

DM545/DM871 – Linear and integer programming

Exercise Sheet 5, Autumn 2026

Solution:

Included.

Exercises with the symbol $^+$ are to be done at home before the class. Exercises with the symbol $*$ will be tackled in class and should be at least read at home. The remaining exercises are left for self training after the exercise class.

Exercise 1 $^+$

Using the Lagrangian multipliers method we showed in class that

$$\text{the dual of } \min\{c^T x \mid Ax = b, x \geq 0\} \text{ is } \max\{b^T y \mid A^T y \leq c\}.$$

Using the same method, show that the dual of

$$\text{the dual of } \max\{c^T x \mid Ax \leq b, x \geq 0\} \text{ is } \min\{b^T y \mid A^T y \geq c, y \geq 0\}.$$

Solution:

We want to create a relaxation by enlarging the feasibility region and ensuring that in the feasibility region the new objective function stays above $c^T x$ because we want to approximate the function from above. For feasible solutions to the original problem we have: $b - Ax \geq 0$. Hence, when we relax this quantity in the objective function we multiply by nonnegative coefficients $y \in \mathbb{R}_+^n$.

$$\begin{aligned} & \min_{y \in \mathbb{R}_+^n} \{ \max_{x \in \mathbb{R}_+^m} \{ c^T x + y^T (b - Ax) \} \} \\ & \min_{y \in \mathbb{R}_+^n} \{ \max_{x \in \mathbb{R}_+^m} \{ (c^T - y^T A)x + y^T b \} \} \end{aligned}$$

The inner problem is easy to solve when we are trying to minimize the outer problem, since this latter one we will try to make $c - A^T y \leq 0$. Hence, we are left with the problem

$$\begin{aligned} \max \quad & b^T y \\ \text{subject to} \quad & A^T y \geq c \\ & y \in \mathbb{R}_+^n \end{aligned}$$

Exercise 2 $*$

Consider the following problem:

$$\begin{aligned} \text{maximize} \quad & z = x_1 - x_2 \\ \text{subject to} \quad & x_1 + x_2 \leq 2 \\ & 2x_1 + 2x_2 \geq 2 \\ & x_1, x_2 \geq 0 \end{aligned}$$

In the ordinary simplex method this problem does not have an initial feasible basis. Hence, the method has to be enhanced by a preliminary phase to attain a feasible basis. Traditionally we talk about a *phase I–phase II* simplex method. In phase I an initial feasible solution is sought and in phase II the ordinary simplex is started from the initial feasible solution found.

There are two ways to carry out phase I.

- Solving an auxiliary LP problem defined by introducing auxiliary variables and minimizing them in the objective. The solution of the auxiliary LP problem gives an initial feasible basis or a proof of infeasibility. (This was used in Sheet 4.)
- Applying the dual simplex on a possibly modified problem to find a feasible solution. If the initial infeasible tableau of the original problem is not optimal then the objective function can be temporarily modified for this phase in order to make the initial tableau optimal although not feasible. Opposite to the primal simplex method, the dual simplex method iterates through infeasible basic solutions, while maintaining them optimal, and stops when a feasible solution is reached.

Dual Simplex: The strong duality theorem states that we can solve the primal problem by solving its dual. You can verify that applying the *primal simplex method* to the dual problem corresponds to the following method, called *dual simplex method* that works on the primal problem:

1. (Feasibility condition) select the leaving variable by picking the basic variable whose right-hand side term is negative, i.e., select i^* with $b_{i^*} < 0$.
2. (Optimality condition) pick the entering variable by scanning across the selected row and comparing ratios of the coefficients in this row to the corresponding coefficients in the objective row, looking for the largest negated. Formally, select j^* such that $j^* = \min\{|c_j/a_{ij}| : a_{ij} < 0\}$
3. Update the tableau around the pivot in the same way as with the primal simplex.
4. Stop if no right-hand side term is negative.

Duality can help us with the issue of initial feasible basic solutions. In the problem above, the initial tableau is infeasible and not optimal, hence we cannot apply the primal simplex nor the dual simplex. However, if the objective function was $w = -x_1 - x_2$, then we would have the conditions of infeasibility and optimality needed by the dual simplex. You can understand this also looking at the dual problem. With the new objective function the initial basic solution of the dual problem would be feasible and we could solve the problem solving the dual problem with the primal simplex. In contrast, with original objective function z the primal simplex has infeasible initial basis in both problems. So we can change temporarily the objective function z with w and apply the dual simplex method to the primal problem. When it stops we reached a feasible solution that is optimal with respect to w . We can then reintroduce the original objective function and continue iterating with the primal simplex. The phase I–phase II simplex method that uses the dual simplex is also called the *dual-primal simplex method*.

