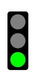
42114: Integer Programming

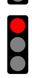
Exercises

Updated: August 18, 2022

Difficulty indicators

 : Easy exercise

 : Moderate exercise

 : Hard exercise

Week 1: Integer Programming

Suggested order:

- Wolsey: Exercise 1.1
- Wolsey: Exercise 1.2 (i) and (ii)
- Modelling Knapsack Problem
- Manpower Planning Problem
- Wolsey: Exercise 1.5

All Wolsey exercises are copied from or inspired by exercises from "Integer Programming" 1st edition by Laurence Wolsey.

Wolsey Exercise 1.1

Suppose that you are interested in choosing a set of investments $\{1, \ldots, 7\}$ using 0-1 variables. Model the following constraints:

(i) You cannot invest in all of them.

(ii) You must choose at least one of them.

(iii) Investment 1 cannot be chosen if investment 3 is chosen.

(iv) Investment 4 can be chosen only if investment 2 is also chosen.

(v) You must either choose both investments 1 and 5 or choose neither.

(vi) You must choose either at least one of the investments 1, 2 and 3 or at least two investments from 2, 4, 5 and 6.

Wolsey Exercise 1.2

Formulate the following as mixed integer programs, both with x_1 and x_2 as constant parameters and then with x_1 and x_2 as decision variables:

(i) $u = \min\{x_1, x_2\}$, assuming that $0 \leq x_j \leq C$ for $j = 1, 2$

(ii) $v = |x_1 - x_2|$ with $0 \leq x_j \leq C$ for $j = 1, 2$

Modelling Knapsack Problems

In the first lecture we saw how we modelled the 0-1-Knapsack Problem. Remember in the knapsack problem we have a set of n items numbered from 1 to n. Each of these items has a weight a_i and an expected return (profit) c_i. Finally there is a budget or capacity of the knapsack b that cannot be exceeded.

For this problem we define a binary variable x_i for each item i. This variable is 1 if the item is in the solution (i.e. in the knapsack) and 0 otherwise.

The model will look like:

$$\begin{aligned} \max \quad & \sum_{i=1}^{n} c_i x_i & & \\ \text{s.t.} \quad & \sum_{i=1}^{n} a_i x_i & \leq \quad & b \\ & x_i & \in \quad & \{0,1\} \quad \forall i = 1...n \end{aligned} \tag{1}$$

There exist other versions or variants of the knapsack problem. In the following questions a variant of the 0-1-knapsack problem is presented. For each of these give a mathematical model for the variant:

1. **The Subset Sum Problem:** In the subset sum problem, n items are given each with positive integer weights $a_1, ..., a_n$ and **no** profit is given. Now find the subset of items such that the corresponding total weight is maximized without exceeding the capacity b.

2. **Precedence Constrained Knapsack Problem:** In the precedence constrained knapsack problem a set A of pairs of items is given. If the pair (i_1, i_2) is in A it means that item i_2 can only be in the knapsack if item i_1 is also in the knapsack (so i_1 "precedes" i_2). Still it is okay for item i_1 to be in the knapsack while i_2 is not in the knapsack. Now this has to be modelled into the problem.

3. **Multiple Knapsack Problem:** In the multiple knapsack problem, there are several knapsacks $j = 1, ..., m$ where we can put items into. Each has its own capacity b_j. Now the n items have to be arranged in m knapsacks (or being left outside) in order to maximize profit.

4. **Multiple Choice Knapsack Problem:** In the multiple choice knapsack problem each item i exists in different variants N_i. So if $l \in N_i$ then l is a variant of item i. Each variant has a different weight and a different profit. Now we need to choose exactly one variant of each item to put into the knapsack, while maximizing profit.

Manpower Planning Problem

In this strategic manpower planning problem we consider a strategic assessment of the demand of manpower for an organisation. Consider an organisation that is open 7 days a week with 1 shift a day.

- The number of employees needed varies from day to day but is constant on a weekly basis (we will call those numbers b_1 for Monday, b_2 for Tuesday etc.). So every Monday needs the same number of manpower, every Tuesday is identical wrt. manpower etc., but there might be daily differences (i.e. needing more manpower on Monday than on Tuesday).
- All employees must work 5 consecutive days and have two days off.
- The objective is to minimize the number of employees and find out when they have to work.

1. Formulate a mathematical model that solves the strategic manpower planning problem as described above.

2. Solve the problem for $b_1 = 10, b_2 = 5, b_3 = 10, b_4 = 5, b_5 = 10, b_6 = 5, b_7 = 10$.

3. Solve the problem for $b_1 = 8, b_2 = 8, b_3 = 8, b_4 = 8, b_5 = 8, b_6 = 8, b_7 = 7$.

4. How can you compute the unused number of man days? Calculate the number of unused days for the two examples above.

5. The labour costs might vary depending upon the days off. How can we change the objective function to incorporate that?

Wolsey Exercise 1.5

John Dupont is attending a summer school where he must take four courses per day. Each course lasts an hour, but because of the large number of students, each course is repeated several times per day by different teachers. Section i of course k denoted (i, k) meets at the hour t_{ik}, where courses start on the hour between 10 a.m. and 7 p.m. John's preferences for when he takes courses are influenced by the reputation of the teacher, and also the time of day. Let p_{ik} be his preference for section (i, k). Unfortunately, due to conflicts, John cannot always choose the sections he prefers.

(i) Formulate an integer program to choose a feasible course schedule that maximizes the sum of John's preferences.

(ii) Modify the formulation, so that John never has more than two consecutive hours of classes without a break.

(iii) Modify the formulation, so that John chooses a schedule in which he starts his day as late as possible.

Week 2: Formulations

Suggested order:

- NASA Project Problem
- A Container Problem
- IP Exam 2015: Question 2
- IP Exam 2013: Question 3.2 and 3.3

NASA Capital Budget Model

This exercise originally featured as an example of greedy heuristics in Hillier and Lieberman "Introduction to Operations Research".

The North American Space Administration (NASA) need to decide which missions and operations to carry out for the period from 2022 and 25 years ahead. Each mission will span all 25 years and have certain costs in each of the subperiods of 5 years. The administration has a limited budget for missions.

The table below shows the 14 missions that potentially can be supported, their costs during each of the five 5-year periods and finally a value which sum should be maximised over all missions.

		Expenses (USD billion					
j	Mission	2022-2026	2027-2031	2032-2036	2037-2041	2042-2046	Value
1	Communication sat.	6					200
2	Orbital Microwave	2	3				3
3	Io Lander	3	5				20
4	Uranus Probe I.					10	50
5	Uranus Probe II.		5	8			70
6	Mercury Probe			1	8	4	20
7	Saturn Probe	1	8				5
8	Infrared Imaging				5		10
9	Ground-based SETI	4	5				200
10	Large Orbital Structs		8	4			150
11	Color Imaging			2	7		18
12	Medical Technology	5	7				8
13	Polar Orbital Platform		1	4	1	1	300
14	Geosynchronus SETI		4	5	3	3	185
	Budget	10	12	14	14	14	

As can be seen, we have a limited budget for each five year period, and if I eg. choose to carry out mission number 14, the geosynchronus SETI, it has no expenses in the first period, costs four billion in the second period, 5 billion in the third period, 3 billion in the fourth period and 3 billion in the fifth period for an accumulated cost of 15 billion over the 5 periods. In each five-year period, the sum of the costs in that period, for all missions carried out, cannot exceed the budget of that five year period.

In addition, there are some extra constraints:

- Certain missions are not compatible, that is, both of them cannot be carried out at the same time. The incompatible projects are 4 and 5, 8 and 9, and 11 and 14.
- Furthermore, the missions 4 to 7 are dependent on that mission 3 is carried out. And also mission 11 cannot be carried out without mission 2 being executed.

1. Make an integer program (actually a binary program) for the NASA Capital budget problem as stated above. You only need one variable for each mission. Let x_i for mission i be 1 if you choose to carry out the mission and 0 otherwise.
2. Implement your model in Julia. Solve the problem to optimality and report the optimal solution.

A Container Problem

We have n containers of various sizes $v_1, v_2, ..., v_n$. There is liquid in the containers. The volume of the liquid in each of the containers is $a_i > 0$, $i = 1, 2, ..., n$. Now, we want to combine the contents of the containers so that we get the fewest number of (non-empty) containers in the end. During the combination, we must adhere to the following rules:

- A liquid **cannot** be split into two (or more) containers.
- If a container is used at the end, the liquid that was originally in it, must remain in it.

1. Make a methematical model for the problem.
2. Given the values $n = 6$, $v = (500, 500, 300, 300, 200, 100)$, and $a = (350, 300, 200, 150, 100, 100)$, first solve solve the model in Julia with all your variables being continuous, then solve the model again in Julia developed with your appropriate variable definitions. Comment on the two values.

