum mark is 100, but if you get less, this bonus mark can boost your final mark.

The program should have the ability to solve any non-linear problem like $f(x)=x^2$ with any algorithm, but you need to explain the code for this part. The rest of this assignment, you only need to show how everything works that meets all the criteria.

Video submission

You need to submit a video or Onedrive link in a PDF document to a video that will show how your program meets all the criteria. Marks will **only** be awarded for the video as we are your client and will not be marking your code. Even if you make a video just about your code, you will not get a lot

of marks because we want to see a working project after you explained your code. If there were issues with group members not contributing, put it in a PDF that you will upload with your video.

No time limit on the video, but make sure you show all the criteria as fast as possible.

Mark Allocation

Criteria	Weight
Content	
Outline	2
Input File	3
Output File	2
Primal Simplex Algorithm	4
Revised Primal Simplex Algorithm	4
Branch & Bound Simplex Algorithm or Revised Branch & Bound Simplex Algorithm	20
Branch & Bound Knapsack Algorithm	16
Cutting Plane Algorithm or Revised Cutting Plane Algorithm	14
Sensitivity Analysis	25
Error Handling (e.g. cannot solve with simplex chosen and special case requirements)	5
Interface presentation	5
Total	**100**
Non-linear problem solved	10

Additional Information

- All work must be done in your assigned group.
- Belgium Campus have software that can **scan for plagiarism** and a student caught doing this will get 0 for this assignment.
- Late assignments will not be accepted; missing the deadline is an automatic 0.