Apply the phase I–phase II simplex method where the phase I is carried out with the dual simplex to the problem above and verify that they lead to the same solutions.

Solution:

$$\begin{array}{rclcl} \max & x_1 & - & x_2 & = & z \\ & x_1 & + & x_2 & \leq & 2 \\ & 2x_1 & + & 2x_2 & \geq & 2 \\ & & & x_1, x_2 & \geq & 0 \end{array}$$

We put in equational standard form by introducing a slack variable $s_1 \geq 0$ and a surplus variable $s_2 \geq 0$:

$$\begin{array}{rclcl} \max & x_1 & - & x_2 & = & z \\ & x_1 & + & x_2 & + & s_1 & = & 2 \\ & 2x_1 & + & 2x_2 & - & s_2 & = & 2 \\ & & & x_1, x_2, s_1, s_2 & \geq & 0 \end{array}$$

This form is not canonical and therefore the first tableau does not have a feasible starting solution.

Dual-Primal Simplex Method

Phase I Let's write the dual of the problem above:

$$\begin{array}{rclcl} \max & x_1 & - & x_2 & = & z \\ & x_1 & + & x_2 & \leq & 2 \\ & 2x_1 & + & 2x_2 & \geq & 2 \\ & & & x_1, x_2 & \geq & 0 \end{array} \quad \begin{array}{rclcl} \min & 2y'_1 & + & 2y'_2 & = & w \\ & y'_1 & + & 2y'_2 & \geq & 1 \\ & y'_1 & + & 2y'_2 & \geq & -1 \\ & & & y'_1 & \geq & 0 \\ & & & y'_2 & \leq & 0 \end{array} \quad \begin{array}{l} y'_1 = y_1 \\ y'_2 = -y_2 \\ \longrightarrow \end{array}$$

$$\begin{array}{rclcl}
 \min & 2y_1 & - & 2y_2 & = & w \\
 & y_1 & - & 2y_2 & \geq & 1 \\
 & y_1 & - & 2y_2 & \geq & -1 \\
 & & & y_1, y_2 & \geq & 0
 \end{array}$$

If we put this LP problem in standard form:

$$\begin{array}{rclcl}
 \max & -2y_1 & + & 2y_2 & = & w \\
 & -y_1 & + & 2y_2 & \leq & -1 \\
 & -y_1 & + & 2y_2 & \leq & 1 \\
 & & & y_1, y_2 & \geq & 0
 \end{array}$$

and look at the tableau:

y1	y2	s1	s2	-z	b
-1	2	1	0	0	-1
-1	2	0	1	0	1
-2	2	0	0	1	0

we see that the initial tableau like for the primal problem is infeasible.

However, the dual problem has an advantage, if we change temporarily the objective function of the primal problem to $\eta = -x_1 - x_2$, the dual problem becomes:

$$\begin{array}{rcl}
 \max & -x_1 - x_2 = \eta & \min & 2y_1 - 2y_2 = \gamma & \max & -2y_1 + 2y_2 = \gamma \\
 & x_1 + x_2 \leq 2 & & y_1 - 2y_2 \geq -1 & & -y_1 + 2y_2 \leq 1 \\
 & 2x_1 + 2x_2 \geq 2 & & y_1 - 2y_2 \geq -1 & & -y_1 + 2y_2 \leq 1 \\
 & x_1, x_2 \geq 0 & & y_1, y_2 \geq 0 & & y_1, y_2 \geq 0
 \end{array}$$

and the corresponding tableau has an easy feasible basic solution:

y1	y2	s1	s2	-z	b
-1	2	1	0	0	1
-1	2	0	1	0	1
-2	2	0	0	1	0

We can then solve to optimality with the primal simplex: the variable y_2 enters the basis and the variable s_2 exits. The new tableau becomes:

	y1	y2	s1	s2	-z	b
R1	-1	2	1	0	0	1
R2' = R2/2	-1/2	1	0	1/2	0	1/2
R3' = R3 - R2	-1	0	0	-1	1	-1/2

and it is optimal. At this stage we can go back to the primal problem where we now have a feasible solution, change the objective function back to the original one and continue with the primal simplex.