IP Exam 2015: Question 2 (15%)

The production schedule of a factory machine has to be determined. The machine is producing two products A and B. Demands for the two products are given over a 5-day period (the five working days of a week). As setting up the machine for producing either product A or B is very time-consuming, only one of the products is produced at any given day. For each day in the 5-day period we need to determine a production schedule showing which product is being produced and in what quantity.

Let d_t^A and d_t^B be the demands for products A and B, respectively, on day t, where $t = 1, \ldots, 5$. The machine is set-up for the production every morning. The set-up cost for each product is constant and is denoted f_A for product A and f_B for product B. Production is limited for product A to l_A units a day and respectively l_B units a day for product B. Storage cost for each unit is dependent on the day of the week and is denoted s_t^A and s_t^B. The objective is to determine a production schedule with minimum cost.

Question 2.1

Formulate the problem as a Mixed Integer Programming problem. Assume that the initial and final inventory of both products are 0 units.

Question 2.2

The factory machine is upgraded such that the set-up of the machine for a certain product is not required if the machine produces the same product as the previous day. So if product A is produced on eg. three consecutive days, the set-up cost is only being paid on the first day. Modify your model to take the machine upgrade into account. Since we do not know the set-up before day 1, we always pay the set-up cost on the first day.

IP Exam 2013: Question 3 (15%)

Given a $0-1$ knapsack set $K = \{x \in \{0,1\}^n : \sum_{i=1}^{n} a_i x_i \leq b\}$. Recall that a subset C of indices is a *minimal cover* if $\sum_{i \in C} a_i > b$ and $\sum_{i \in C \setminus \{j\}} a_i \leq b$ for every $j \in C$. That is, the knapsack cannot contain all items in C, but every proper subset of C can be loaded.

Now consider the following knapsack set: $K^1 = \{x \in \{0,1\}^4 : 6x_1 + 6x_2 + 6x_3 + 5x_4 \leq 16\}$

Subquestion 3.1

Find the minimal cover formulation K^C for the knapsack set K^1, so find the set of all minimal covers and write up the formulation K^1.

The minimal covers for K^1 are $\{1,2,3\}, \{1,2,4\}, \{1,3,4\}, \{2,3,4\}$. Translated into the constraints that will form K^C we get:

$$\begin{aligned} x_1 + x_2 + x_3 &\leq 2 \\ x_1 + x_2 + x_4 &\leq 2 \\ x_1 + x_3 + x_4 &\leq 2 \\ x_2 + x_3 + x_4 &\leq 2 \end{aligned}$$

So the **formulation** according to the definition of Wolsey will be:

$$\begin{aligned} K^C = \{x \in [0,1]^4 : x_1 + x_2 + x_3 &\leq 2 \\ x_1 + x_2 + x_4 &\leq 2 \\ x_1 + x_3 + x_4 &\leq 2 \\ x_2 + x_3 + x_4 &\leq 2\} \end{aligned}$$

So note since this is a formulation the variables are not defined as binary.

Subquestion 3.2

Let K' be the formulation based on K, that is, $K' = \{x \in R^4 : 0 \leq x \leq 1, 6x_1 + 6x_2 + 6x_3 + 5x_4 \leq 16\}$. Is $K^C \subseteq K'$ or vice versa. Explain.

Subquestion 3.3

For this instance is the minimal cover formulation a *better* formulation. Explain?

Week 3: Relaxations

Suggested order:

- Relaxation Exercises
- Wolsey: Chapter 2, Exercise 3
- Wolsey: Chapter 2, Exercise 4
- Wolsey: Chapter 2, Exercise 7

All Wolsey exercises are copied from or inspired by exercises from ”Integer Programming” 1st edition by Laurence Wolsey.

Relaxation Exercises

Exercise 1

For the integer programming model

$$
\begin{array}{llrcl}
(P_1) & \max & 3x_1 + 4x_2 + 5x_3 & & \\
& & & & \\
& \text{s.t.} & 2x_1 + x_2 + 4x_3 & \leq & 14 \\
& & x_2 - 4x_3 & \geq & 4 \\
& & \multicolumn{3}{l}{x_1, x_2, x_3 \geq 0 \text{ and integer}}
\end{array}
$$

1. Argue that P_2 is a relaxation of P_1, where P_2 is defined by

$$
\begin{array}{llrcl}
(P_2) & \max & 3x_1 + 4x_2 + 5x_3 & & \\
& & & & \\
& \text{s.t.} & 2x_1 + x_2 + 4x_3 & \leq & 14 \\
& & x_2 - 4x_3 & \geq & 4 \\
& & \multicolumn{3}{l}{x_1, x_2, x_3 \geq 0}
\end{array}
$$

   What is the name of this relaxation?

2. Argue that P_3 is a relaxation of P_1, where P_3 is defined by

$$
\begin{array}{llrcl}
(P_3) & \max & 3x_1 + 4x_2 + 5x_3 & & \\
& & & & \\
& \text{s.t.} & 2x_1 + x_2 + 4x_3 & \leq & 14 \\
& & \multicolumn{3}{l}{x_1, x_2, x_3 \geq 0 \text{ and integer}}
\end{array}
$$

3. Argue that P_4 is a relaxation of P_1, where P_4 is defined by

$$
\begin{array}{llrcl}
(P_4) & \max & 5x_1 + 8x_2 + 5x_3 & & \\
& & & & \\
& \text{s.t.} & 2x_1 + x_2 + 4x_3 & \leq & 14 \\
& & x_2 - 4x_3 & \geq & 4 \\
& & \multicolumn{3}{l}{x_1, x_2, x_3 \geq 0}
\end{array}
$$

   Is P_4 a relaxation of P_2? Is P_4 a relaxation of P_3?

Exercise 2

P_5 and P_6 are defined as

$$
\begin{array}{llrcl}
(P_5) & \max & 5x_1 + 8x_2 + 5x_3 & & \\
& & & & \\
& \text{s.t.} & x_1 + x_2 + x_3 & \leq & 14y \\
& & \multicolumn{3}{l}{x_1, x_2, x_3, y \in \{0, 1\}}
\end{array}
\qquad
\begin{array}{llrcl}
(P_6) & \max & 5x_1 + 8x_2 + 5x_3 & & \\
& & & & \\
& \text{s.t.} & x_1 & \leq & y \\
& & x_2 & \leq & y \\
& & x_3 & \leq & y \\
& & \multicolumn{3}{l}{x_1, x_2, x_3, y \in \{0, 1\}}
\end{array}
$$

Show that P_5 is a relaxation of P_6. Is P_6 a relaxation of P_5?

Exercise 3

For the integer programming model

$$
\begin{array}{lll}
(P_7) & \max & 2x_1 + 4x_2 + 6x_3 + 8x_4 \\
\\
& \text{s.t.} & x_1 + x_2 + 2x_3 + x_4 \quad = \quad 3 \\
& & 4x_1 + 2x_2 + x_3 \quad \leq \quad 7 \\
& & 2x_2 + 3x_3 + 3x_4 \quad \leq \quad 8 \\
& & 4x_3 + 5x_4 \quad \leq \quad 9 \\
& & x_1, x_2, x_3, x_4 \in \{0, 1\}
\end{array}
$$

Argue that P_8, as defined below, is a relaxation of P_7

$$
\begin{array}{lll}
(P_8) & \max & -2x_1 + 2x_3 + 4x_4 + 12 \\
\\
& \text{s.t.} & x_1 + x_2 + 2x_3 + x_4 \quad \geq \quad 3 \\
& & 4x_1 + 2x_2 + x_3 \quad \leq \quad 7 \\
& & 2x_2 + 3x_3 + 3x_4 \quad \leq \quad 8 \\
& & 4x_3 + 5x_4 \quad \leq \quad 9 \\
& & x_1, x_2, x_3, x_4 \in \{0, 1\}
\end{array}
$$

Wolsey Exercise 2.3

Find primal and dual bounds for the integer knapsack problem:

$$
\begin{array}{ll}
\max & 42x_1 + 26x_2 + 35x_3 + 71x_4 + 53x_5 \\
\\
\text{s.t.} & 14x_1 + 10x_2 + 12x_3 + 25x_4 + 20x_5 \quad \leq \quad 69 \\
& x \in \mathbb{Z}^5_+
\end{array}
$$

Wolsey Exercise 2.4

Consider the 0-1 integer program:

$$
(P_1) \qquad \max \left\{ cx \;:\; \sum_{j=1}^{n} a_{ij} x_j = b_i \text{ for } i = 1, \ldots, m,\; x \in \{0,1\}^n \right\}
$$

and the 0-1 equality knapsack problem:

$$
(P_2) \qquad \max \left\{ cx \;:\; \sum_{j=1}^{n} (\sum_{i=1}^{m} u_i a_{ij}) x_j = \sum_{i=1}^{m} u_i b_i,\; x \in \{0,1\}^n \right\}
$$

where $u \in \mathbb{R}^m$. Show that P_2 is a relaxation of P_1.

