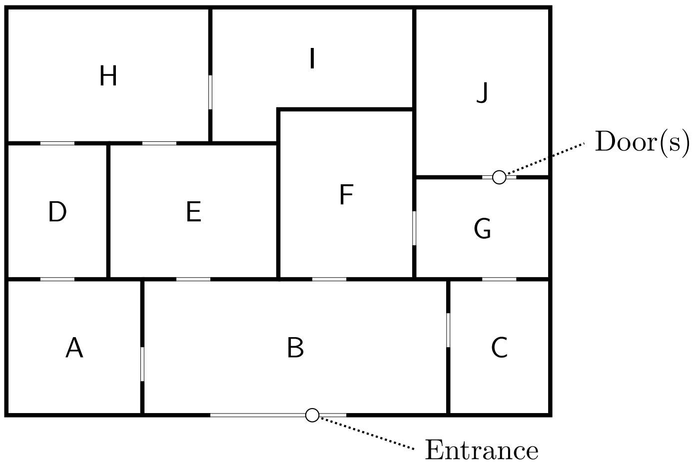
MAT 3007 — Optimization

Solutions 9

Problem 1 (Museum Guards): (approx. *50* points)
A museum director must decide how many guards should be employed to control a newly opened wing of a museum. Budget cuts have forced the director to station guards at the doors between rooms — guarding two rooms at once. A map of the wing and different rooms (A—J) is shown below:



a) Formulate an integer optimization problem to minimize the total number of required guards (while each of the rooms is controlled by at least one guard).

   Describe and explain your optimization model briefly.

b) Implement and solve your model using `CVX`, `MATLAB`, or `Python`. How many museum guards are needed? What is the solution of the LP relaxation of your model?

Solution :

a) For each of the doors, we introduce a binary variable $x_i \in \{0, 1\}$ where 1 indicates that we place a guard at this door. We define:

   – x_1: door between room A and B; x_2: entrance to room B; x_3: door between room B and C;

   – x_4: door between room D and A; x_5: door between room E and B; x_6: door between room F and B; x_7: door between room G and F; x_8: door between room G and C;

   – x_9: door between room H and D; x_{10}: door between room H and E; x_{11}: door between room I and H; x_{12}: door between room J and G;

   For each room, we introduce the respective constraints:

$$
\begin{array}{llll}
\text{Room A:} & x_1 + x_4 \geq 1 & \text{Room B:} & x_1 + x_2 + x_3 + x_5 + x_6 \geq 1 \\
\text{Room C:} & x_3 + x_8 \geq 1 & \text{Room D:} & x_4 + x_9 \geq 1 \\
\text{Room E:} & x_5 + x_{10} \geq 1 & \text{Room F:} & x_6 + x_7 \geq 1 \\
\text{Room G:} & x_7 + x_8 + x_{12} \geq 1 & \text{Room H:} & x_9 + x_{10} + x_{11} \geq 1 \\
\text{Room I:} & x_{11} \geq 1 & \text{Room J:} & x_{12} \geq 1
\end{array}
$$

The full optimization model is then given by:

$$
\begin{array}{lll}
\min_{x \in \mathbb{R}^{12}} \sum_{i=1}^{12} x_i & \text{subject to} & x_i \in \{0,1\} \quad \forall\, i \in \{1,\dots,12\} \\
& & x_1 + x_4 \geq 1, \quad x_1 + x_2 + x_3 + x_5 + x_6 \geq 1 \\
& & x_3 + x_8 \geq 1, \quad x_4 + x_9 \geq 1 \\
& & x_5 + x_{10} \geq 1, \quad x_6 + x_7 \geq 1 \\
& & x_7 + x_8 + x_{12} \geq 1, \quad x_9 + x_{10} + x_{11} \geq 1 \\
& & x_{11} \geq 1, \quad x_{12} \geq 1.
\end{array}
$$

b) The `MATLAB` code to solve the LP relaxation is shown below:

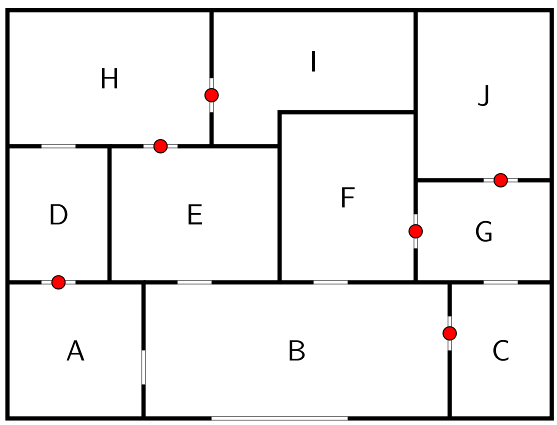
```
A = [1,0,0,1,0,0,0,0,0,0,0,0;
     1,1,1,0,1,1,0,0,0,0,0,0;
     0,0,1,0,0,0,0,1,0,0,0,0;
     0,0,0,1,0,0,0,0,1,0,0,0;
     0,0,0,0,1,0,0,0,0,1,0,0;
     0,0,0,0,0,1,1,0,0,0,0,0;
     0,0,0,0,0,0,1,1,0,0,0,1;
     0,0,0,0,0,0,0,0,1,1,1,0;
     0,0,0,0,0,0,0,0,0,0,1,0;
     0,0,0,0,0,0,0,0,0,0,0,1];
 b = ones(10,1);

 x = linprog(ones(12,1),-A,-b,[],[],zeros(12,1),ones(12,1));
```

The returned solution of this problem is:

$$x_1, x_2, x_5, x_6, x_8, x_9 = 0, \quad x_3, x_4, x_7, x_{10}, x_{11}, x_{12} = 1.$$

Since this solution is binary, it is also a solution of the original binary optimization problem. We can visualize this solution as follows:



In total, 6 museum guards are needed. (We also see that the solution is not unique as several guards can be moved without violating the constraints). Since the LP relaxation already solves the problem, a separate implementation of the integer model is not necessary.

Problem 2 (Branch-and-Bound Method): (approx. *50* points)

Use the branch-and-bound method to solve the following integer program:

$$\begin{array}{lll} \text{maximize} & 17x + 12y & \\ \text{subject to} & 10x + 7y & \leq 40 \\ & x + y & \leq 5 \\ & x, y & \geq 0 \\ & x, y \in \mathbb{Z}. & \end{array}$$

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

Solution : The optimal solution of the relaxed LP is attained at $(x, y) = (1.666, 1.333)$ with optimal value 68.333. This means that the optimal function value of the integer program needs to be less or equal than 68.

We branch on $x = 1.666$. We consider the two branches:

- (S1): $x \leq 1$.
- (S2): $x \geq 2$.

For (S1), the solution of the LP relaxation is given by $(1, 4)^\top$ with objective value 65. This is an integer solution and we obtain the lower bound 65.

For (S2), the optimal solution is given by $(2, 2.857)^\top$ with optimal value 65.

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

- (S3): $y \leq 2$.
- (S4): $y \geq 3$.

We immediately see that (S4) is infeasible. For (S3), the optimal solution is $(2.6.2)^\top$ and the corresponding function value is 68.2.

We continue branching on $x = 2.6$:

- (S5): $x \leq 2 \implies x = 2$.
- (S6): $x \geq 3$.

The optimal solution of the LP relaxation of (S5) is now $(2, 2)^\top$ with optimal value 58. This is an integer solution; however, the optimal value is lower than the current lower bound. The solution of (S6) is given by $x = 3$ and $y = 1.429$ with objective value 68.14.