We can do the same iteration on the primal but with the dual simplex. Let's write the tableau of the primal with the objective function temporarily changed and keeping the old objective as well:

x1	x2	s1	s2	-z	-e	b
1	1	1	0	0	0	2
-2	-2	0	1	0	0	-2
1	-1	0	0	1	0	0
-1	-1	0	0	0	1	0

As we see we have the conditions of the dual simplex satisfied, the tableau is optimal but not feasible. Let's make an iteration of the dual simplex. We choose the row with negative b term and the column with negative pivot that minimizes the ratio test: $|c/a|$. We choose the second row and the second column (again watch out that we do not consider the row of the old objective to decide the column). In other terms, we try to make the solution feasible while minimizing the loss in quality. The operations to update the tableau remain the same as for the primal simplex. We obtain:

	x1	x2	s1	s2	-z	-e	b
R1' = R1 - R2'	0	0	1	1/2	0	0	1
R2' = -1/2 R2	1	1	0	-1/2	0	0	1
R3' = R3 + R2'	2	0	0	-1/2	1	0	1
R4' = R4 + R2'	0	0	0	-1/2	0	1	1

This tableau is optimal for the dual simplex, this means that a feasible solution for the primal problem has been found: $(0, 1, 1, 0)$. We can now proceed with the primal simplex.

Note that the considerations on the dual problem made above were just for explanation purposes, when solving our LP problem we do not need to write the dual form of it or its tableaux. Instead, we just need to switch from dual simplex to primal simplex always working on the original (the primal) formulation of the problem. The dual simplex method is simply a new way of picking the entering and leaving variables in a sequence of primal tableaux.

Phase II We can now remove the temporary objective function and the corresponding column and proceed with the primal simplex.

x1	x2	s1	s2	-z	b
0	0	1	1/2	0	1
1	1	0	-1/2	0	1
2	0	0	-1/2	1	1

x_1 enters the basis and x_2 exits. The tableau is updated consequently:

	x1	x2	s1	s2	-z	b
R1' = R1	0	0	1	1/2	0	1
R2' = R2	1	1	0	-1/2	0	1
R3' = R3 - 2*R2	0	-2	0	1/2	1	-1

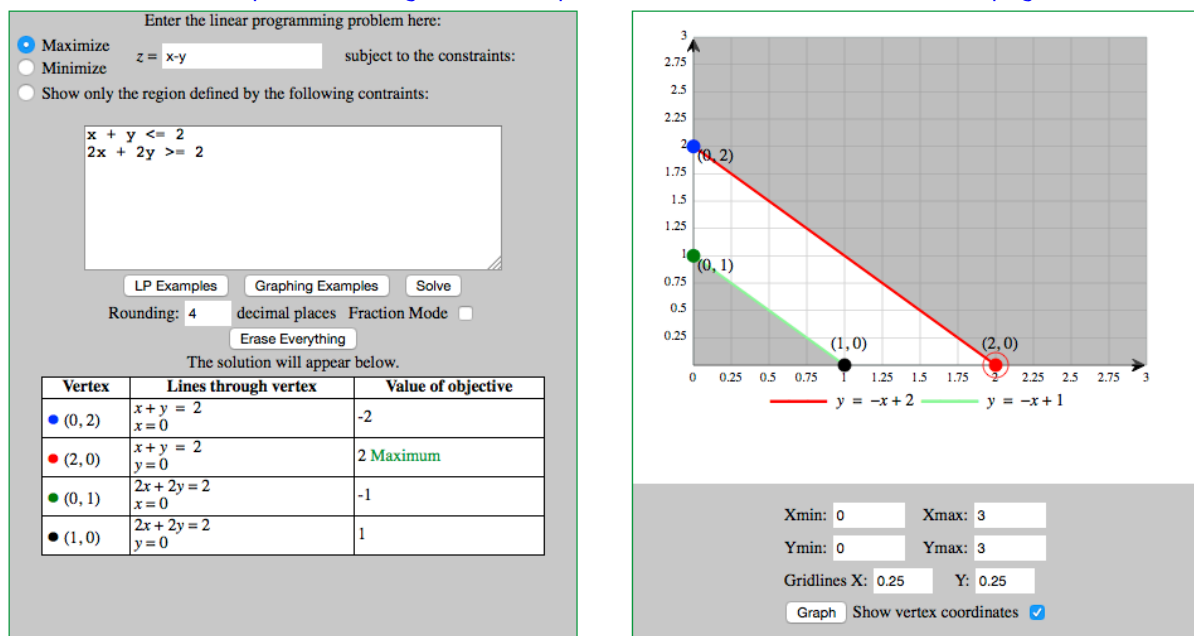
|-----+-----+-----+-----+-----+-----+-----|