Wolsey Exercise 2.7

Consider the instance of the Uncapacitated Facility Location Problem with $m = 6$ clients, $n = 4$ depots and costs:

$$(c_{ij}) = \begin{pmatrix} 6 & 2 & 3 & 4 \\ 1 & 9 & 4 & 11 \\ 15 & 2 & 6 & 3 \\ 9 & 11 & 4 & 8 \\ 7 & 23 & 2 & 9 \\ 4 & 3 & 1 & 5 \end{pmatrix} \quad \text{and} \quad f_j = (21, 16, 11, 24).$$

Apply a greedy heuristic to the given instance.

Week 4: Dynamic Programming

Suggested order:

- Wolsey: Chapter 5, Exercise 1 and 3
- Wolsey: Chapter 5, Exercise 5
- Dynamic Programming Exercises

All Wolsey exercises are copied from or inspired by exercises from "Integer Programming" 1st edition by Laurence Wolsey.

Wolsey Exercise 5.1

Solve the Uncapacitated Lot-Sizing Problem with $n = 4$ periods, unit production costs $p = (1, 1, 1, 2)$, unit storage costs $h = (1, 1, 1, 1)$, set-up costs $f = (20, 10, 45, 15)$, and demands $d = (8, 5, 13, 4)$.

Wolsey Exercise 5.3

Find a maximum weight rooted subtree for the rooted tree shown in Figure 1.

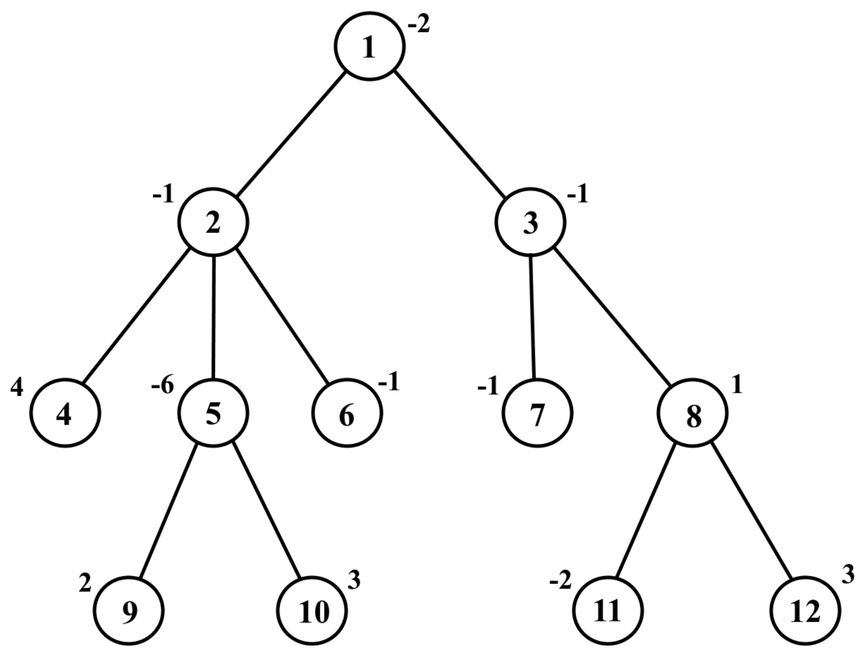


Figure 1: Rooted tree with node weights c_v.

Wolsey Exercise 5.5

Given a digraph $D = (V, A)$, travel times c_{ij} for $(i, j) \in A$ for traversing arcs, and earliest passage times r_j for $j \in V$, consider the problem of minimizing the time required to go from node 1 to node n.

(i) Describe a dynamic programming solution.

(ii) Formulate as a mixed integer program. Is this mixed integer program easy to solve?

Dynamic Programming Exercises

Exercise 1

In a city the number of crimes in each of the three police districts depends on the number of patrol cars assigned to each district. Three patrol cars are available. The table below gives the number of expected crimes for each district given the number of allocated police cars.

	No. of patrol cars			
	0	1	2	3
District 1	14	10	7	4
District 2	25	19	16	14
District 3	20	14	11	8

Use dynamic programming to determine the number of patrol cars assigned to each district so as to minimize the total number of crimes.

Hint: Let $r_i(x)$ be the number of crimes in district i given x patrol cars, that is, the number given in the table above. For a dynamic programming solution we need to come up with a recursive definition. In this case recursion can be based on the number of districts in the problem.

Exercise 2

Three tasks need to be processed in a sequence such that the total processing time is minimized, where the following table gives the processing time p_i for completing task i:

i	1	2	3
p_i	4	9	5

A setup time between consecutive tasks should also be included in the processing time. If task i is processed directly before task j, then the setup time is given by s_{ij}. All the setup costs are given in the following table:

s_{ij}	1	2	3
1		1	1
2	4		3
3	2	3	

For example, the sequence $2 \to 3 \to 1$ has total processing time

$$p_2 + s_{23} + p_3 + s_{31} + p_1 = 9 + 3 + 5 + 2 + 4 = 23$$

Suppose we would like to use dynamic programming to minimize the total processing time.

(a) Let $S \subseteq \{1, 2, 3\}$ be a subset of tasks, and consider the subproblem of finding a sequence of the tasks in S with minimum processing time. Let $c(S)$ denote the processing time for the tasks in S. Consider the recursion:

$$c(S) = \min_{\substack{j \in S \\ T = S \setminus \{j\} \\ i = f(T)}} [c(T) + p_j + s_{ij}]$$

where $S\backslash\{j\}$ means the set that results from removing j from S, and where $f(T)$ is the final task in the optimal sequence for the tasks in the subset T. This recursion does not follow the principle of optimality. Why not?

Hint: The optimal solution to the problem is the sequence $3 \rightarrow 1 \rightarrow 2$. *Note that you have the initial state values* $c(\{1\}) = 4$, $c(\{2\}) = 9$ *and* $c(\{3\}) = 5$. *Is* $c(\{1, 2, 3\})$ *the optimal solution to the problem?*

(b) Redefine the subproblem as follows: Finding a sequence of the tasks in a subset $S \subseteq \{1, 2, 3\}$ with minimum processing time such that a specific $j \in S$ is the final task to be processed. Let $c(S, j)$ be the minimum processing time for this subproblem. This recursion follows the principle of optimality:

$$c(S, j) = \min_{\substack{T = S\backslash\{j\} \\ i \in T}} [c(T, i) + p_j + s_{ij}]$$

Use it to solve the problem, and verify that it indeed gives the optimal solution.

Hint: Note that you have the initial state values $c(\{1\}, 1) = 4$, $c(\{2\}, 2) = 9$ *and* $c(\{3\}, 3) = 5$ *and the optimal solution value is one of the three state values* $c(\{1, 2, 3\}, 1)$, $c(\{1, 2, 3\}, 2)$ *or* $c(\{1, 2, 3\}, 3)$. Just follow the recursion to calculate it.

Week 5: Branch and Bound I

Suggested order:

- Wolsey: Chapter 7, Exercises 1, 2, 3 and 9
- IP Exam 2014: Question 2

All Wolsey exercises are copied from or inspired by exercises from "Integer Programming" 1st edition by Laurence Wolsey.

Wolsey Exercise 7.1

Consider the enumeration tree for a minimization problem shown in Figure 2.

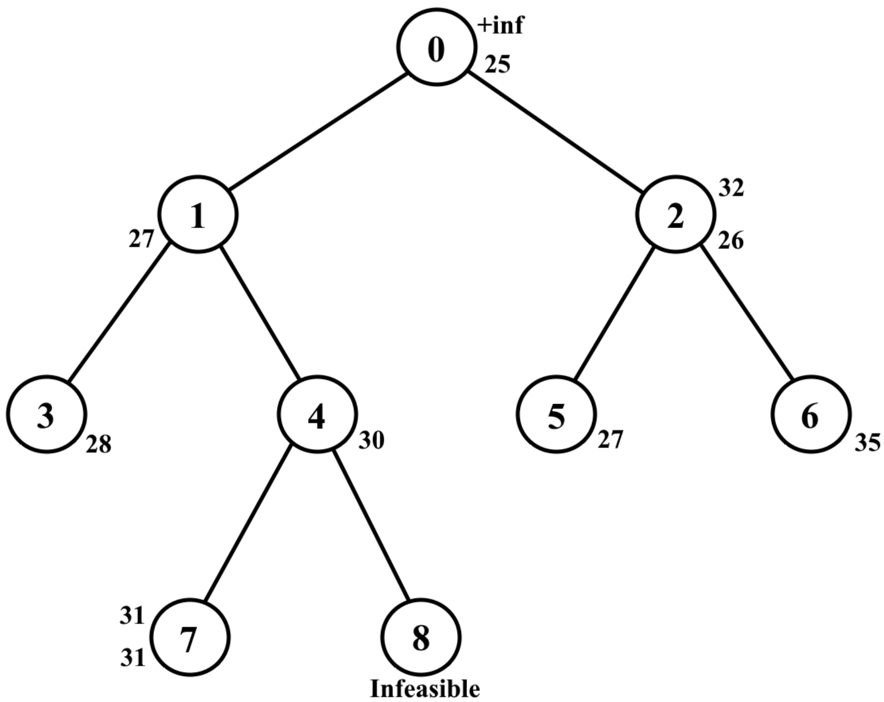


Figure 2: Enumeration tree for a minimization problem.

