Optimisation & Operations Research

Haide College, Spring Semester

Assignment 4 (5%)

Due: 18 May, 23:59

10 marks

1. Solve the following problem using Branch and Bound.

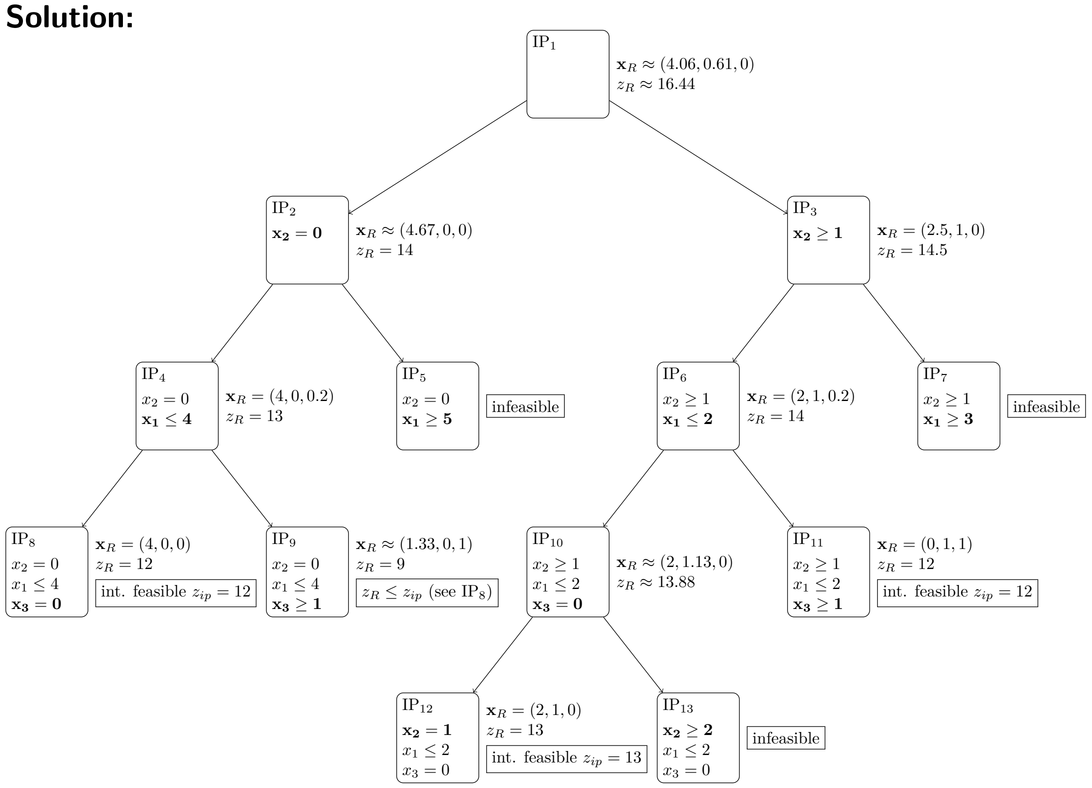
$$\begin{aligned} \max \quad & z = 3x_1 + 7x_2 + 5x_3, \\ \text{subject to} \quad & 2x_1 + 5x_2 + 4x_3 \le 20, \\ & 2x_1 + 8x_2 + 5x_3 \le 13, \\ & 3x_1 + 3x_2 + 10x_3 \le 14, \end{aligned}$$

where $x_1, x_2, x_3 \ge 0$ and integer.

Perform the branch and bound procedure as described in the course notes, and illustrate this procedure with a tree diagram. Please refer to the above ILP as "IP1" and format your solution as per the parts below to facilitate marking. Note these parts are interconnected (you need to fathom leaf nodes to draw the tree diagram, for example).

5 a) Draw a tree diagram using the branching strategy described in the course notes. Clearly indicate the extra constraints due to branching at each new leaf node on you diagram. Include the optimal solution (variables and objective) of each relaxed problems either in your tree diagram or in a separate table.

Solution:



.

Solutions to the relaxed problems alternatively presented as a table.

