MAT 3007 – Optimization

Assignment 9

Due: 11:59pm, Dec. 15 (Friday), 2023

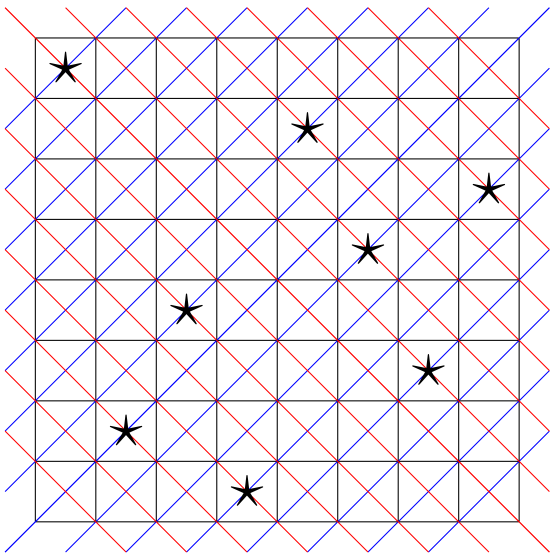
Instructions:

- Homework problems must be carefully and clearly answered to receive full credit. Complete sentences that establish a clear logical progression are highly recommended.
- Please submit your assignment on Blackboard.
- The homework must be written in English.
- Late submission will not be graded.
- Each student must not copy homework solutions from another student or from any other source.

Remark: Please note that only your solutions to problem 1 and 2 will be considered for grading. Problem 3 and 4 are listed in case you would like to have additional practice.

Problem 1 (Eight Queens Problem): (approx. *50* points)
The Eight Queens Problem involves placing eight chess queens on an 8×8 chessboard so that no two queens threaten each other.

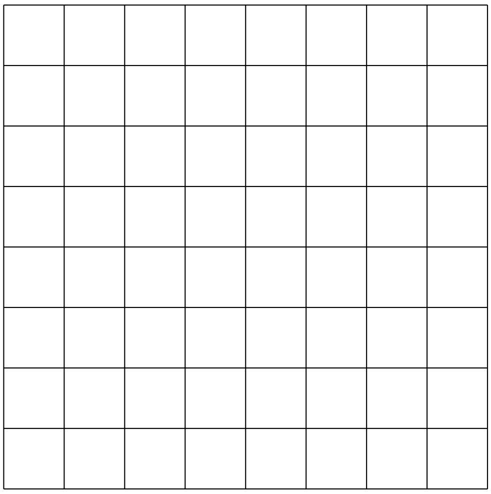
- No two queens on the same row.
- No two queens on the same column.
- No two queens on the same diagonal lines (shown in the figure below).



a) Formulate an integer optimization problem to solve the eight queens problem.

   Describe and explain your optimization model briefly.

b) Implement and solve the LP relaxation of your model using `CVX`, `MATLAB`, or `Python`. What is the solution of the LP relaxation of your model? It is optional for you to solve the original model. Put your solution on the following chess board.



Problem 2 (Branch-and-Bound Method): (approx. *50* points)
Use the branch-and-bound method to solve the following integer program:

$$\begin{array}{lll} \text{maximize} & 17x + 12y & \\ \text{subject to} & 3x + 4y & \leq 25 \\ & 2x - y & \geq 2 \\ & x, y & \geq 0 \\ & x, y \in \mathbb{Z}. & \end{array}$$

Form the branch-and-bound tree and indicate the solution associated with each node (similar to the procedures introduced in the lecture). Please solve the linear programming relaxation and include your calculations and/or code and the solution outputs in your answer.

Problem 3 (Multiple Knapsacks (optional)):
Suppose we have a set of n many items and a set of m different knapsacks. For each item i and knapsack j, the following information is given:

- The item i has value (preference) v_i.
- The weight of item i is a_i.
- The capacity of knapsack j is at most C_j.

a) Formulate an integer program to maximize the total value of items that can be packed in the different knapsack while adhering to the capacity constraint (i.e., the total weight of items in each bag j is not allowed to be larger than C_j).