(i) Give the tightest possible lower and upper bounds on the optimal value z.

(ii) Which nodes can be pruned and which must be explored further?

Wolsey Exercise 7.2

Consider the two-variable integer program:

$$
\begin{array}{rrcrcl}
\max & 9x_1 & + & 5x_2 & & \\
 & & & & & \\
\text{s.t.} & 4x_1 & + & 9x_2 & \leq & 35 \\
 & x_1 & & & \leq & 6 \\
 & x_1 & - & 3x_2 & \geq & 1 \\
 & 3x_1 & + & 2x_2 & \leq & 19 \\
 & x \in \mathbb{Z}_+^2 & & & &
\end{array}
$$

Solve the problem by branch-and-bound graphically.

Wolsey Exercise 7.3

Consider the 0-1 Knapsack Problem:

$$\max\left\{\sum_{j=1}^{n} c_j x_j \;:\; \sum_{j=1}^{n} a_j x_j \le b,\, x \in \{0,1\}^n\right\}$$

with $a_j, c_j > 0$ for $j = 1, \ldots, n$.

(i) Show that if:

- $\dfrac{c_1}{a_1} \ge \cdots \ge \dfrac{c_n}{a_n} > 0$
- $\displaystyle\sum_{j=1}^{r-1} a_j \le b$
- $\displaystyle\sum_{j=1}^{r} a_j > b$

then the solution of the LP relaxation is:
$x_j = 1$ for $j = 1, \ldots, r-1$, $x_r = \dfrac{\left(b - \sum_{j=1}^{r-1} a_j\right)}{a_r}$ and $x_j = 0$ for $j > r$.

(ii) Solve the instance:

$$\begin{array}{rl} \max & 17x_1 + 10x_2 + 25x_3 + 17x_4 \\ \\ \text{s.t.} & 5x_1 + 3x_2 + 8x_3 + 7x_4 \quad \le \quad 12 \\ & x \in \{0,1\}^4 \end{array}$$

Wolsey Exercise 7.9

Consider the 0-1 problem:

$$\begin{array}{rrcrcrcrcrcr} \max & 5x_1 & - & 7x_2 & - & 10x_3 & + & 3x_4 & - & 5x_5 & & \\ \\ \text{s.t.} & x_1 & + & 3x_2 & - & 5x_3 & + & x_4 & + & 4x_5 & \le & 0 \\ & -2x_1 & - & 6x_2 & + & 3x_3 & - & 2x_4 & - & 2x_5 & \le & -4 \\ & & & 2x_2 & - & 2x_3 & - & x_4 & + & x_5 & \le & -2 \\ & x \in \{0,1\}^5 & & & & & & & & & & \end{array}$$

Simplify using logical inequalities. Can we conclude something about the problem from this?

IP Exam 2014: Question 2 (15%)

Consider the Integer Linear Programming model $(P)\quad z = \max\{cx : Ax \le b, x \text{ integer}\}$ solved by a Branch & Bound algorithm. Figure 3 represents a situation during the execution of the Branch & Bound algorithm (the symbol $\emptyset$ indicates an infeasible solution). We are about to process node P_6. The values in the nodes correspond to upper and lower bounds of the subproblem, so node P_2 has an upper bound of 70 and a lower bound of 58.

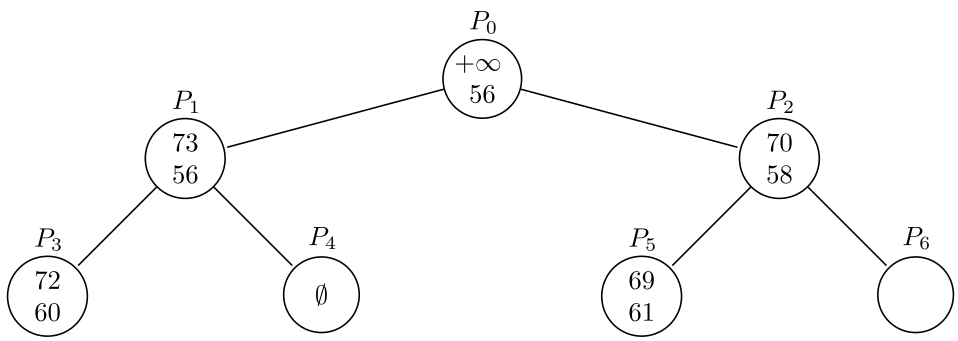


Figure 3: Branch & Bound tree for P

Subquestion 2.1

Give the tightest possible lower and upper bounds on the optimal value of z.

Subquestion 2.2

In which situations and for which values of upper and lower bound at node P_6 is it possible to prune the tree below the node P_6?

Subquestion 2.3

If any, for which values of upper bound and lower bound at node P_6 is it possible to indicate the optimal solution and close all nodes of the tree?

Week 6: Branch and Bound II

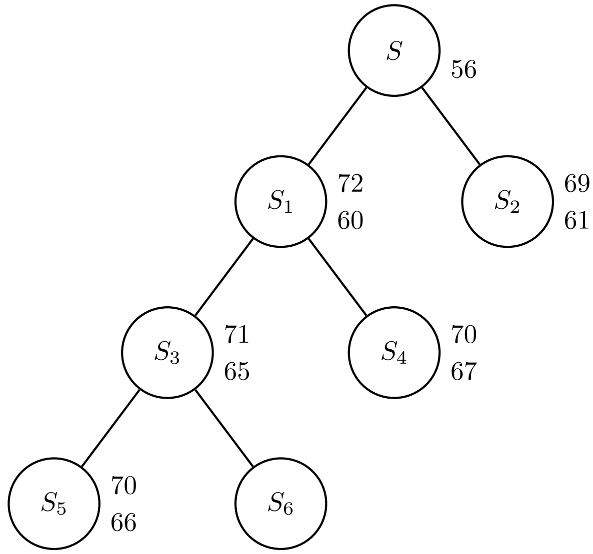
Suggested order:

- Branch and Bound Exercise
- Wolsey: Chapter 7, Exercise 5 (i) and (ii)
- IP Exam 2013: Question 4
- IP Exam 2019: Question 4.3, 4.4 and 4.5

All Wolsey exercises are copied from or inspired by exercises from "Integer Programming" 1st edition by Laurence Wolsey.

Branch and Bound Exercise

Consider the branch and bound tree below for a minimization problem.



1. Write down the tightest primal and dual bounds for this problem given the current primal and dual bounds in the tree.
2. You are about to solve a relaxation and find a heuristic feasible solution for S_6. Specify objective function values for your relaxation and heuristic that has the following effects:

   (i) Node S_2 is pruned
   (ii) Node S_4 is pruned
   (iii) Node S_5 is pruned
   (iv) Node S_6 is pruned
   (v) The problem is solved
3. What node selection strategy was used? Was it successful? Motivate your answer.

Wolsey Exercise 7.5

Definition 2.3 (Wolsey): A *1-tree* is a subgraph consisting of two edges adjacent to node 1, plus edges of a tree on nodes $\{2, \dots, n\}$.

(i) Solve the STSP instance with $n = 5$ and distance matrix:

$$(c_e) = \begin{pmatrix} - & 10 & 2 & 4 & 6 \\ - & - & 9 & 3 & 1 \\ - & - & - & 5 & 6 \\ - & - & - & - & 2 \end{pmatrix}$$

by branch-and-bound using a 1-tree relaxation (as defined above) to obtain bounds.

(ii) Solve the TSP instance with $n = 4$ and distance matrix:

$$(c_e) = \begin{pmatrix} - & 7 & 6 & 3 \\ 3 & - & 6 & 9 \\ 2 & 3 & - & 1 \\ 7 & 9 & 4 & - \end{pmatrix}$$

by branch-and-bound using an assignment relaxation to obtain bounds.