A reduced cost is still positive, hence we make s_2 enters in the basis and s_1 leave. This leads to

	x_1	x_2	s_1	s_2	$-z$	b
R1' = 2*R1	0	0	2	1	0	2
R2' = R2 + R1	1	1	1	0	0	2
R3' = R3 - R1	0	-2	-1	0	1	-2

The tableau is now optimal and the corresponding basic feasible solution is $x = (2, 0)$ and has value $z^* = 2$.

We can visualize the problem using the LP Grapher tool linked from the course webpage:



Exercise 3* Sensitivity Analysis

A furniture-manufacturing company can produce four types of products using three resources.

- A bookcase requires three hours of work, one unit of metal, and four units of wood and it brings in a net profit of 19 Euro.
- A desk requires two hours of work, one unit of metal and three units of wood, and it brings in a net profit of 13 Euro.
- A chair requires one hour of work, one unit of metal and three units of wood and it brings in a net profit of 12 Euro.
- A bedframe requires two hours of work, one unit of metal, and four units of wood and it brings in a net profit of 17 Euro.
- Only 225 hours of labor, 117 units of metal and 420 units of wood are available per day.

In order to decide how much to make of each product so as to maximize the total profit, the managers solve an LP problem.

1) Write the mathematical programming formulation of the problem.

Solution:

$$\begin{aligned} \max \quad & 19x_1 + 13x_2 + 12x_3 + 17x_4 \\ & 3x_1 + 2x_2 + x_3 + 2x_4 \leq 225 \\ & x_1 + x_2 + x_3 + x_4 \leq 117 \\ & 4x_1 + 3x_2 + 3x_3 + 4x_4 \leq 420 \\ & x_1, x_2, x_3, x_4 \geq 0 \end{aligned}$$

2) With the help of a computational environment such as Python for carrying out linear algebra operations, write the optimal tableau, which has x_1, x_3 and x_4 in basis.

You can achieve this task in two ways:

- using the traditional simplex method
- writing the matrices A_B and A_N which are the block matrices derived from A by subsetting the basis and non-basis columns, respectively. Then calculate $A_B^{-1}A_N$, and finally derive the optimal tableau and verify that the solution is indeed optimal.

Solution:

The initial tableau is:

x_1	x_2	x_3	x_4	x_5	x_6	x_7	-z	b
3	2	1	2	1	0	0	0	225
1	1	1	1	0	1	0	0	117
4	3	3	4	0	0	1	0	420
19	13	12	17	0	0	0	1	0

We know that there will be 3 variables in basis. The text of the problem tells us which these 3 variables are: 1, 3, 4. Hence,

$$A_B = \begin{bmatrix} 3 & 1 & 2 \\ 1 & 1 & 1 \\ 4 & 3 & 4 \end{bmatrix} \quad A_N = \begin{bmatrix} 2 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 3 & 0 & 0 & 1 \end{bmatrix}$$

We can calculate $A_B^{-1}A_N$ in Python or in R:

```
> B=matrix(c(3,1,2,1,1,1,4,3,4),byrow=TRUE,ncol=3)
> B1=solve(B)
> B1*B1 # check to make sure it is correct!
      [,1] [,2] [,3]
[1,]    1    0    0
[2,]    0    1    0
[3,]    0    0    1
> N=matrix(c( 2, 1, 0, 0, 1, 0, 1, 0, 3, 0, 0, 1),ncol=4,byrow=TRUE)
> B1*N
      [,1] [,2] [,3] [,4]
[1,]    1    1    2   -1
[2,]    1    0    4   -1
[3,]   -1   -1   -5    2
> cN=c(13,0,0,0)
> cB=c(19,12,17)
> cN-cB*B1*N
```

```

      [,1] [,2] [,3] [,4]
[1,]    -1    -2    -1    -3
> cB%*%B1%*%c(225,117,420)
> 1827

