Notes for Exam 1 – Systems Optimization

Tips

- Inequality signs (<=, >=, etc… for graphs and systems)
- Be sure to check min/max before optimizing

Problem 1 - (20 points)
Pepita Perez is a famous woodworker that makes "rustic" large and "fancy" medium-sized dining tables. To make these tables, he uses only three materials: wood, paint, and glue. Table 1 provides the basic data of the problem:

Table 1. Raw material utilization and availability for production of tables, along with profit per type of table

	"Rustic" large dining tables (units of raw material per table)	"Fancy" medium dining tables (units of raw material per table)	Maximum daily availability (units of raw material)
Wood (2'x4'x16' studs)	10	5	90
Paint (ounces)	4	8	72
Glue (ounces)	2	2	22

Profit per table ($1000)	10	2

Given that Pepita Perez is so famous and respected, the demand for her tables is always very high, so she always sells all the tables she makes.

You want to determine the production plan that maximizes Pepita Perez's profit. To do this, first you decide to formulate this problem as an LP model. In particular:

a. (5 points) Define the decision variables for this model

X_1 : # of rustic tables made
X_2 : # of fancy tables made

b. (5 points) What is the objective function of this LP model? Do not forget to indicate if you are maximizing or minimizing it.

$$\text{Maximize} \quad 10X_1 + 2X_2$$

c. (5 points) What are the constraints for this problem?

$$10X_1 + 5X_2 \leq 90$$
$$4X_1 + 8X_2 \leq 72$$
$$2X_1 + 2X_2 \leq 22$$
$$X_1 \geq 0$$
$$X_2 \geq 0$$

Graph Optimization Problem

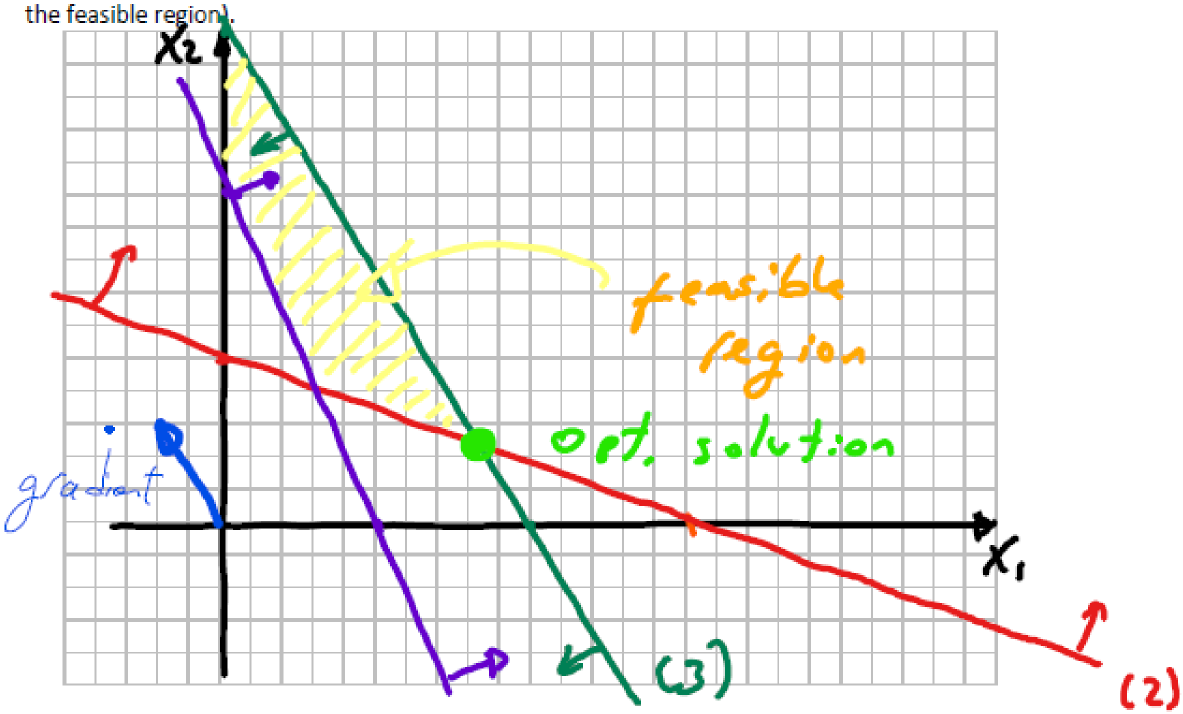
Problem 2 (25 points)
Suppose you have the following LP model