IP Exam 2013: Question 4 (20%)

Consider the IP problem P: $z = \min\{cx : Ax \leq b, x \geq 0 \text{ and integer}\}$. This can be solved using the Branch & Bound method. In Figure 4 a situation during the run of the Branch & Bound method is shown. The symbol $\emptyset$ indicates an infeasible subproblem. The values in the nodes correspond to upper and lower bounds of the subproblem, so subproblem P_4 has an upper bound of 62 and a lower bound of 57.

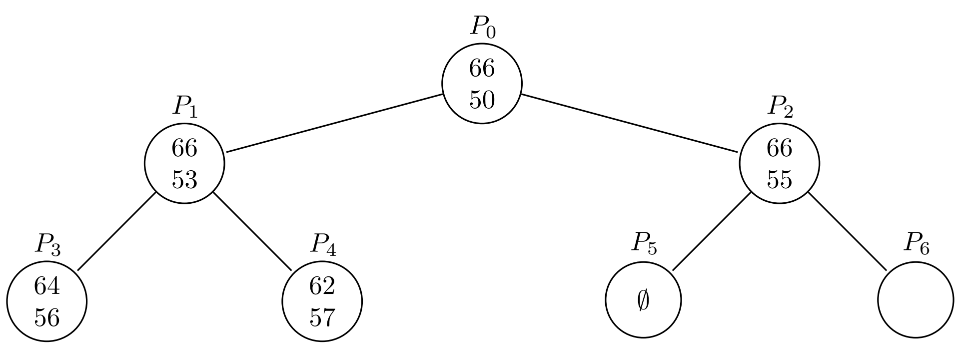


Figure 4: Branch & Bound tree for P

Subquestion 4.1

Between which values is the optimal solution contained?

Subquestion 4.2

For which values of upper and lower bound at node P_6 is it possible to prune the tree below the node?

Subquestion 4.3

For which values of upper bound and lower bound at node P_6 is it possible to indicate the optimal solution and close all nodes of the tree?

IP Exam 2019: Question 4 (25%)

An instance of the Symmetric Traveling Salesman Problem is defined by Table 1. Nodes are denoted 1 to 5. Since the problem is symmetric we have $c_{ij} = c_{ji}$ and therefore only the upper triangular part of the distance matrix is shown.

	1	2	3	4	5
1	-	6	11	9	13
2	-	-	8	15	16
3	-	-	-	7	21
4	-	-	-	-	19
5	-	-	-	-	-

Table 1: Distance matrix (with entries c_{ij}) for the Symmetric Traveling Salesman Problem.

Subquestion 4.1

Use the "nearest neighbour heuristic" starting from each of the five nodes to give the best solution and its value (if there are several solutions obtaining the same best value present them all).
The best solution is the cheapest tour. Alternatives are:

$$\begin{array}{l} (1,2,3,4,5,1) \text{ with total length } 6+8+7+19+13 = 53 \\ (2,1,4,3,5,2) \text{ with total length } 6+9+7+21+16 = 59 \\ (3,4,1,2,5,3) \text{ with total length } 7+9+6+16+21 = 59 \\ (4,3,2,1,5,4) \text{ with total length } 7+8+6+13+19 = 53 \\ (5,1,2,3,4,5) \text{ with total length } 13+6+8+7+19 = 53 \end{array}$$

So the solutions generated starting from either node 1, 4 or 5 with a total length of 53 are the best tours the nearest neighbour heuristic can generate.

Subquestion 4.2

Use the Tree Heuristic to generate a heuristic solution for the problem. When performing the walk in the Tree Heuristic, start from node 3 and if there are more than one option always select the node with the smallest index. Present the solution and its value. Based on the solution you found what is a lower bound on the optimal solution for this instance of the Symmetric TSP problem?

For the Tree Heuristic we first produce the MST with edges $(1,5),(1,2),(2,3),(3,4)$. Now we "double the edges" and start our tour at node 3. The solution for the Tree Heuristic for the STSP is $(3,2,1,5,4,3)$. Total length is $8+6+13+19+7 = 53$.

We know that the Tree Heuristic has a worst-case bound of 2, that is $\frac{z_H}{z} \leq 2$ (proposition 12.4 on page 213 in Wolsey 1st edition). That gives $\frac{53}{2} = \frac{z_h}{2} \leq z$, that is $z \geq 26.5$. This can be tightened to 27.

Subquestion 4.3

Specify the 1-tree with node 1 as the "1 node" to establish a lower bound for the problem. Given the information from subquestions 4.1, 4.2 and the 1-tree, give the best possible upper and lower bounds for the problem.

Subquestion 4.4

Use the 1-tree from subquestion 4.3 to perform one branching operation. Branching is performed on a node with a degree ≥ 3. If more than one node has degree ≥ 3 take the one with the lowest index. Do **not** use strengthening.

Present the branching tree you have now with the information from subquestion 4.3 as the "top node" in the branch and bound tree. For each branch, describe what branching has been applied and for each node, specify the upper and lower bound obtained for that particular node.

Subquestion 4.5

Given the branch and bound tree in Subquestion 4.4 can any node be pruned? Why/why not? And based on the branch and bound tree give the best upper and lower bounds to the problem?

Week 7: Lagrangian Duality

Suggested order:

- Lagrangian Relaxation Exercises: Exercise 1 and 2
- Wolsey: Chapter 10, Exercise 1
- Lagrangian Relaxation Exercises: Exercise 3 and 4

All Wolsey exercises are copied from or inspired by exercises from "Integer Programming" 1st edition by Laurence Wolsey.

Lagrangian Relaxation Exercises

Exercise 1

Consider the 0-1 knapsack problem

$$\begin{array}{lll} \max & x_1 + 3x_2 + 2x_3 + 4x_4 & \\ & & \\ \text{s.t.} & 2x_1 + 2x_2 + x_3 + 3x_4 & \leq \quad 5 \\ & x \in \{0,1\}^n & \end{array}$$

1. Write up the Lagrangian relaxation when dualizing the knapsack capacity constraint.

2. Solve the Lagrangian relaxation by inspection when the Lagrangian coefficient u is 0, $\frac{1}{2}$ and 1.

Exercise 2

Consider the assignment problem

$$\begin{array}{llll} \min & \displaystyle\sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij} & & \\ & & & \\ \text{s.t.} & \displaystyle\sum_{i=1}^{n} x_{ij} & = \quad 1 & \forall j = 1,\ldots,n \\ & \displaystyle\sum_{j=1}^{n} x_{ij} & = \quad 1 & \forall i = 1,\ldots,n \\ & x \in \{0,1\}^{n\times n} & & \end{array}$$

1. Write up the Lagrangian relaxation when dualizing the first assignment constraint.

2. For which values of the Lagrangian coefficients u_j is the Lagrangian relaxation a *relaxation*? *Hint:* Make use of Definition 2.1 (ii) in Wolsey.

Wolsey Exercise 10.1

Consider an instance of uncapacitated facility location problem with $m = 6$, $n = 5$, delivery costs

$$(c_{ij}) = \begin{pmatrix} 6 & 2 & 1 & 3 & 5 \\ 4 & 10 & 2 & 6 & 1 \\ 3 & 2 & 4 & 1 & 3 \\ 2 & 0 & 4 & 1 & 4 \\ 1 & 8 & 6 & 2 & 5 \\ 3 & 2 & 4 & 8 & 1 \end{pmatrix}$$

and fixed costs $f_j = (4, 8, 11, 7, 5)$.
Using the dual vector $u = (5, 6, 3, 2, 6, 4)$, solve the Lagrangian subproblem $IP(u)$ to get an optimal solution $(x(u), y(u))$ and a lower bound $z(u)$. You should dualize the demand constraints.
Modify the dual solution $(x(u), y(u))$ to construct a good primal feasible solution. How far is this solution from optimal?

Notice that the c_{ij}-matrix is a *cost* matrix, so this UFL problem is a minimization problem.

Hint: See Example 10.1 on pages 169-170 in Wolsey 1st edition or pages 197-198 in Wolsey 2nd edition.

Lagrangian Relaxation Exercises

Exercise 3

Consider a 0-1 knapsack problem

$$\begin{gathered} Z = \max 11x_1 + 5x_2 + 14x_3 \\ 3x_1 + 2x_2 + 4x_3 \leq 5 \\ x \in \{0,1\}^3 \end{gathered}$$

Construct a Lagrangian dual by dualizing the knapsack constraint.
What is the optimal value of the dual variable?