   Hint: You can introduce variables x_{ij} to denote whether item i is placed in knapsack j.

b) Consider the following list of items and bags:

Item	Laptop	T-Shirt	Swim. Trunks	Sunglasses	Apples	Opt. Book	Water
Value	3	1	3	2	1	4	2
Weight	2	0.5	0.5	0.2	0.5	1	1.3

Knapsack 1	Knapsack 2
$C_1 = 2.5$	$C_2 = 2.5$

Formulate the corresponding IP in that case. What are the optimal solutions to the IP and its LP relaxation (you can use `CVX`, `MATLAB` or `Python` to solve the problems)? Is there an integrality gap in this case?

Solution :

a) We introduce the binary decision variables:

$$x_{ij} = \begin{cases} 1 & \text{if item } i \text{ is packed in knapsack } j, \\ 0 & \text{if item } i \text{ is not packed in knapsack } j. \end{cases}$$

The constraints of the multiple knapsack problem are then given by:

- Capacity constraints: $\sum_{i=1}^{n} a_i x_{ij} \le C_j$.
- Each item can be only be placed in at most one knapsack: $\sum_{j=1}^{m} x_{ij} \le 1$.
- Binary constraints: $x_{ij} \in \{0,1\}$ for all i and j.

Overall the full optimization problem is given by:

$$\begin{aligned} \text{maximize} \quad & \sum_{i=1}^{n} v_i \sum_{j=1}^{m} x_{ij} \\ \text{subject to} \quad & \sum_{i=1}^{n} a_i x_{ij} \le C_j, \qquad \sum_{j=1}^{m} x_{ij} \le 1 \\ & x_{ij} \in \{0,1\}, \quad \forall\, i,j. \end{aligned}$$

b) In this case, we have $n = 7$ and $m = 2$. Let us define $x := (x_{11}, ..., x_{71}, x_{12}, ..., x_{72})^\top$, $a = (a_1, ..., a_7)^\top$, and $v = (v_1, ..., v_7)^\top$. Then, the optimization problem can be represented as follows:

$$\max_x \ (v^\top, v^\top)x \qquad \text{s.t.} \qquad \begin{bmatrix} a^\top & 0 \\ 0 & a^\top \end{bmatrix} x \le \begin{bmatrix} 2.5 \\ 2.5 \end{bmatrix}, \quad \begin{bmatrix} I & I \end{bmatrix} x \le \mathbb{1}, \quad x_{ij} \in \{0,1\} \quad \forall\, i,j.$$

The `MATLAB` code can be found below:

```
v = [3,1,3,2,1,4,2]; c = [v,v];
a = [2,0.5,0.5,0.2,0.5,1,1.3];
A = [a,zeros(1,7);zeros(1,7),a];

cvx_begin
    cvx_solver gurobi
    variable x(14) binary
    maximize c*x
    subject to
        A*x <= [2.5;2.5];
        [eye(7),eye(7)]*x <= ones(7,1);
        0<=x<=1;
cvx_end
```

We obtain the following solution:

- Knapsack 1: Laptop, T-Shirt, Weight: 2.5.
- Knapsack 2: Swim.Trunks, Sunglasses, Apples, Opt. Book Weight: 2.2.

The optimal value is 14 and the total number of packed items is 6. The optimal value of the relaxed LP is 14.5, i.e., there is an integrality gap of 0.5.

Problem 4 (Manufacturing Company (optional)):
A manufacturing company plans to build new factories (variables x_1 and x_2) and warehouses (variables x_3 and x_4) in Shenzhen and/or Beijing. The company wants to solve the following binary integer program to determine the location and number of the potential factories and warehouses:

$$\begin{array}{lll} \text{maximize} & 9x_1 + 5x_2 + 6x_3 + 4x_4 & \\ \text{subject to} & 6x_1 + 3x_2 + 5x_3 + 2x_4 & \leq 10 \\ & x_3 + x_4 & \leq 1 \\ & x_3 - x_1 & \leq 0 \\ & x_4 - x_2 & \leq 0 \\ & x_1, x_2, x_3, x_4 & \in \{0, 1\}. \end{array}$$

a) Discuss and interpret the meaning of the constraints "$x_3 + x_4 \leq 1$", "$x_3 - x_1 \leq 0$", and "$x_4 - x_2 \leq 0$".