Problem	$\mathbf{x}_R$	z_R
IP1	(4.06;0.61;0)	16.44
IP2	(4.67;0;0)	14
IP3	(2.5;1;0)	14.5
IP4	(4;0;0.2)	13
IP5	infeasible	-
IP6	(2;1;0.2)	14
IP7	infeasible	-
IP8	(4;0;0)	12
IP9	(1.33;0;1)	9
IP10	(2;1.13;0)	13.88
IP11	(0;1;1)	12
IP12	(2;1;0)	13
IP13	infeasible	-

1 b) Include documented MATLAB code to solve each relaxed LP.

Solution:

```
%% Define the constraints and objective for this LP
A = [2 5 4;2 8 5;3 3 10];
b = [20;13;14];
c = [3 7 5];

%% Solve the original relaxed LP and store result
A4_IP1 = optimproblem;
x = optimvar('x',3,'LowerBound',0);
A4_IP1.Objective = c*x;
A4_IP1.ObjectiveSense = 'maximize';
A4_IP1.Constraints.original = A*x<=b;
[sol,fval] = solve(A4_IP1);
output = {sol.x,fval};

%% Branch from IP1, solve with added constraint x2=0
A4_IP2 = A4_IP1;
A4_IP2.Constraints.IP2 = x(2)<=0;
[sol,fval] = solve(A4_IP2);
output = [output;{sol.x,fval}];

%% Branch from IP1, solve with added constraint x2>=1
A4_IP3 = A4_IP1;
A4_IP3.Constraints.IP3 = x(2)>=1;
[sol,fval] = solve(A4_IP3);
output = [output;{sol.x,fval}];

%% Branch from IP2, solve with added constraint x1<=4
A4_IP4 = A4_IP2;
A4_IP4.Constraints.IP4 = x(1)<=4;
[sol,fval] = solve(A4_IP4);
output = [output;{sol.x,fval}];

%% Branch from IP2, solve with added constraint x1>=5
```

```
A4_IP5 = A4_IP2;
A4_IP5.Constraints.IP5 = x(1)>=5;
[sol,fval] = solve(A4_IP5);
output = [output;{sol.x,fval}];

%% Branch from IP3, solve with added constraint x1<=2
A4_IP6 = A4_IP3;
A4_IP6.Constraints.IP6 = x(1)<=2;
[sol,fval] = solve(A4_IP6);
output = [output;{sol.x,fval}];

%% Branch from IP3, solve with added constraint x1>=3
A4_IP7 = A4_IP3;
A4_IP7.Constraints.IP7 = x(1)>=3;
[sol,fval] = solve(A4_IP7);
output = [output;{sol.x,fval}];

%% Branch from IP4, solve with added constraint x3=0
A4_IP8 = A4_IP4;
A4_IP8.Constraints.IP8 = x(3)<=0;
[sol,fval] = solve(A4_IP8);
output = [output;{sol.x,fval}];

%% Branch from IP4, solve with added constraint x3>=1
A4_IP9 = A4_IP4;
A4_IP9.Constraints.IP9 = x(3)>=1;
[sol,fval] = solve(A4_IP9);
output = [output;{sol.x,fval}];

%% Branch from IP6, solve with added constraint x3=0
A4_IP10 = A4_IP6;
A4_IP10.Constraints.IP10 = x(3)<=0;
[sol,fval] = solve(A4_IP10);
output = [output;{sol.x,fval}];

%% Branch from IP6, solve with added constraint x3>=1
A4_IP11 = A4_IP6;
A4_IP11.Constraints.IP11 = x(3)>=1;
[sol,fval] = solve(A4_IP11);
output = [output;{sol.x,fval}];

%% Branch from IP10, solve with added constraint x2=1
A4_IP12 = A4_IP10;
A4_IP12.Constraints.IP12 = x(2)<=1;
[sol,fval] = solve(A4_IP12);
output = [output;{sol.x,fval}];

%% Branch from IP10, solve with added constraint x2>=2
A4_IP13 = A4_IP10;
A4_IP13.Constraints.IP13 = x(2)>=2;
[sol,fval] = solve(A4_IP13);
output = [output;{sol.x,fval}];
```

Note that since you must show all branches of the tree fathomed completely, you cannot solve the whole problem using MATLAB's `intlinprog`. It may be a good idea to check the solution that way.