Exercise 4

Consider the IP:

$$\begin{array}{rlcl} \min & x_2 & & \\ & & & \\ \text{s.t.} & 18x_1 + 11x_2 & \geq & 33 \\ & 10x_1 - 18x_2 & \leq & 9 \\ & x_1, x_2 \in \{0,1,2,3\} & & \end{array}$$

Which gives you a better Lagrangian bound, if you relax the first constraint, or if you relax the second constraint? How does it compare to the LP relaxation value and the optimal integer solution value?

Hint: Use Theorem 10.3 (1st edition)/10.1 (2nd edition) and draw the constraints on a graph.

Week 10: Cutting Planes

Suggested order:

- IP Exam 2019: Question 2.1 and 2.3
- Wolsey: Chapter 8, Exercises 1, 2 and 3
- IP Exam 2020: Question 4.1
- IP Exam 2013: Question 1

All Wolsey exercises are copied from or inspired by exercises from "Integer Programming" 1st edition by Laurence Wolsey.

IP Exam 2019: Question 2 (30%)

Consider the 0-1 Knapsack Problem (P) given by:

$$\begin{aligned} z = \max \sum_{j=1}^{n} p_j x_j & \\ \text{s.t.} \sum_{j=1}^{n} w_j x_j \leq b & \\ x_j \in \{0,1\} \qquad & j = 1, \dots, n \end{aligned}$$

Table 2 describes a problem instance with $n = 10$ and $b = 68$.

Item j	1	2	3	4	5	6	7	8	9	10
Profit p_j	100	80	77	90	70	80	4	8	20	4
Weight w_j	10	10	11	15	12	20	2	8	40	16

Table 2: Data for the 0-1-Knapsack problem with 10 items.

Subquestion 2.1

Present the optimal solution and its value for the LP-relaxation of (P).

Subquestion 2.3

Determine whether each of the following inequalities is valid for (P) and if so, whether adding it would strengthen the LP relaxation.

1. $x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \leq 5$
2. $9x_1 + 9x_2 + 9x_9 + 9x_{10} \leq 68$
3. $x_1 + x_2 + x_3 + x_4 \leq 3$

Wolsey Exercise 8.1

For each of the three sets below, find a missing valid inequality and verify graphically that its addition to the formulation gives $\text{conv}(X)$.

(i) $X = \{x \in \{0,1\}^2 \,:\, 3x_1 - 4x_2 \leq 1\}$

(ii) $X = \{(x,y) \in \mathbb{R}^1_+ \times \{0,1\}^1 \,:\, x \leq 20y,\, x \leq 7\}$

(iii) $X = \{(x,y) \in \mathbb{R}^1_+ \times \mathbb{Z}^1_+ \,:\, x \leq 6y,\, x \leq 16\}$

Wolsey Exercise 8.2

In each of the examples below, a set X and a point x or (x, y) are given. Find a valid inequality for X cutting off the point.

(i) $X = \{(x, y) \in \mathbb{R}^2_+ \times \{0,1\}^1 \; : \; x_1 + x_2 \leq 2y, \; x_j \leq 1 \text{ for } j = 1, 2\}$

$$(x_1, x_2, y) = (1, 0, 0.5)$$

(ii) $X = \{(x, y) \in \mathbb{R}^1_+ \times \mathbb{Z}^1_+ \; : \; x \leq 9, \; x \leq 4y\}$

$$(x, y) = \left(9, \frac{9}{4}\right)$$

(iii) $X = \{(x, y) \in \mathbb{R}^2_+ \times \mathbb{Z}^1_+ \; : \; x_1 + x_2 \leq 25, \; x_1 + x_2 \leq 8y\}$

$$(x_1, x_2, y) = \left(20, 5, \frac{25}{8}\right)$$

(iv) $X = \{x \in \mathbb{Z}^5_+ \; : \; 9x_1 + 12x_2 + 8x_3 + 17x_4 + 13x_5 \geq 50\}$

$$(x_1, x_2, x_3, x_4, x_5) = \left(0, \frac{25}{6}, 0, 0, 0\right)$$

(v) $X = \{x \in \mathbb{Z}^4_+ \; : \; 4x_1 + 8x_2 + 7x_3 + 5x_4 \leq 33\}$

$$(x_1, x_2, x_3, x_4) = \left(0, 0, \frac{33}{7}, 0\right)$$

Wolsey Exercise 8.3

Prove that $x_2 + x_3 + 2x_4 \leq 6$ is a valid inequality for

$$X = \{x \in \mathbb{Z}^4_+ \; : \; 4x_1 + 5x_2 + 9x_3 + 12x_4 \leq 34\}.$$

IP Exam 2020: Question 4 (25%)

In a fixed-charge location problem m potential supply facilities having capacities $s_1, s_2, \ldots s_m$ are given. They can be constructed to serve n customers having corresponding demands $d_1, d_2, \ldots, d_n$. The objective function seeks to minimize the sum of the fixed costs of construction of supply facilities and the variable costs of distributing supply to meet customer demands. Let x_{ij} denote the shipping quantity from facility i to customer j, for $i = 1, \ldots, m$ and $j = 1, \ldots, n$. Let y_i be a binary variable that equals 1 if supply facility i is constructed and 0 otherwise. Then the following constraint is sufficient to ensure that whenever $y_i = 0$, we must have $x_{ij} = 0$ for all $j = 1, \ldots, n$ and that when $y_i = 1$, we have the full capacity (s_i) of supply facility i available, $i = 1, \ldots, m$:

$$\sum_{j=1}^{n} x_{ij} \leq s_i y_i \quad \text{for } i = 1, \ldots, m$$

Additionally, the following constraint ensures that all customers get its demand fullfilled.

$$\sum_{i=1}^{m} x_{ij} = d_j \ \text{ for } j = 1, \ldots, n$$

Subquestion 4.1

Argue that $x_{ij} \leq \min\{s_i, d_j\} y_i$ for $i = 1, \ldots, m$ and $j = 1, \ldots, n$ is a valid inequality for the problem described above.

IP Exam 2013: Question 1 (10%)

Given is the Integer Programming (IP) problem

$$\begin{aligned} z = \max\ & 7x_1 + 10x_2 \\ \text{s.t.}\ & -x_1 + 3x_2 + x_3 = 6 \\ & 7x_1 + x_2 + x_4 = 35 \\ & x_1, x_2, x_3, x_4 \geq 0 \text{ and integer} \end{aligned}$$

The simplex tableau for the Linear Programming (LP) relaxation is

BV	z	x_1	x_2	x_3	x_4	RHS
z	1	0	0	$\frac{63}{22}$	$\frac{31}{22}$	$66\frac{1}{2}$
x_2	0	0	1	$\frac{7}{22}$	$\frac{1}{22}$	$3\frac{1}{2}$
x_1	0	1	0	$-\frac{1}{22}$	$\frac{3}{22}$	$4\frac{1}{2}$

where BV stands for *Basic Variable* and RHS stands for *Right Hand Side*.

Subquestion 1.1

For basic variables x_1 and x_2 specify Gomory cuts both as inequalities and as equalities (with added slack variables).

Subquestion 1.2

Is it possible to use the row for the objective function to generate a Gomory cut? If no, explain why not. If yes, give a justification and specify the cut both as an inequality as well as an equality (with added slack variable).

Subquestion 1.3

Consider the inequality $2x_1 + 2x_2 \geq 3$. Given that x_1 and x_2 are non-negative integer variables the LP relaxation for this constraint will be feasible for $x_1 = 1.5$ and $x_2 = 0$. Use integer rounding to strengthen the inequality to produce a constraint (a cut) that is not feasible for $x_1 = 1.5$ and $x_2 = 0$.

Week 11: Strong Valid Inequalities

Suggested order:

- Wolsey: Chapter 9, Exercise 3 (i)-(iii)
- Wolsey: Chapter 9, Exercise 5
- Wolsey: Chapter 9, Exercise 4 (i)
- IP Exam 2017: Question 4
- IP Exam 2014: Question 5

All Wolsey exercises are copied from or inspired by exercises from "Integer Programming" 1st edition by Laurence Wolsey.

Wolsey Exercise 9.3

In each of the examples below, a set X and a point x are given. Find a valid inequality for cutting off the point.