$$\begin{aligned} &\text{Minimize } z = -4x_1 + 6x_2 \\ &\text{s.t.} \\ &4x_1 + 2x_2 \geq 40 \quad (1) \\ &2x_1 + 6x_2 \geq 60 \quad (2) \\ &6x_1 + 4x_2 \leq 120 \quad (3) \\ &x_1, x_2 \geq 0 \end{aligned}$$

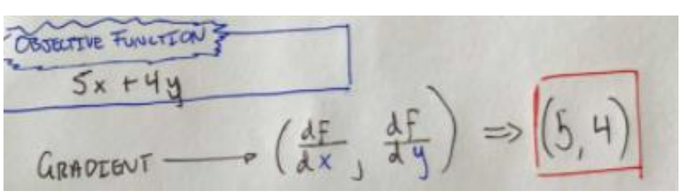
a. (5 points) What is the gradient of this problem?

(-4, 6)

b. (15 points) Plot the gradient and the feasible region (clearly indicating all the constraints and "shading" the feasible region).



Solving Gradient given objective function



Initialize Two-phase and Big-M – *Linear System*

$$
\begin{aligned}
&\text{Minimize } z = 2x_1 - 4x_2 + 8x_3 \\
&\text{s.t.} \\
&\qquad 2x_1 + x_2 \leq 20 \\
&\qquad x_1 + 4x_2 + 8x_3 \leq 40 \\
&\qquad x_1, x_2, x_3 \geq 0
\end{aligned}
$$

Big M - Linear System

$$
\begin{aligned}
\text{Max} \quad & -20\hat{w}_1 - 40\hat{w}_2 - Mr \\
\text{s.t.} \quad & -\hat{w}_1 - \hat{w}_2 + y_1 = 2 \\
& \hat{w}_1 + 4\hat{w}_2 - y_2 + r = 4 \\
& -8\hat{w}_2 + y_3 = 8 \\
& \hat{w}_1, \hat{w}_2, y_1, y_2, y_3, r \geq 0
\end{aligned}
$$

Two-phase – Linear System

$$
\begin{aligned}
\text{Min} \quad & r \\
\text{s.t.} \quad & -\hat{w}_1 - \hat{w}_2 + y_1 = 2 \\
& \hat{w}_1 + 4\hat{w}_2 - y_2 + r = 4 \\
& -8\hat{w}_2 + y_3 = 8 \\
& \hat{w}_1, \hat{w}_2, y_1, y_2, y_3, r \geq 0
\end{aligned}
$$

Initialize Two-phase and Big-M – *Matrix*

Maximize $z = 2x_1 + 3x_2$

subject to

$$2x_1 + x_2 \leq 4$$
$$x_1 + x_2 \geq 2$$
$$x_1, x_2 \geq 0$$

Big M – Matrix

- -1 = slack var
- 1000 = slack row at *r*
- -1 = Z row at *r*
- RHS = positive
- LHS = same as original constraint (except slack var)
- Make r column identity, then optimize via simplex