3 c) For each *fathomed* nodes of the tree clearly explain why the node is fathomed. Present

this information as a list of fathomed nodes, along with the reason they are fathomed.

> **Solution:** Here is a list of fathomed nodes, along with the reason they are fathomed.
>
> - IP_5, IP_7 and IP_{13}: relaxed problem is infeasible
> - IP_8, IP_{11} and IP_{12}: integer feasible solution
> - IP_9: bound by integer feasible solution, $z_R = 9$ which is less than $z_{ip} = 12$ from IP_8

1 d) State the optimal solution to the ILP.

> **Solution:** The optimal solution is given in IP_12, whose relaxation is integer feasible with the largest value of the objective. This is
>
> $$\mathbf{x}^* = (2, 1, 0), \quad z^* = 13.$$

---

10 marks

2. Recall the following knapsack problem from Assignment 3.

Consider the following linear program representing a knapsack problem for six items with total value z and maximum volume 35.

$$\begin{array}{llllllllllllll} \text{max.} & z & = & 19x_1 & + & 22x_2 & + & 30x_3 & + & 37x_4 & + & 11x_5 & + & 42x_6, \\ \text{s.t.} & & & 7x_1 & + & 6x_2 & + & 11x_3 & + & 13x_4 & + & 4x_5 & + & 13x_6 \quad \leq \quad 35, \end{array}$$

with $x_i = 0, 1$, for $i = 1, \dots, 6$.

1 a) Relax the integral constraints (to give the constraint that all the x_i are $0 \leq x_i \leq 1$) and solve the above problem using the problem based approach in MATLAB. Upload your code to MATLAB Grader for checking, you do not need to include it in your PDF submission.

> **Solution:**
>
> ```
> %% Define the objective and constraint
> f = [19 22 30 37 11 42];
> A = [7 6 11 13 4 13];
> b = 35;
>
> %% Solve the original relaxed LP
> IP1 = optimproblem;
> x = optimvar('x',6,'LowerBound',0,'UpperBound',1);
> IP1.Objective = f*x;
> IP1.ObjectiveSense = 'maximize';
> IP1.Constraints.original = A*x<=b;
> [sol,fval] = solve(IP1);
> ```

8 b) Find the optimal solution to the problem using the branch and bound as described in the course notes.

Since the tree for this problem is a large, so you should draw this diagram to show only the structure of the tree and the labels of the problems.

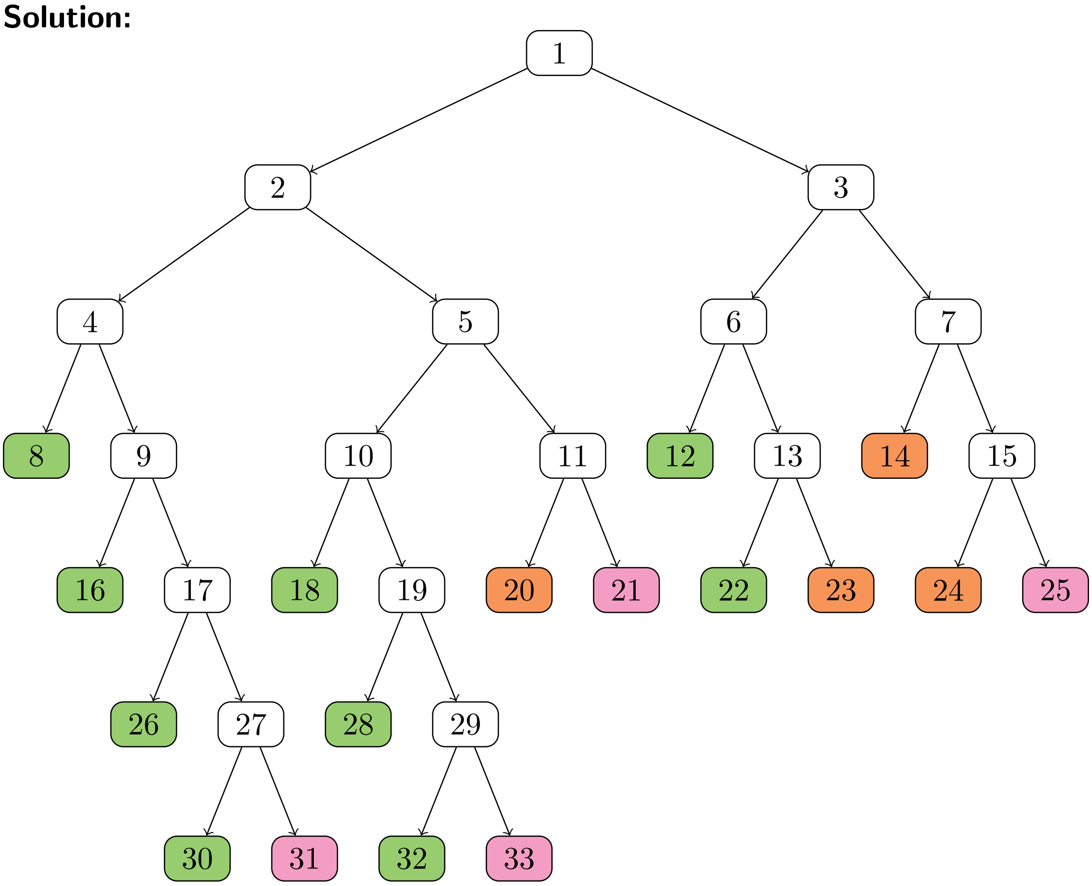
The detail of the relaxed problems and branching should then be provided in a table like the one below. Make sure you include reasons for fathoming.