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

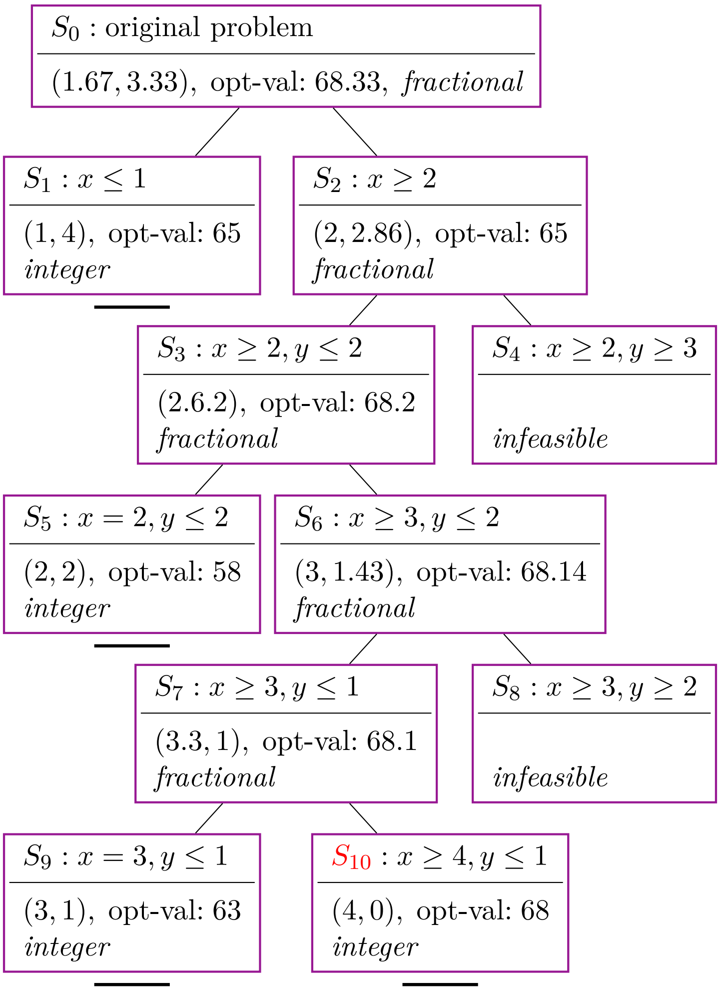
- (S7): $y \leq 1$.
- (S8): $y \geq 2$.

We immediately see that (S8) is infeasible. For (S7), the optimal solution is $(3.3, 1)^\top$ and the corresponding function value is 68.1.

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

- (S9): $x \geq 3 \implies x = 3$.
- (S10): $x \geq 4$.

The solution of the LP relaxation of (S9) is given by $(3,1)^\top$ with optimal value 63. For (S10), we obtain the solution $(4,0)^\top$ with objective function value 68. Hence, $(4,0)^\top$ is the optimal solution of the problem and we can stop here. A complete picture of the procedure given as follows:



Problem 3 (Multiple Knapsacks):
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	2	1	3	2	1	4	2
Weight	2	0.5	0.5	0.1	0.5	1	1.5

Knapsack 1	Knapsack 2
$C_1 = 3$	$C_2 = 2$

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

Overall the full optimization problem is given by:

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

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:

$$\max_x \; (v^\top, v^\top)x \qquad \text{s.t.} \qquad \begin{bmatrix} a^\top & 0 \\ 0 & a^\top \end{bmatrix} x \leq \begin{bmatrix} 3 \\ 2 \end{bmatrix}, \quad \begin{bmatrix} I & I \end{bmatrix} x \leq \mathbb{1}, \quad x_{ij} \in \{0,1\} \quad \forall\, i,j.$$

The `MATLAB` code can be found below:

```
% generate problem
a = [2,0.5,0.5,0.1,0.5,1,1.5];
v = [2,1,3,2,1,4,2];

A = [a,zeros(1,7);zeros(1,7),a;eye(7),eye(7)];
b = [3;2;ones(7,1)];

lb  = zeros(14,1);
ub  = ones(14,1);

intcon = 1:14;
```

```
s = {' Laptop',' T-Shirt ',' Swim.Trunks',' Sunglasses',' Apples',' Opt.Book','
    Water'};

% integer problem
[x,fval,~,output] = intlinprog([-v';-v'],intcon,A,b,[],[],lb,ub);

bag1 = '';
bag2 = '';

for j = 1:7
    if x(j)
        bag1 = strcat(bag1,s{j},',');
    end
    if x(j+7)
        bag2 = strcat(bag2,s{j},',');
    end
end

fprintf('Knapsack 1: %s Weight: %g\nKnapsack 2: %s Weight: %g\nOptimal value: %g,
    Number of Items: %i\n',bag1,a*x(1:7),bag2,a*x(8:end),-fval,sum(x));

% LP Relaxation
[y,lval]        = linprog([-v';-v'],A,b,[],[],lb,ub);

fprintf('Integrality Gap: %g\n',abs(lval-fval));
```

We obtain the following solution:

- Knapsack 1: Sunglasses, Apples, Water, Weight: 2.1.
- Knapsack 2: T-Shirt, Swim.Trunks, Opt.Book, Weight: 2.