```

This code gives us:

$$\bar{A} = A_B^{-1}A_N = \begin{bmatrix} 1 & 1 & 2 & -1 \\ 1 & 0 & 4 & -1 \\ -1 & -1 & -5 & 2 \end{bmatrix} \quad x_B^* = \begin{bmatrix} x_1 \\ x_3 \\ x_4 \end{bmatrix} = A_B^{-1}b = \begin{bmatrix} 39 \\ 48 \\ 30 \end{bmatrix}$$

$$\bar{c}_N = [\bar{c}_2 \quad \bar{c}_5 \quad \bar{c}_6 \quad \bar{c}_7] = [-1 \quad -2 \quad -1 \quad -3]$$

and we can write the final tableau as:

x_1	x_2	x_3	x_4	x_5	x_6	x_7	-z	b
1	1	0	0	1	2	-1	0	39
0	1	1	0	0	4	-1	0	48
0	-1	0	1	-1	-5	2	0	30
0	-1	0	0	-2	-1	-3	1	-1827

Since all reduced costs are negative then the tableau and the corresponding solution are optimal.

- 3) What is the increase in price (reduced cost) that would make product x_2 worth to be produced?

Solution:

The increase in price of a quantity strictly larger than 1 would make the product 2 worth being produced. Indeed, let $c'_2 = c_2 + \delta$ be the new price. We know that the coefficient in the objective function goes in the reduced cost calculation multiplied by 1. Hence, to have a positive reduced cost we have:

$$-1 + \delta > 0 \quad \Rightarrow \quad \delta > 1$$

We could also recalculate the reduced cost from scratch using the multipliers π : $c'_2 + \sum_{i=1}^3 \pi_i a_{i2}$. The value of π_i are read from the final tableau and they correspond to the reduced costs of the slack variables, ie, $(-2, -1, -3)$.

- 4) What is the marginal value (shadow price) of an extra hour of work or amount of metal and wood?

Solution:

The marginal values are the values of the dual variable y_1, y_2, y_3 . From the strong duality theorem, we know that $y_i = -\pi_i = -\bar{c}_{n+i}$, $i = 1..m$. Hence, $y = (2, 1, 3)$.

An extra hour of work has marginal value of 2, that is, having one unit more of work would improve the revenue by 2. For the other two resources the marginal values are 1 and 3, respectively.

We can cross check these conclusions: by the complementary slackness theorem, the fact that all three dual variables are strictly positive indicates that all three constraints in the primal are active $\equiv$ tight $\equiv$ binding. Hence, it makes sense to have that an increase in the capacity of those constraints implies an increase in the profit. The conclusion that all three constraints are tight can be also reached by the fact that the slack variables are 0 in the final tableau. If some constraint was not tight, then the marginal value of the corresponding resource would be zero since an increase in its capacity does not imply an immediate improvement in total profit.

- 5) Are all resources totally utilized, i.e. are all constraints "binding", or is there slack capacity in some of them? Answer this question in the light of the complementary slackness theorem.

Solution:

Since all dual variables are strictly larger than zero, then all constraints are binding. Indeed for the complementary slackness theorem, we have that:

$$\left(b_i - \sum_{j=1}^n a_{ij}x_j^* \right) y_i^* = 0, \quad i = 1, \dots, m$$

- 6) From the economical interpretation of the dual why product x_2 is not worth producing? What is its imputed cost?

Solution:

It is not worth producing 2 because $\sum_i y_i a_{i2} > c_2$, that is, we are better off selling the raw materials to produce the product. Indeed y_i is the price of one unit of resource i and a_{i2} is the amount of i necessary to produce 2.

$$\sum_i y_i a_{i2} = 2 * (2) + 1 * (1) + 3 * (3) = 14 > 13$$

Perform a sensitivity analysis for the following variants:

- 7) The net profit brought in by each desk increases from 13 Euro to 15 Euro.

Solution:

We saw earlier that if the price of product 2 increases by more than 1 then the reduced cost becomes positive and it enters the basis. We can iterate the revised simplex as follows:

Step 1 and 2 to determine the entering variable are already done in the point a) above.

