Readme

Maria Baldeon Calisto

This code applies the quadrant shrinking method to solve tri-objective pure integer linear programs. The code is written in Python 3.6 with CPLEX optimizer.

1 Input file

The tri-objective integer linear instance information has to be saved in a text file under the name of parameters.txt. An example of the file is added in the folder.
The format is described using the following example problem:

$$min\ Z_1 = 4x_1 + 5x_2 + 6x_3$$
$$min\ Z_2 = 5x_1 + 1x_2 + 1x_3$$
$$min\ Z_3 = 3x_1 + 2x_2 + 1x_3$$

subject to

$$x_1 + 2x_2 - 5x_3 \leq 5$$
$$x_2 - 6x_3 \leq 5$$
$$-x_1 + 3x_2 + 2x_3 = 2$$
$$x_1, x_2, x_3\ \epsilon\ Z_+$$

The input file should follow these directions:

	Number of variables	Number of Constraints	Number of Constrainst with inequality
	3	7	6
Coefficients of first OF	4	5	6
Coefficients of second OF	5	1	1
Coefficients of third OF	3	2	1
Coefficients of the constraints	1	2	-5
	0	1	-6
	-1	3	2
Right hand side of constraints	5	5	2

Figure 1: Example of parameters.txt

The first line includes information about the problem: number of variables, total number of constraints and the number of the constraints that are modeled as inequality. The subsequent three lines must include the coefficients of the three objective functions. The coefficients for the constraints must follow. The constraints must be organized such that the inequalities are located before the equality constraints. Finally, the right hand side of the constraints are located in the last line.

2 Output file

The output file is saved under the name problem_solutions.txt . An example of the report is located in the folder. It reports the efficient points found, their corresponding efficient solutions and the run time.
The results are straight forward to read. An example of output is shown below:

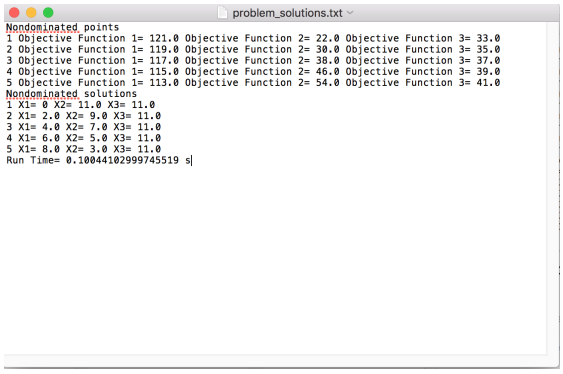

Figure 2: Example of problem_solutions.txt

3 How to run the program

The project can be compiled in the Terminal by writing: *python Homework3.py*
The input file and Homework3.py file must be located in the same folder. The output file will be saved in the same folder.

4 Description of the Functions

The code is composed of 7 functions:
read_file: reads the parameter.txt file.
set_parameters: sets all the parameters of the model defined by the user in dictionaries.
construct_model: constructs an object from the model class. Based on the parameters sent to the function it creates the constraints and decision variables.
solve_model: solves the model and returns the optimal solution and value of the objective function.
two_phase_search: obtains a non-dominated solution by first minimizing the 3rd objective function and then putting constraints over the values of all the objective functions and minimizing the sum of all the functions.
quadrant_shrinking:applies the quadrant shrinking method to obtain a list of efficient solutions and points.
write_output: writes the ouput text with the non dominated points, efficient solutions and execution time.

To implement the code [1] and [2] tutorials where used.

References

[1] Tutorial: Beyond linear programming (cplex part 2). [Online]. Available: https://github.com/IBMDecisionOptimization/tutorials/blob/master/jupyter/Beyond_Linear_Program

[2] Ibm decision optimization: Cplex modeling for python. [Online]. Available: https://rawgit.com/IBMDecisionOptimization/docplex-doc/master/docs/index.html