(i) $X = \{x \in \{0,1\}^5 \,:\, 9x_1 + 8x_2 + 6x_3 + 6x_4 + 5x_5 \leq 14\}$

$$(x_1, x_2, x_3, x_4, x_5) = \left(0, \frac{5}{8}, \frac{3}{4}, \frac{3}{4}, 0\right)$$

(ii) $X = \{x \in \{0,1\}^5 \,:\, 9x_1 + 8x_2 + 6x_3 + 6x_4 + 5x_5 \leq 14\}$

$$(x_1, x_2, x_3, x_4, x_5) = \left(\frac{1}{4}, \frac{1}{8}, \frac{3}{4}, \frac{3}{4}, 0\right)$$

(iii) $X = \{x \in \{0,1\}^5 \,:\, 7x_1 + 6x_2 + 6x_3 + 4x_4 + 3x_5 \leq 14\}$

$$(x_1, x_2, x_3, x_4, x_5) = \left(\frac{1}{7}, 1, \frac{1}{2}, \frac{1}{4}, 1\right)$$

Wolsey Exercise 9.5

Consider the set

$$X = \{(x, y) \in \mathbb{R}^4_+ \times \{0,1\}^4 \,:\, x_1 + x_2 + x_3 + x_4 \geq 36,\ x_1 \leq 20y_1,\ x_2 \leq 10y_2,\ x_3 \leq 10y_3,\ x_4 \leq 8y_4\}$$

(i) Derive a valid inequality that is a 0-1 knapsack constraint.

(ii) Use this to cut off the fractional point $x = (20, 10, 0, 6)$, $y = (1, 1, 0, \frac{3}{4})$ with an inequality involving only y variables.

Wolsey Exercise 9.4

Consider the knapsack set

$$X = \{x \in \{0,1\}^6 \,:\, 12x_1 + 9x_2 + 7x_3 + 5x_4 + 5x_5 + 3x_6 \leq 14\}$$

and the cover inequality $x_3 + x_5 + x_6 \leq 2$ that is valid for X.

(i) Lift the inequality to obtain a strong valid inequality $\alpha_1 x_1 + \alpha_2 x_2 + \alpha_4 x_4 + x_3 + x_5 + x_6 \leq 2$ for X.

IP Exam 2017: Question 4 (25%)

Consider the Integer Linear Programming model (P) given by:

$$\begin{array}{rrll} z & = & \max & 6x_1 + 28x_2 + 11x_3 + 36x_4 + 16x_5 + 20x_6 \\ & & \text{s.t.} & 3x_1 + 5x_2 + 4x_3 + 6x_4 + 8x_5 + 9x_6 \leq 10 \\ & & & x_1, \ldots, x_6 \in \{0, 1\} \end{array}$$

with the LP relaxation optimum $x = (0, \frac{4}{5}, 0, 1, 0, 0)$.

Subquestion 4.1

We have during the course looked at different "standard problems". What is the name of this standard problem?

Subquestion 4.2

Write up the LP relaxation of (P).

Subquestion 4.3

Is the model below a relaxation of (P)?

$$\begin{array}{rrll} z & = & \max & 6x_1 + 28x_2 + 11x_3 + 36x_4 + 16x_5 + 20x_6 \\ & & \text{s.t.} & 3x_1 + 5x_2 + 4x_3 + 6x_4 + 8x_5 + 9x_6 \leq 10 \\ & & & 0 \leq x_i \text{ for } i = 1, \ldots, 6 \end{array}$$

If it is a relaxation is it tighter than the LP relaxation of subquestion 4.2 or vice versa?

Subquestion 4.4

Determine whether each of the following inequalities are valid for (P) and if so, whether they would strengthen the LP relaxation.

1. $6x_1 + 20x_6 \leq 10$
2. $x_1 + x_3 + x_4 \leq 1$
3. $x_2 + x_4 \leq 1$

Subquestion 4.5

What cover inequality for (P) will the separation problem detect? Present the separation problem for the given instance and the inequality derived.

Subquestion 4.6

Consider the cover inequality $x_1 + x_2 + x_3 \leq 2$. Is it a minimal cover inequality? Can you strengthen the cover inequality by x_4? For which values of β can you strengthen the cover inequality by βx_4?

IP Exam 2014: Question 5 (20%)

The Minimum Spanning Tree (MST) problem is a classic optimization problem. There exist several tailor-made algorithms to solve the problem to optimality but in this question we will look at the problem based on it's formulation as an Integer Linear Programming problem.

In the MST problem an undirected graph $G = (V, E)$ is given. For each edge $e \in E$ there is associated a cost c_e. The solution T to an MST problem is a subset of edges in G that forms a tree (so no cycles). In addition, "spanning" means that the tree needs to "visit" all nodes. The cost of a spanning tree is the sum of the cost of the edges in the tree. The MST calls for finding a minimum cost spanning tree. It can be formulated as an Integer Linear Programming problem:

$$
\begin{aligned}
z = \min \sum_{e \in E} c_e x_e & \\
\text{s.t.} \sum_{e \in E} x_e = n - 1 & \\
\sum_{e \in E(S)} x_e \leq |S| - 1 & \quad \text{for } S \subset V, |S| \geq 2 \\
x_e \in \{0, 1\} & \quad \text{for } e \in E
\end{aligned}
$$

where n is the number of nodes in the problem. We will denote the model (P). Remember that $E(S)$ denotes the set of edges with both endpoints in S.

Subquestion 5.1

Since there are exponentially many of the constraints on the form $\sum_{e \in E(S)} x_e \leq |S| - 1$ (subtour elimination constraints) even for small problem sizes solving the model directly in an IP solver might not be feasible.

Instead, we might use a solution approach based on the cutting plane algorithm and LP-based Branch & Bound. Here we will only focus on solving the root node of the Branch & Bound tree.

We start out with the LP relaxation of (P). In addition, we remove all subtour elimination constraints. After solving the LP relaxation we use the cutting plane algorithm to search for a violated inequality among the $\sum_{e \in E(S)} x_e \leq |S| - 1$ constraints. If we find a violated inequality it is added to the LP relaxation which is then resolved. In this way we continue until we cannot identify any more violated inequalities. Then the LP will return the optimum of the LP relaxation of (P).

In order to use the cutting plane algorithm we need a separation algorithm for the constraints $\sum_{e \in E(S)} x_e \leq |S| - 1$. Devise a separation algorithm for the problem. Present it as an Integer Linear Programming model.

"Post-exam hint": The separation algorithm can be constructed in the same way as in Wolsey for the Generalized subtour eliminiation constraints from the prize-collecting TSP (pages 154-155 1st edition/183-184 2nd edition). Assume the optimal LP solution is x^*. If it is violated we will have $\sum x^*_{e \in E(S)} > |S| - 1$.

Subquestion 5.2

Consider the following problem: $G = (V, E)$ where $V = \{1, 2, 3, 4, 5\}$ and where $E = \{(1, 2), (1, 3), (2, 4), (2, 5), (3, 4), (4, 5)\}$. The costs are given as $c_{12} = 1, c_{13} = 2, c_{24} = 4, c_{25} = 5, c_{34} = 3, c_{45} = 6$.

What is the solution to the LP relaxation (P) without the subtour elimination constraints (so the initial LP problem in our solution approach)?

Subquestion 5.3

Use your separation algorithm to construct a subtour elimination constraint that cuts away the LP optimum you found in subquestion 5.2. Writing up the separation problem and solving it by inspection is allowed.

Week 12: Heuristics

Suggested order:

- IP Re Exam 2020: Question 3
- IP Exam 2016: Question 4
- IP Exam 2020: Question 4.2
- Wolsey: Chapter 12, Exercises 1
- Wolsey: Chapter 12, Exercises 5

All Wolsey exercises are copied from or inspired by exercises from ”Integer Programming” 1st edition by Laurence Wolsey.

IP Re Exam 2020: Question 3 (25%)

An instance of the Symmetric Traveling Salesman Problem (STSP) is defined by Table 3. Nodes are denoted 1 to 5. Since the problem is symmetric we have $c_{ij} = c_{ji}$ and therefore only the upper triangular part of the distance matrix is shown.

	1	2	3	4	5
1	-	6	11	19	17
2	-	-	8	15	14
3	-	-	-	7	21
4	-	-	-	-	9
5	-	-	-	-	-

Table 3: Distance matrix (with entries c_{ij}) for the STSP.

Subquestion 3.1

The "nearest neighbour heuristic" for the STSP needs a node to start from. The length of the tour can be different based on where you start from. Use each of the different nodes as starting node and give the best solution and its value (if there are several solutions obtaining the same best value present them all).

Subquestion 3.2

Specify the 1-tree with node 1 as the "1 node" to establish a lower bound for the problem. Given the information from subquestions 3.1 and the 1-tree, give the most narrow interval for the optimal solution. If we wanted to use the 1-tree in a branch-and-bound how would we branch and how many branch-and-bound nodes would we create based on this 1-tree?