We need to do Step 3 to determine the leaving variable: we need to find the constraint that limit the increase of x_2 , θ . We solve first $A_B d = a$ in d . Here, a is the column of the matrix A (augmented with the slack variables) from the initial tableau corresponding to the entering variable x_2 . We use the inverse of A_B calculated earlier in a) above in R:

```
> B1%*%c(2,1,3)
      [,1]
[1,]    1
[2,]    1
[3,]   -1
```

that is

$$d = A_B^{-1} a = A_B^{-1} \begin{bmatrix} 2 \\ 1 \\ 3 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix}$$

Then the new solution x_B is derived from the old one by means of d and the increase θ :

$$x_B = \begin{bmatrix} x_1 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 39 \\ 48 \\ 30 \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix} \theta \geq 0$$

The increase θ must be such that the value of the variables still remains feasible, ie, $x_i \geq 0$. Hence $\theta \leq 39$ and the leaving variable is x_1 , since it is the one that goes to zero. The new solutions is

$$x_B = \begin{bmatrix} x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 39 \\ 48 \\ 30 \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix} \theta = \begin{bmatrix} 39 - 39 \\ 48 - 39 \\ 30 + 39 \end{bmatrix} = \begin{bmatrix} 0 \\ 9 \\ 69 \end{bmatrix}$$

and the objective value:

```
> c=c(19,15,12,17)
> c%*%c(0,39,9,69)
      [,1]
[1,] 1866
```

- 8) The availability of metal increases from 117 to 125 units per day

Solution:

This is a change in the RHS term of constraint 2. The optimality of the current solution does not change, since all reduced costs stay negative, but we need to check if it is still feasible. We need to look at the final tableau and recompute the b terms of all constraints. We can do this with $A_B^{-1}b$:

```
> b=c(225,125,420)
> B1%*%b
      [,1]
[1,]    55
[2,]    80
[3,]   -10
```

The last constraint becomes negative, hence we need to iterate with the dual simplex.

- 9) The company may also produce coffee tables, each of which requires three hours of work, one unit of metal, two units of wood and bring in a net profit of 14 Euro.

Solution:

We need to check if the reduced cost of the new variable would become positive by computing $c_0 + \sum_i \pi_i a_{ij}$:

```
> 14-3*2-1*1-2*3
[1] 1
```

which is positive, hence we need to iterate as done in point 1).

- 10) The number of chairs produced must be at most five times the numbers of desks

Solution:

This corresponds to introduce a new constraint: $x_3 \leq 5x_2$. In the new standard form we have a new slack variable x_8 . Adding the constraint in the tableau and bringing back the tableau in canonical standard form we observe that a RHS term becomes negative. Hence, we need to iterate with the dual simplex. After one iteration with the dual simplex, the final tableau becomes:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 4/3 & -5/6 & 1/6 & 0 & 31 \\ 0 & 1 & 0 & 0 & 0 & 2/3 & -1/6 & -1/6 & 0 & 8 \\ 0 & 0 & 0 & 1 & -1 & -13/3 & 11/6 & -1/6 & 0 & 38 \\ 0 & 0 & 1 & 0 & 0 & 10/3 & -5/6 & 1/6 & 0 & 40 \\ 0 & 0 & 0 & 0 & -2 & -1/3 & -19/6 & -1/6 & 1 & -1819 \end{bmatrix}$$

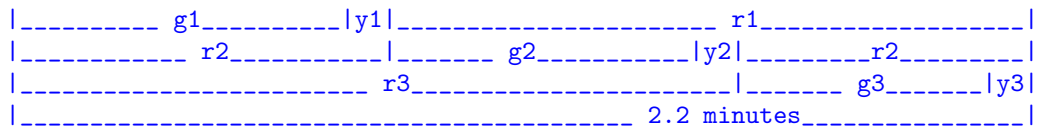
If after the introduction of the constraint the current solution stayed feasible then, we would have needed to check whether it was also optimal. We can either repeat the steps done at part 1 above to compute the new reduced costs or we can include the new row in the final tableau and proceed to put the tableau in canonical form. Then we look at the value of the reduced costs.

Exercise 4* Traffic Light Control

Automobile traffic from three highways, H1, H2, and H3, must stop and wait for a green light before exiting to a toll road. The tolls are 40 kr, 50 kr, and 60 kr for cars exiting from H1, H2, and H3, respectively. The flow rates from H1, H2, and H3 are 550, 650, and 450 cars per hour. The traffic light

cycle may not exceed 2.2 minutes, and the green light on any highway must be at least 22 seconds. The yellow light is on for 10 seconds. The toll gate can handle a maximum of 500 cars per hour. Assuming that no cars move on yellow, and that clearly when a highway has the green signal the other two have the red, determine the optimal green time interval for the three highways that will maximize toll gate revenue per traffic cycle.