It is **not** required that you submit any MATLAB code for this question, you just need to summarise results in your table.

Problem	Parent	New constraint	z_R	$\mathbf{x}_R$	Branched/fathomed	Reason for fathoming
1	-	-	...	...	...	...
2	1	...	...	...	...	...
3	1	...	...	...	...	...

Hint: To solve the relaxed problems in this question it is strongly recommended that you use the "problem-based" approach in MATLAB (see Practical 3 for details).

Solution:



In the above tree diagram the fathomed nodes are coloured as follows;

- green: integer feasible solution
- pink: infeasible
- orange: bound by integer feasible solution ($z_R \leq z_{ip}$)

Here is the table summarising the branch and bound procedure:

IP	Parent	Constraint	z_R	$\mathbf{x}_R$	Bran/fath	Reason for fathoming
1	-	-	109.25	(0,1,0,1,0.75,1)	branched	-
2	1	$x_5 = 0$	109.18	(0,1,0.27,1,0,1)	branched	-
3	1	$x_5 = 1$	109.15	(0,1,0,0.92,1,1)	branched	-
4	2	$x_3 = 0$	109.14	(0.43,1,0,1,0,1)	branched	-
5	2	$x_3 = 1$	108.23	(0,1,1,0.38,0,1)	branched	-
6	3	$x_4 = 0$	107.71	(0.14,1,1,0,1,1)	branched	-
7	3	$x_4 = 1$	108.77	(0,1,0,1,1,0.92)	branched	-
8	4	$x_1 = 0$	101	(0,1,0,1,0,1)	fathomed	integer feasible, $z_{ip} = 101$
9	4	$x_1 = 1$	108.62	(1,1,0,0.69,0,1)	branched	-
10	5	$x_4 = 0$	107.57	(0.71,1,1,0,0,1)	branched	-
11	5	$x_4 = 1$	105.15	(0,1,1,1,0,0.38)	branched	-
12	6	$x_1 = 0$	105	(0,1,1,0,1,1)	fathomed	integer feasible, $z_{ip} = 105$
13	6	$x_1 = 1$	107.66	(1,1,0.45,0,1,1)	branched	-
14	7	$x_6 = 0$	102.71	(0.14,1,1,1,1,0)	fathomed	bounded, $z_R < z_{ip}$ (IP12)
15	7	$x_6 = 1$	108.33	(0,0.83,0,1,1,1)	branched	-
16	9	$x_4 = 0$	83	(1,1,0,0,0,1)	fathomed	integer feasible
17	9	$x_4 = 1$	107.08	(1,1,0,1,0,0.69)	branched	-
18	10	$x_1 = 0$	94	(0,1,1,0,0,1)	fathomed	integer feasible
19	10	$x_1 = 1$	106.54	(1,1,1,0,0,0.85)	branched	-
20	11	$x_6 = 0$	102.57	(0.71,1,1,1,0,0)	fathomed	bounded, $z_R < z_{ip}$ (IP12)
21	11	$x_6 = 1$	-	-	fathomed	infeasible
22	13	$x_3 = 0$	94	(1,1,0,0,1,1)	fathomed	integer feasible
23	13	$x_3 = 1$	104.62	(1,1,1,0,1,0.54)	fathomed	bounded, $z_R < z_{ip}$ (IP12)
24	15	$x_2 = 0$	103.64	(0,0,0.45,1,1,1)	fathomed	bounded, $z_R < z_{ip}$ (IP12)
25	15	$x_2 = 1$	-	-	fathomed	infeasible
26	17	$x_6 = 0$	78	(1,1,0,1,0,0)	fathomed	integer feasible
27	17	$x_6 = 1$	105.33	(1,0.33,0,1,0,1)	branched	-
28	19	$x_6 = 0$	71	(1,1,1,0,0,0)	fathomed	integer feasible
29	19	$x_6 = 1$	105.67	(1,0.67,1,0,0,1)	branched	-
30	27	$x_2 = 0$	98	(1,0,0,1,0,1)	fathomed	integer feasible
31	27	$x_2 = 1$	-	-	fathomed	infeasible
32	29	$x_2 = 0$	91	(1,0,1,0,0,1)	fathomed	integer feasible
33	29	$x_2 = 1$	-	-	fathomed	infeasible