Subquestion 3.3

The Dive-and-Fix heuristic is described in Wolsey on page 214-215 (1st edition)/260 (2nd edition). It is described as a method to be used on a mixed binary integer programming problems, but we can also use it for the STSP with 1-tree for bounding. Let F be the set of "fixed" edges, that is, the edges that cannot be part of the 1-tree as in the branching rule for Brand-and-Bound using 1-trees. This corresponds to fixing a binary variable to 0 or 1 in the original Dive-and-Fix algorithm. The algorithm will be as follows:

Algorithm 1: Dive-and-Fix for STSP using 1-trees

Data: The initial 1-tree T
Result: Termination when T is a tour

```
1 F = ∅;
2 Let i be the node with the highest degree in the T;
3 Exclude the edge (i, j) with the highest cost from T. Add (i, j) to F;
4 Find the 1-tree in the resulting graph, where edges in F are removed from the original graph –
    denote the solution T;
5 if T infeasible then
6 |   Stop (the heuristic has failed)
7 else
8 |   Go back to line 2
9 end
```

Use the Dive-and-Fix on the data from Table 3. Present the solution and its value. Also present, which edges are excluded and in which order. In case that the Dive-and-Fix heuristic does not find a feasible solution, report the decisions taken by the Dive-and-Fix heuristic until it fails.

Subquestion 3.4

In general, a Dive-and-Fix may fail to deliver a feasible solution. What about the above Dive-and-Fix for STSP with 1-trees, can it fail? Why/why not?

IP Exam 2016: Question 4 (30%)

A watchman has to inspect several buildings. In each of the buildings there are a series of check points he needs to visit. As soon as he enters a building he has to visit all the check pointsin that building before he can continue to another building. The aim in the *the Watchmans tours problem* is to construct a tour through the check points in the buildings that minimize the overall distance of the tour.

Let us define the problem in a complete graph $G = (V, E)$ where V is the vertex set, each vertex representing a check point, and E is the set of all edges between any two vertices. A non-negative cost c_{ij} is associated with each edge (i, j). The vertex set V is partitioned into m mutual exclusive and exhaustive clusters (one for each building) $V_1, V_2, \ldots, V_m$.

An example with $m = 4$ can be seen below.

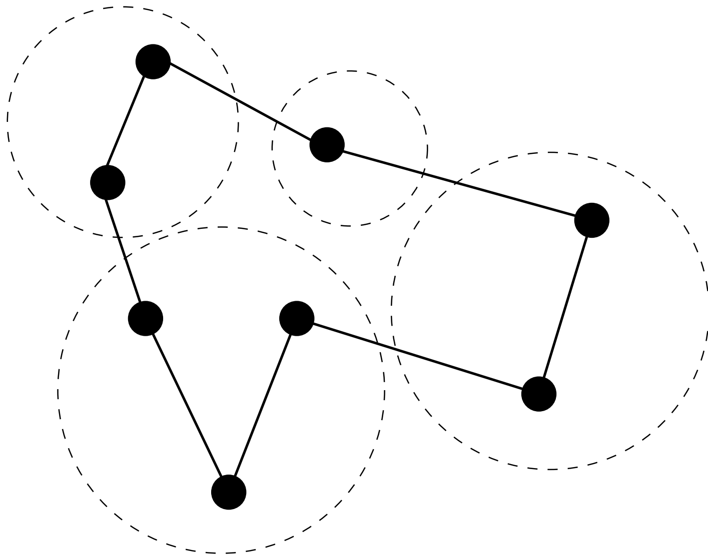


Figure 5: The Wathmans tour. Only edges in the solution are shown.

Subquestion 4.1

Let x_{ij} be a binary variable indicating whether (i,j) is used in the optimal solution $(=1)$ or not $(=0)$. The integer programming model for the problem could be formulated like the following:

$$
\begin{aligned}
\min \; & \sum_{(i,j)\in E} c_{ij}x_{ij} \\
\text{s.t.} \; & \sum_{j\in V, i\neq j} x_{ij} = 1 \quad \text{for all } i \in V \\
& \sum_{i\in V, i\neq j} x_{ij} = 1 \quad \text{for all } j \in V \\
& \sum_{i\in S}\sum_{j\neq S} x_{ij} \geq 1 \quad \text{for all } S \subset V, S \neq \emptyset \\
& \sum_{i,j\in V_k} x_{ij} = R_k \quad \text{for all } k = 1, 2, \ldots, m \\
& x_{ij} \in \{i,j\} \text{ for all } (i,j) \in E
\end{aligned}
$$

We just need to assign a value to the parameter R_k on the right hand side of the last constraint for each cluster (building) k. State and explain this value.

Subquestion 4.2

The problem can also be solved by transforming it into the classic TSP problem. Let the set of vertices V and the set of edges E remain unchanged. We only have to change the cost c_{ij}. Denote the changed cost $\hat{c}_{ij}$. Describe how to determine $\hat{c}_{ij}$ based on c_{ij}, so that by solving the classic TSP problem with $\hat{c}_{ij}$ as cost you get an optimal solution to the watchmans problem.

Subquestion 4.3

How does the cost of the TSP tour relate to the cost of the clustered TSP tour?

Subquestion 4.4

Below is the values for a small problem with $m = 3$ given. Use your transformation to a TSP problem and then find a heuristic solution using the **The Tree Heuristic for STSP**. Distance matrix is given as:

	1	**2**	**3**	**4**	**5**
1	–	11	9	15	16
2	–	–	14	10	10
3	–	–	–	13	11
4	–	–	–	–	8
5	–	–	–	–	–

$V_1 = \{1\}$, $V_2 = \{2, 3\}$ and finally $V_3 = \{4, 5\}$.

IP Exam 2020: Question 4 (25%)

In a fixed-charge location problem m potential supply facilities having capacities $s_1, s_2, \ldots s_m$ are given. They can be constructed to serve n customers having corresponding demands $d_1, d_2, \ldots, d_n$. The objective function seeks to minimize the sum of the fixed costs of construction of supply facilities and the variable costs of distributing supply to meet customer demands. Let x_{ij} denote the shipping quantity from facility i to customer j, for $i = 1, \ldots, m$ and $j = 1, \ldots, n$. Let y_i be a binary variable that equals 1 if supply facility i is constructed and 0 otherwise. Then the following constraint is sufficient to ensure that whenever $y_i = 0$, we must have $x_{ij} = 0$ for all $j = 1, \ldots, n$ and that when $y_i = 1$, we have the full capacity (s_i) of supply facility i available, $i = 1, \ldots, m$:

$$\sum_{j=1}^{n} x_{ij} \leq s_i y_i \quad \text{for } i = 1, \ldots, m$$

Additionally, the following constraint ensures that all customers get its demand fulfilled.

$$\sum_{i=1}^{m} x_{ij} = d_j \;\; \text{for } j = 1, \ldots, n$$

Subquestion 4.2

Assume that all s_i's and d_j's are integer. Consider the dive-and-fix heuristic described in Wolsey in section 12.5 (1st edition, section 13.4 in 2nd edition) to produce a feasible solution to the fixed-charge location problem.

Data for an instance with three supply facilities and three customers are given. $s_1 = 120, s_2 = 25$ and $s_3 = 25$ and three customers with demands $d_1 = 40$ and $d_2 = d_3 = 10$.

The initial LP solution used by the dive-and-fix is: $y_1 = \frac{1}{3}, y_2 = \frac{2}{5}$ and $y_3 = \frac{10}{25}$ and $x_{11} = 40, x_{22} = x_{23} = x_{32} = x_{33} = 5$. All other x_{ij} are equal to 0.

Which variable will be fixed and to what? What happens with the LP when solved as part of "Solve the resulting LP" (referring to Wolsey's description of dive-and-fix in section 12.5/13.4)?

Wolsey Exercise 12.1

You are given an instance of STSP with the following distance matrix:

$$\begin{pmatrix} - & & & & & & \\ 28 & - & & & & & \\ 57 & 28 & - & & & & \\ 72 & 45 & 20 & - & & & \\ 81 & 54 & 3 & 10 & - & & \\ 85 & 57 & 28 & 20 & 22 & - & \\ 80 & 63 & 57 & 72 & 81 & 63 & - \end{pmatrix}$$

Apply the following heuristics to the problem:

(i) Nearest Neighbour Heuristic

(ii) Cheapest Insertion Heuristic

(iii) Farthest Insertion Heuristic

(iv) Savings Heuristic (”Pure greedy” in Wolsey)

(v) Tree Heuristic

(vi) Christofides’ Heuristic (”Tree/matching” in Wolsey 1st edition)

Wolsey Exercise 12.5

Consider the 0-1 knapsack problem:

$$z = \max\{\sum_{j\in N} c_j x_j \;:\; \sum_{j\in N} a_j x_j \le b,\, x \in \{0,1\}^n\}$$

with $a_j > 0$ for $j \in N$.
Then consider a greedy heuristic that chooses the better of two solutions:

- The linear programming solution with variables rounded down to integer values.
- The best solution in which just one variable is set to one.

The solution of the greedy heuristic is called z^G.
Show that $z^G \ge \frac{1}{2} z$.

Contents