Convert Primal Problem to its Dual

Kanan Jafarli, Ibrahim Yunuslu

May 22, 2021

1 Converting Linear Problem to Dual

We say that it is linear programming problem.

$$maximizeC(x)c_1x_1 + ... + c_nx_n$$
$$\begin{cases} a_{11}x_1 + ... + a_{1n} \leq b_1 \\ . \\ . \\ . \\ a_{m1}x_1 + ... + a_{mn} \geq b_m \\ x_i \geq 0 \end{cases}$$

If you have constraints like below will be:

$$\alpha_1x_1 + ... + \alpha_nx_n \geq \beta \implies -\alpha_1x_1 - ... - \alpha_nx_n \leq -\beta$$

$$\alpha_1x_1 + ... + \alpha_nx_n = \beta \implies \begin{cases} \alpha_1x_1 + ... + \alpha_nx_n \leq \beta \\ -\alpha_1x_1 - ... - \alpha_nx_n \leq -\beta \end{cases}$$

The dual problem is obtained by the opposite of the transpose of the matrix of the canonical form of the primal problem. When in the primal problem object function is maximized, in dual it will be minimized.

2 Code explanation

The program first reads the LP.txt file (Primal problem) in list format. It is then edited accordingly in another list for better editing. There are 5 smaller methods that each performs its function:

- CheckFormation (newList) checks if the text file formatting is correct based on the pronunciation requirements. In more detail, it checks for the existence of min or max in the first String element of the list and st or subject to in the second line. It also checks the existence of the signs - and + with the logic that between the first x without its sign until the end of the arithmetic expression holds: number x - 1 = number of signs. This is because the sign of the first x can be omitted. Also, in the first line if there is the word max it also locates x from max so I find first the symbol = of the first row and then the position of the first x without its sign. In addition, it detects the existence of the symbol = in the technological constraints (which includes the existence of $\leq$ and = and $\geq$), if found then it finds its position = and checks if after that it finds a number.
- CalculateDimensions (newList) finds the dimensions of the tables and returns them.
- Add_One_or_minusOne (newList) finds the positions of the x in each line and checks if there is a character other than a number or - or + in the position before x and inserts 1 or -1 in front of x, respectively. If it is found in a position that 1 or -1 needs to be added, 1 or 2 must be added to the positions of the next x (for 1 it adds 1 and for -1 it adds 2 to the positions of the next x that have not been examined yet) so as not to our program checks wrong places.
- InsertIntoTables (newList, m, n) sets the parameters of x in the tables c for the objective function and A for the technological constraints, in the table Eqin are placed -1 or 1 or 0 if there is $\leq$ or $\geq$ or = respectively and in Table b is the numbers after the above symbols. For tables A, b, c specific patterns are used to find the numbers before x (coefficients), numbers after x and numbers after =. For tables A and c are the numbers before and after x and the numbers after x act as pointers to the column where the x parameter is to be entered.

- GetMinMax (newList) finds if the word min or max is in the first line and returns -1 or 1 respectively.

In main file with the help of the temp function I change the tables c, b between them and the contents of A. Thus, these tables contain the results of a binary table. Then in the text variable type String we try to do the appropriate formatting. The variable count is used to hold which y is y1 or y2 and so on. Since we know that all variables are ≥ 0 we follow the steps of the algorithm and in the constraints the symbols ≥ 0 or ≤ 0 appear depending on the cases min or max of the primary problem. Again, depending on the min or max cases of the primary problem, the constraints of the new variables y are either free or ≥ 0 or ≤ 0. Finally, the contents of the String type variable text are recorded in the DualP.txt file.