The optimal value is 13 and the total number of packed items is 6. The optimal value of the relaxed LP is 13.9, i.e., there is an integrality gap of 0.9. In particular, the solution of the relaxed LP is not binary and fraction of the same items are packed in knapsack one and two.

Problem 4 (Manufacturing Company):
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:

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

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

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): $x_2 = 1$.

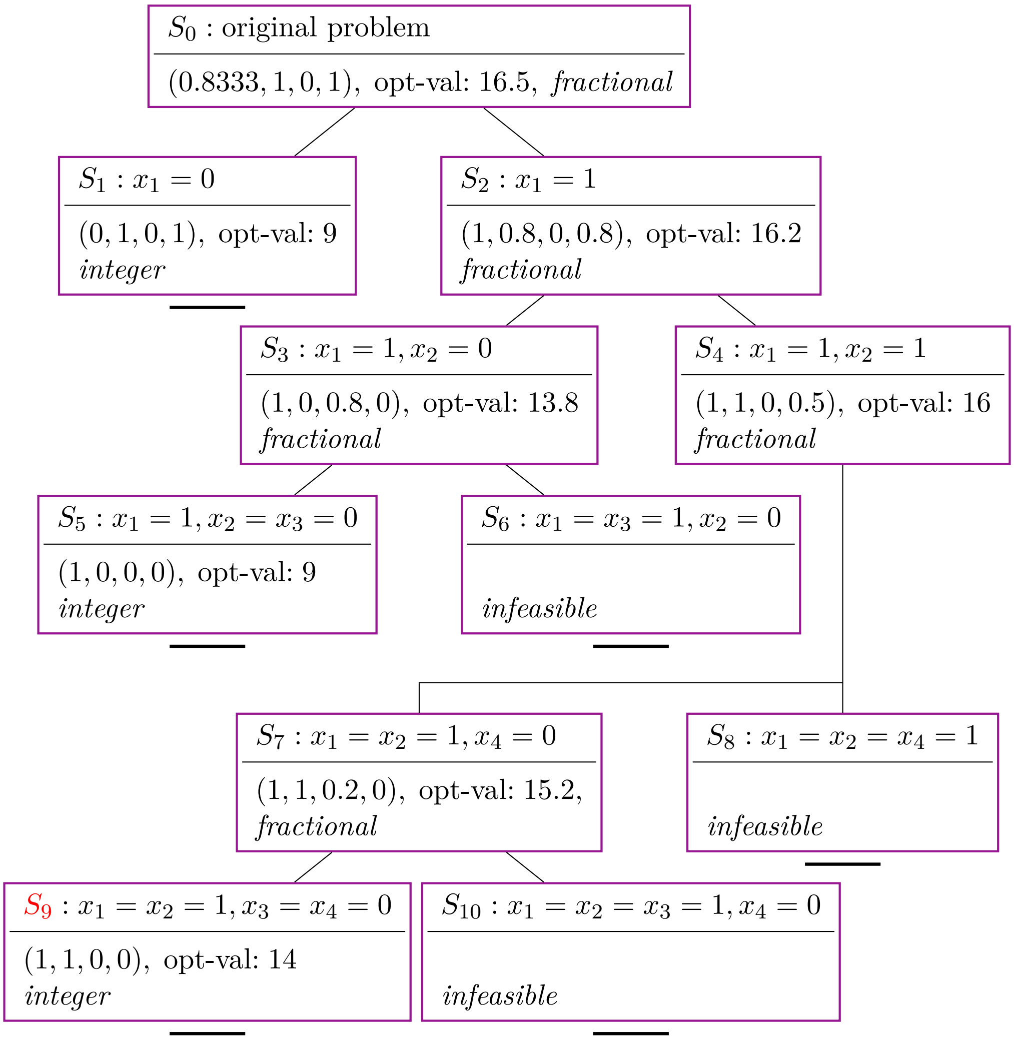
As before, the constraint $x_2 = 0$ implies $x_4 = 0$ and the solution of (S3) is $(1, 0, 0.8, 0)^\top$ with optimal value 13.8. We continue branching on x_3:

- (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