Assignment 1: Maximizing Your LA Grade

2021 Fall EECS205002 Linear Algebra

Due: 2020/10/20

Student A takes Linear Algebra course this semster. The course has 10 points for class participation, 40 points for programming assignment, and 50 points for exams. Student A participates the class enthusiastically, so he does not worry about the 10% class participation. But he only wants to spend 6 hours per week for studying LA. He found that he has the following facts.

- If he spends 1 hour per week to study and do exercises, he can get 10 points for the exam, and 2 points for the programming assignments.
- If he spends 1 hour per week in the programming, he can get 1 point for the exam, and 14 points for the programming assignments.

Now he wants to know how many hours he should study and how many hours he should work on the programming to maximize his LA grades.

Suppose he plans to spend x hours per week for studying LA, and y hours for LA programming assignments. The grade that he will get is

$$10 + (10x + y) + (2x + 14y),$$

where 10 points is for classparticipation, $(10x + y)$ is the grade for exam, and $(2x + 14y)$ is the grades for programming. But he has the following constraints.

$$\begin{aligned}
x + y \leq 6 &\ // \text{ the maximum effort for study LA.} \\
10x + y \leq 50 &\ // \text{ the maximum grade for exams is 50.} \\
2x + 14y \leq 40 &\ // \text{ the maximum grade for assignments is 40.} \\
x, y \geq 0 &\ // \text{ all hours should be nonnegative.}
\end{aligned}$$

The problem to maximize his grade is called *Linear Programming*, which is expressed as follows

$$\begin{aligned}
\max_{x,y} G(x, y) = & 12x + 15y + 10 \\
\text{subject to } & x + y \leq 6 \\
& 10x + y \leq 50 \\
& 2x + 14y \leq 40 \\
& x, y \geq 0
\end{aligned} \tag{1}$$

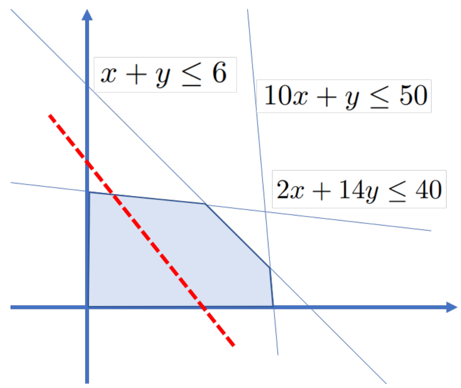

Figure 1: Constraints for the linear programming.

If you plot the inequalities in the x-y coordinate, you will get something looked like Figure 1. The theory of linear programming tells that the solution to this optimization problem must be at some intersection points of constraints.

A simple algorithm to solve the linear programming is summarize as follows

1. Enumerate all intersection points using the constraints.
2. Check the feasibility of all intersection points.
3. For each valid intersection point, compute its objective function G.
4. Find the maximum G, which is the solution.

For this problem, there are three constraints besides $x \geq 0$ and $y \geq 0$. To find an intersection point, we need two constraints. There are three constraints, so we have $C_2^3 = 3$ cases. For each case, we need to solve a linear system.

	Case 1	Case 2	Case 3
Equation	$\begin{bmatrix} 1 & 1 \\ 10 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 6 \\ 50 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 \\ 2 & 14 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 6 \\ 40 \end{bmatrix}$	$\begin{bmatrix} 10 & 1 \\ 2 & 14 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 50 \\ 40 \end{bmatrix}$
Solution	$(x, y) = (4.89, 1.11)$	$(x, y) = (3.67, 2.33)$	$(x, y) = (4.78, 2.17)$
Feasible	Yes	Yes	No
Grade	85. 33	88.99	~~99.91~~

So the solution of the linear programming is $(x, y) = (3.67, 2.33)$, which means he needs to spend 3.67 hours on study and 2.33 hours on programming, and the best score he can get is 88.99.

1 Assignments

1. (20%) Design your own application which can be formulated as a linear programming problem. Your problem should have at least four variables.

2. (20%) Write a program to solve your own problem defined in (1).

3. (20%) Suppose there are n variables and m constraints, $n < m$. Design and implement an algorithm to enumerate all intersection points.

4. (20%) Enumerating all the intersection points needs to solve many linear systems. However, those linear systems are related. Suppose an $N \times N$ matrix A is the coefficient matrix of N hyperplanes and matrix A' is the coefficient matrix of the same hyperplanes except the last one. In another word, A' is the same as A except the last row.

   - Show that $A' = A + uv^T$, where u are v are two vectors.
   - Show that the inverse of A' can be computed by the following equality (Sherman–Morrison formula),

   $$A'^{-1} = (A + uv^T)^{-1} = A^{-1} - \frac{A^{-1}uv^TA^{-1}}{1 + v^TA^{-1}u}.$$

   So if you know what A^{-1}, you can compute A'^{-1} faster. (The Gaussian elimination takes $O(N^3)$ time. Computing above equation only takes $O(N^2)$.)

5. (20%) Can you think an approach to find the optimal solution without enumerating all the intersection points? You are welcome to google it, but you need to state it by your own word and cite the source.

2 Submission

1. Write a report in PDF file that includes the answers of question (1), (4), and (5).
2. The code of (2), (3) should be implemented in the file `implement.py`.
3. Zip them and submit to eeclass.