1 c) Finally, briefly comment on how effective branch and bound was for this problem. For example, would a brute force search have been more efficient?

Solution: In this branch and bound procedure 33 nodes are explored versus 126 for a exhaustive brute force search. This means that branch and bound is very efficient in terms of restricting the possible solution space. This is possibly not as computationally efficient as it first seems since the relaxed solution must be found at each branch and bound node, versus just evaluating the objective in the exhaustive search.

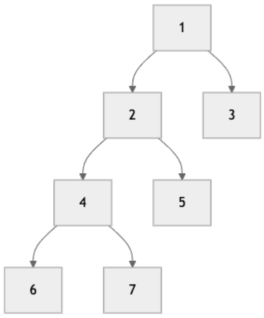
Example solution

Here is an example of the style formatting for Question 2.

Recall in Tutorial 5, Question 2 we performed Branch and Bound on the following problem:

$$\begin{array}{rrcrcrcl}
\max\ z & = & 3x_1 & + & x_2 & + & 3x_3 & \\
 & & -2x_1 & + & 4x_2 & - & 4x_3 & \leq 18 \\
 & & 5x_1 & - & 3x_2 & + & x_3 & \leq 13 \\
 & & -x_1 & + & 2x_2 & + & 3x_3 & \leq 19
\end{array}$$

Here is a minimal tree diagram that would be sufficient for your assignment submission.



The table summarising the information from the relaxed problems, solution procedure and branching/fathoming would be:

Problem	Parent	New constraint	z_R	$\mathbf{x}_R$	Branched/fathomed	Reason for fathoming
1	-	-	43	(8.71,10.85,2)	branched	-
2	1	$x_1 \leq 8$	41.18	(8,9.82,2.45)	branched	-
3	1	$x_1 \geq 9$	-	-	fathomed	infeasible
4	2	$x_3 \leq 2$	40.5	(8,10.5,2)	branched	-
5	2	$x_3 \geq 3$	39	(7.14,8.57,3)	fathomed	$z_r < z_{ip}$ (IP6)
6	4	$x_2 \leq 10$	40	(8,10,2)	fathomed	integer feasible
7	4	$x_2 \geq 11$	-	-	fathomed	infeasible

See provided Live Script `Tute5_example.mlx` on Cloudcampus for an example of how to solve the above relaxed problems with a problem based workflow. The main feature here is that we store information about the solution procedure in a table. The optimsation problems are stored too, which makes is easier to add an extra constraint each time there is a new branched problem.