b) Use the branch-and-bound method to solve the integer problem. You are allowed to use an LP solver to solve each of the relaxed linear programs. Please specify the branch-and-bound tree and what you did at each node

Solution :

a) The condition $x_3 + x_4 \leq 1$ means that either none or exactly one warehouse is built in Shenzhen or Beijing. The condition "$x_3 - x_1 \leq 0$" implies that a factory is built in Shenzhen ($x_1 = 1$) if a warehouse is built in Shenzhen ($x_3 = 1$). Similarly, the constraint "$x_4 - x_2 \leq 0$" implies that a factory is built in Beijing ($x_2 = 1$) if a warehouse is built in Beijing ($x_4 = 1$), i.e.,

$$x_3 = 1 \implies x_1 = 1 \quad \text{and} \quad x_4 = 1 \implies x_2 = 1.$$

b) The optimal solution of the relaxed LP is attained at $(x_1, x_2, x_3, x_4) = (0.8333, 1, 0, 1)$ with optimal value 16.5. This means that the optimal function value of the integer program needs to be less or equal than 16.

We branch on $x_1 = 0.8333$. We consider the two branches:

- (S1): $x_1 = 0$.
- (S2): $x_1 = 1$.

For (S1), the constraint "$x_3 - x_1 \leq 0$" immediately implies $x_3 = 0$ and we can choose $x_2 = x_4 = 1$. Hence, the solution of (S1) is $(0, 1, 0, 1)^\top$ with optimal value 9. This is an integer solution and we obtain the lower bound 9.

For (S2), the optimal solution is given by $(1, 0.8, 0, 0.8)^\top$ with optimal value 16.2.

We need to further branch on x_2. We consider the two branches:

- (S3): $x_2 = 0$.
- (S4): $x_2 = 1$.

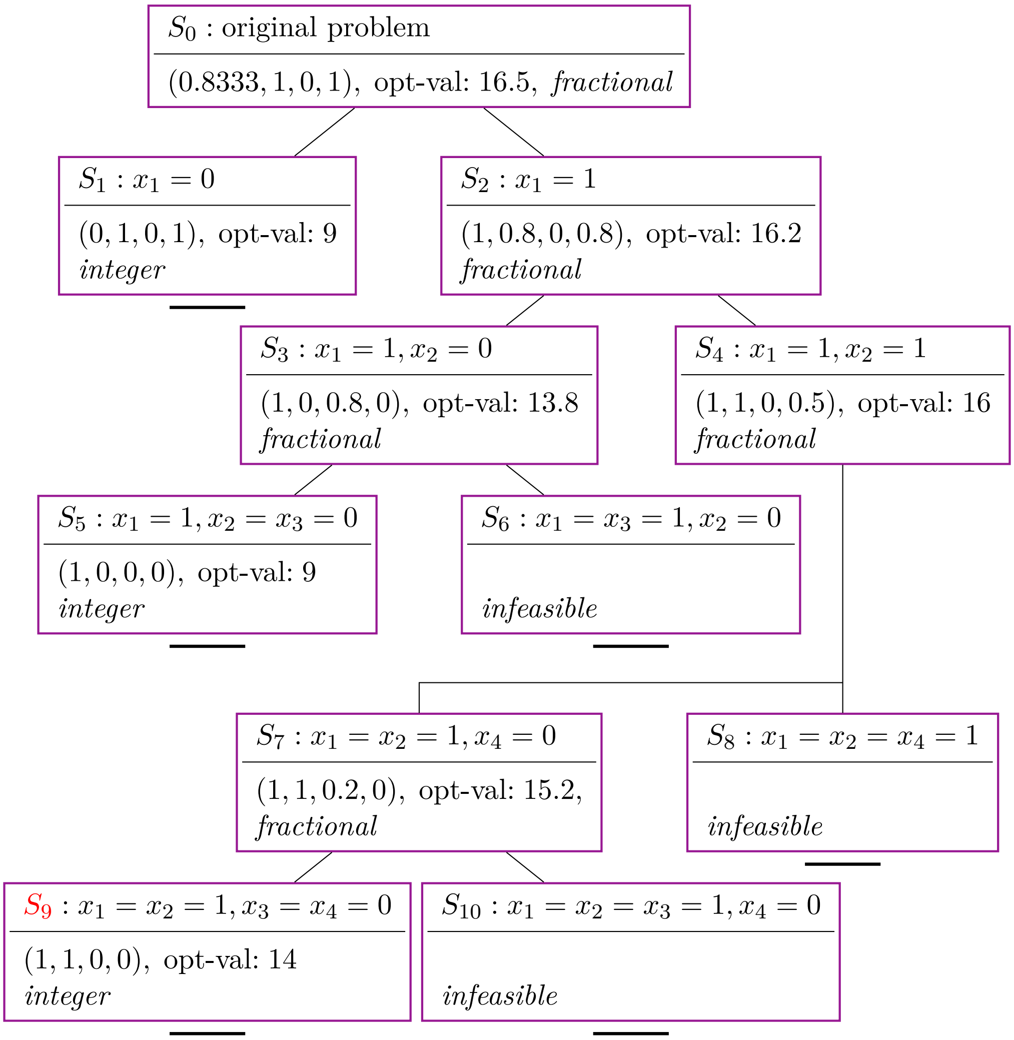
As before, the constraint $x_2 = 0$ implies $x_4 = 0$ and the solution of (S3) is $(1, 0, 0.8, 0)^\top$ with optimal value 13.8. We continue branching on x_3:

- (S5): $x_3 = 0$.
- (S6): $x_3 = 1$.

We immediately see that (S6) is infeasible. For (S5), the only feasible point is $(1, 0, 0, 0)^\top$ and the corresponding function value is 9. This point does not improve the current lower bound. We continue with (S4). The optimal solution is $(1, 1, 0, 0.5)^\top$ with optimal value 16. We continue branching on x_4:

- (S7): $x_4 = 0$.
- (S8): $x_4 = 1$.

(S8) is infeasible. The optimal solution of (S7) is given by $(1, 1, 0.2, 0)^\top$ with optimal value 15.2. Final branching on x_3 yields the points $(1, 1, 0, 0)^\top$ (with optimal value 14) and $(1, 1, 1, 0)^\top$ (which is infeasible). Consequently, $(1, 1, 0, 0)^\top$ is the optimal solution. The complete branching tree is given below.


S_0 : original problem
$(0.8333, 1, 0, 1)$, opt-val: 16.5, *fractional*
$S_1 : x_1 = 0$
$(0, 1, 0, 1)$, opt-val: 9
integer
$S_2 : x_1 = 1$
$(1, 0.8, 0, 0.8)$, opt-val: 16.2
fractional
$S_3 : x_1 = 1, x_2 = 0$
$(1, 0, 0.8, 0)$, opt-val: 13.8
fractional
$S_4 : x_1 = 1, x_2 = 1$
$(1, 1, 0, 0.5)$, opt-val: 16
fractional
$S_5 : x_1 = 1, x_2 = x_3 = 0$
$(1, 0, 0, 0)$, opt-val: 9
integer
$S_6 : x_1 = x_3 = 1, x_2 = 0$
infeasible
$S_7 : x_1 = x_2 = 1, x_4 = 0$
$(1, 1, 0.2, 0)$, opt-val: 15.2,
fractional
$S_8 : x_1 = x_2 = x_4 = 1$
infeasible
$S_9 : x_1 = x_2 = 1, x_3 = x_4 = 0$
$(1, 1, 0, 0)$, opt-val: 14
integer
$S_{10} : x_1 = x_2 = x_3 = 1, x_4 = 0$
infeasible