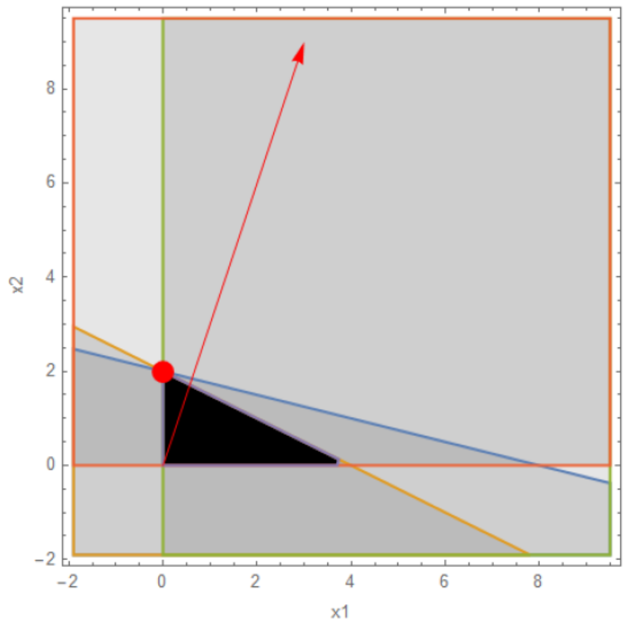
Basis	Z	x1	x2	s1	s2	r	sol
Z	1	-2	-3	0	0	1000	0
	0	2	1	1	0	0	4
	0	1	1	0	-1	1	2

Two Phase – Matrix

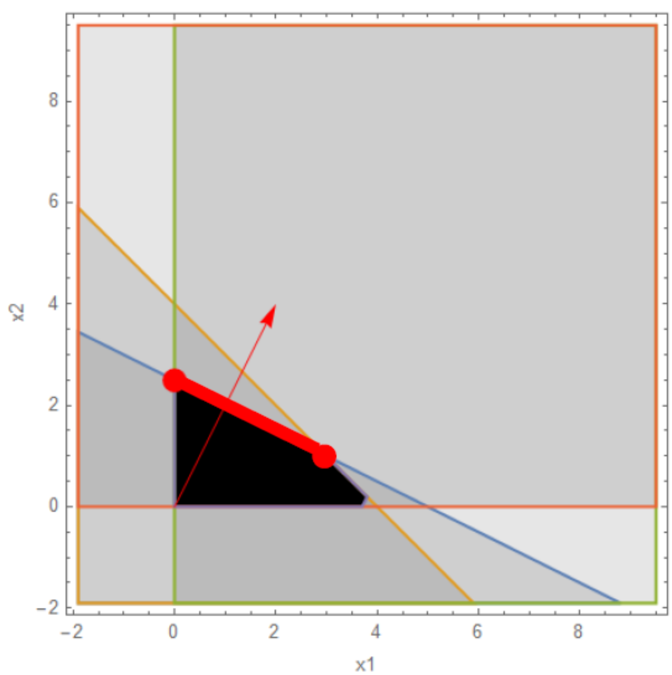
- -1 = slack var
- 1 = slack row at *r*
- -1 = Z row at *r*
- RHS = positive
- LHS = same as original constraint (except slack var)
- Make r column identity, then optimize via simplex
- Drop R row once found initial basic solution

Basis	z	x1	x2	s1	s2	r	sol
z	1	-2	-3	0	0	-1	0
	0	2	1	1	0	0	4
	0	1	1	0	-1	1	2

Degeneracy: Two constraints overlap at the optimal solution. Remove one constraint if this happens.



Alternative Optima: two feasible optimal solutions. Use path of least iterations.



```
#Import gurobi and name model
from gurobipy import *
model=Model("Reddy_Mikks_Company")

#Define decision variables
x1={}
x1=model.addVar(vtype=GRB.CONTINUOUS, lb=0, ub=GRB.INFINITY)

x2={}
x2=model.addVar(vtype=GRB.CONTINUOUS, lb=0, ub=GRB.INFINITY)

#Define objective function
z=5*x1+4*x2
model.setObjective(z)
model.modelSense=GRB.MAXIMIZE
model.update()

#Add constraints
model.addConstr(6*x1+4*x2<=24)
model.addConstr(x1+2*x2<=6)
model.addConstr((-1)*x1+x2<=1)
model.addConstr(x2<=2)

model.update()

#Solve the model
model.optimize()

#printing outputs
if model.status==GRB.OPTIMAL:
    print ("\n Optimal value (profit in USD thousands):", model.objVal)
    print ("--- Production quantities---")
    print ("x1", x1.x)
    print ("x2", x2.x)
```