Solution:



Let g_i , y_i and r_i be the durations of green, yellow and red lights for the cars exiting highway i . All time units are in seconds. No cars move on yellow.

$$\begin{aligned} \max \quad & z = 40(550/3600)g_1 + 50(650/3600)g_2 + 60(450/3600)g_3 \\ & (550/3600)g_1 + (650/3600)g_2 + (450/3600)g_3 \leq (500/3600)(g_1 + g_2 + g_3 + 3 \cdot 10) \\ & g_1 + g_2 + g_3 + 3 \cdot 10 \leq 2.2 \cdot 60 \\ & g_1, g_2, g_3 \geq 22 \end{aligned}$$

The objective function measures the income using the flow per hour transformed in flow per second. The first constraint ensures that the limit of 500 cars per hours is not exceeded. The second ensure the duration of the cycle. The last constraints ensure that the duration of the green light satisfies the minimum requirement.

```
#!/usr/bin/python3
import pyomo.environ as po

model = po.ConcreteModel("traffic_light")

infinity = float('inf')
# declare decision variables
model.g1 = po.Var(bounds=(22,infinity),domain=po.NonNegativeReals)
model.g2 = po.Var(bounds=(22,infinity),domain=po.NonNegativeReals)
model.g3 = po.Var(bounds=(22,infinity),domain=po.NonNegativeReals)

# declare objective
model.profit = po.Objective(
    expr = 40*(550/3600)*model.g1+50*(650/3600)*model.g2+60*(450/3600)*model.g3,
    sense = po.maximize)

# declare constraints
model.capacity = po.Constraint(expr = (550/3600)*model.g1+(650/3600)*model.g2+(450/3600)*
    model.g3<= (500/3600)*(model.g1+model.g2+model.g3+3* 10))
model.max_cycle = po.Constraint(expr = model.g1+model.g2+model.g3+3*10<= 2.2 * 60)

model.pprint()
model.write("traffic_light.lp")

# solve
solver = po.SolverFactory('gurobi')
results = solver.solve(model, tee=True)

if results.solver.termination_condition == po.TerminationCondition.optimal:
    for v_data in model.component_data_objects(po.Var, descend_into=True):
        print("Found: "+v_data.name+", value = "+str(po.value(v_data)))
```

```
max
x4: +6.111111111111116 x1 +9.027777777777768 x2 +7.5 x3

s.t.

c_u_x5_:
+0.01388888888888895 x1 +0.04166666666666657 x2 -0.01388888888888895 x3 <= 4.166666666666667

c_u_x6_:
+1 x1 +1 x2 +1 x3 <= 102

c_e_ONE_VAR_CONSTANT:
ONE_VAR_CONSTANT = 1.0

bounds
    22 <= x1 <= +inf
    22 <= x2 <= +inf
    22 <= x3 <= +inf
end
```

3 Var Declarations

```

g1 : Size=1, Index=None
    Key : Lower : Value : Upper : Fixed : Stale : Domain
    None : 22 : None : inf : False : True : NonNegativeReals
g2 : Size=1, Index=None
    Key : Lower : Value : Upper : Fixed : Stale : Domain
    None : 22 : None : inf : False : True : NonNegativeReals
g3 : Size=1, Index=None
    Key : Lower : Value : Upper : Fixed : Stale : Domain
    None : 22 : None : inf : False : True : NonNegativeReals

```

1 Objective Declarations

```

profit : Size=1, Index=None, Active=True
    Key : Active : Sense : Expression
    None : True : maximize : 6.11111111111112*g1 + 9.02777777777777*g2 + 7.5*g3

```

2 Constraint Declarations

```

capacity : Size=1, Index=None, Active=True
    Key : Lower : Body
    None : -Inf : 0.152777777777778*g1 + 0.180555555555555*g2 + 0.125*g3 - 0.138888888888888
max_cycle : Size=1, Index=None, Active=True
    Key : Lower : Body : Upper : Active
    None : -Inf : g1 + g2 + g3 + 30 : 132.0 : True

```

6 Declarations: g1 g2 g3 profit capacity max_cycle

Using license file /Library/gurobi900/gurobi.lic

Academic license - for non-commercial use only

Reading time = 0.00 seconds

x4: 3 rows, 4 columns, 7 nonzeros

Gurobi Optimizer version 9.0.0 build v9.0.0rc2 (mac64)

Optimize a model with 3 rows, 4 columns and 7 nonzeros

Model fingerprint: 0x5801878b

Coefficient statistics:

```

Matrix range      [1e-02, 1e+00]
Objective range    [6e+00, 9e+00]
Bounds range       [2e+01, 2e+01]
RHS range          [1e+00, 1e+02]

```

Presolve removed 3 rows and 4 columns

Presolve time: 0.01s

Presolve: All rows and columns removed

Iteration	Objective	Primal Inf.	Dual Inf.	Time
0	8.23055556e+02	0.000000e+00	0.000000e+00	0s

Solved in 0 iterations and 0.01 seconds

Optimal objective 8.23055556e+02

Found: g1, value = 22.0

Found: g2, value = 58.0

Found: g3, value = 22.0

Exercise 5 Factory Planning and Machine Maintenance

Tasks 1-2 of [Factory Planning and Maintenance Case](#).

Exercise 6

The following exercise will be useful to understand the advantages of the revised simplex that we will study in the next lecture.

Solve the systems $\mathbf{y}^T E_1 E_2 E_3 E_4 = [1 \ 2 \ 3]$ and $E_1 E_2 E_3 E_4 \mathbf{d} = [1 \ 2 \ 3]^T$ with

$$E_1 = \begin{bmatrix} 1 & 3 & 0 \\ 0 & 0.5 & 0 \\ 0 & 4 & 1 \end{bmatrix} \quad E_2 = \begin{bmatrix} 2 & 0 & 0 \\ 1 & 1 & 0 \\ 4 & 0 & 1 \end{bmatrix} \quad E_3 = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 3 \\ 0 & 0 & 1 \end{bmatrix} \quad E_4 = \begin{bmatrix} -0.5 & 0 & 0 \\ 3 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

Solution:

This exercise is to show that the two systems can be solved quite easily. Let's take first $\mathbf{y}^T E_1 E_2 E_3 E_4 = [1 \ 2 \ 3]$, we use the backward transformation and solve the sequence of linear systems:

$$\mathbf{u}^T E_4 = [1 \ 2 \ 3], \quad \mathbf{v}^T E_3 = \mathbf{u}^T, \quad \mathbf{w}^T E_2 = \mathbf{v}^T, \quad \mathbf{y}^T E_1 = \mathbf{w}^T$$

$$\mathbf{u}^T \begin{bmatrix} -0.5 & 0 & 0 \\ 3 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} = [1, 2, 3]$$

Since the eta matrices have always one 1 in two columns then the solution can be read up easily. From the third column we find $u_3 = 3$. From the second column, we find $u_2 = 2$. Substituting in the first column, we find $-0.5u_1 + 3 * 2 + 1 * 3 = 1$, which yields $u_1 = 16$. The next system is:

$$\mathbf{v}^T \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 3 \\ 0 & 0 & 1 \end{bmatrix} = [16, 2, 3]$$

From the first column we get $v_1 = 16$, from the second column $v_2 = 2$ from the last column $v_3 = -19$. The next:

$$\mathbf{w}^T \begin{bmatrix} 2 & 0 & 0 \\ 1 & 1 & 0 \\ 4 & 0 & 1 \end{bmatrix} = [16, 2, -19]$$

which gives $\mathbf{w} = [45, 2, -19]$. The next:

$$\mathbf{y}^T \begin{bmatrix} 1 & 3 & 0 \\ 0 & 0.5 & 0 \\ 0 & 4 & 1 \end{bmatrix} = [45, 2, -19]$$

from we finally find $\mathbf{d} = [119/2, 2, -19]$.

We can carry out the operations in Python for cross checking the results but the function `linalg.solve` would not exploit the advantages of the eta matrices when solving the system of linear equations.

```
import numpy as np

E4=np.array([[ -0.5,0,0],[3,1,0],[1,0,1]])
E3=np.array([[1,0,1],[0,1,3],[0,0,1]])
E2=np.array([[2,0,0],[1,1,0],[4,0,1]])
E1=np.array([[1,3,0],[0,0.5,0],[0,4,1]])
u=np.linalg.solve(E4.T,np.array([1,2,3]))
print(u)
v=np.linalg.solve(E3.T,u)
print(v)
w=np.linalg.solve(E2.T,v)
print(w)
d=np.linalg.solve(E1.T,w)
print(d)
```

```
[16.  2.  3.]
[ 16.  2. -19.]
[ 45.  2. -19.]
[ 59.5  2. -19. ]
```