

## Question 1

(a)

To formulate an integer program, we need to formulate a set of decision variables, constraints, and the objective function.

### Decision Variables

Let  $x_1, x_5, x_{10}, x_{20}, x_{50}, x_{100}$  each denote the number of coins of denomination 1 cent, 5 cents, 10 cents, 20 cents, 50 cents, and \$1 respectively.

### Constraints

(i) Integrality & Non-Negativity Constraints

Since we cannot take fractional number of coins,  $x_1, x_5, x_{10}, x_{20}, x_{50}, x_{100}$  are all limited to non-negative integers. Furthermore, we know that there are only 30 copies of each denomination available. Thus:

$$x_i \in \{0, 1, \dots, 30\}, \quad \forall i = 1, 5, 10, 20, 50, 100$$

(ii) Maximum Weight of Returned Coins

The total maximum weight of the returned coins has to be less than 95 g. As such:

$$1.24x_1 + 1.7x_5 + 2.36x_{10} + 3.85x_{20} + 6.56x_{50} + 7.62x_{100} < 95$$

(iii) Total Sum of Money

Finally, we must break down precisely SGD10.74 using the coins. Thus:

$$0.01x_1 + 0.05x_5 + 0.1x_{10} + 0.2x_{20} + 0.5x_{50} + x_{100} = 10.74$$

### Objective Function

We want to minimize the number of coins to use. Therefore, our objective function must output the total number of coins to use. This can simply be expressed as:

$$Z = x_1 + x_5 + x_{10} + x_{20} + x_{50} + x_{100}$$

### Model

Finally, we have our model formulation:

$$\begin{aligned} \min \quad & x_1 + x_5 + x_{10} + x_{20} + x_{50} + x_{100} \\ \text{s.t.} \quad & 1.24x_1 + 1.7x_5 + 2.36x_{10} + 3.85x_{20} + 6.56x_{50} + 7.62x_{100} < 95 \\ & 0.01x_1 + 0.05x_5 + 0.1x_{10} + 0.2x_{20} + 0.5x_{50} + x_{100} = 10.74 \\ & x_i \in \{0, 1, \dots, 30\}, \quad \forall i = 1, 5, 10, 20, 50, 100 \end{aligned}$$

**(b)**

Neither of the solutions are feasible.

$$x_1 = 4, \quad x_5 = 0, \quad x_{10} = 0, \quad x_{20} = 1, \quad x_{50} = 2, \quad x_{100} = 9.5$$

is not a feasible solution because  $x_{100}$  takes a fractional value, violating one of the constraints. Meanwhile,

$$x_1 = 4, \quad x_5 = 0, \quad x_{10} = 0, \quad x_{20} = 1, \quad x_{50} = 3, \quad x_{100} = 9$$

is not a feasible solution because the total weight of all the coins are 97.07g, which exceed the maximum total weight of 95g.

**(c)**

Introduce indicator variable  $\delta \in \{0, 1\}$  where:

$$\begin{aligned} x_{100} \geq 8 &\iff \delta = 1 \\ \delta = 1 &\implies x_{50} \leq 5 \\ \therefore x_{100} \geq 8 &\implies x_{50} \leq 5 \end{aligned}$$

By the contrapose of ( $x_{100} \geq 8 \implies \delta = 1$ ), we get:

$$\begin{aligned} \delta = 0 &\implies x_{100} < 8 \\ \delta = 0 &\implies x_{100} \leq 7 \quad \text{since } x_{100} \text{ integer} \end{aligned}$$

To summarize this implication in an inequality:

$$x_{100} \leq 7 + M_1\delta$$

When  $\delta = 0$ , there is no extra constraint on  $x_{100}$  and thus  $x_{100} \leq 30$ . Therefore,  $30 = 7 + M_1$ , giving:

$$\boxed{x_{100} \leq 7 + 23\delta}$$

Additionally, we also know that:

$$\delta = 0 \implies x_{50} \leq 5 + M_2$$

for some  $M_2$  which generalizes the constraint. To summarize this implication in an inequality:

$$x_{50} \leq 5 + M_2(1 - \delta)$$

In general, when  $\delta = 0$ , we have  $x_{50} \leq 30$ . Thus  $30 = 5 + M_2$ , giving:

$$\boxed{x_{50} \leq 30 - 25\delta}$$

Therefore, the condition “if we use at least eight dollar coins we can use at most five 50 cent coins” can be described as:

$$\begin{aligned} x_{100} &\leq 7 + 23\delta \\ x_{50} &\leq 30 - 25\delta \\ \delta &= \begin{cases} 1, & \text{if } x_{100} \geq 8 \\ 0, & \text{otherwise} \end{cases} \end{aligned}$$

(d)

When  $\delta = 0$ , the constraints become:

$$x_{100} \geq 8 - 8\delta$$

$$x_{100} \geq 8$$

$$x_{50} \geq 6 - 6(1 - \delta)$$

$$x_{50} \geq 0$$

In other words, when  $\delta = 0$ , we use at least eight dollar coins (and any amount of 50 cent coins not more than 30).

Meanwhile, when  $\delta = 1$ , the constraints become:

$$x_{100} \geq 8 - 8\delta$$

$$x_{100} \geq 0$$

$$x_{50} \geq 6 - 6(1 - \delta)$$

$$x_{50} \geq 6$$

In other words, when  $\delta = 1$ , we use at least six 50 cent coins (and any amount of dollar coins not more than 30).

Since either  $\delta = 1$  or  $\delta = 0$ , logically speaking, we either use at least eight dollar coins or we use at least six 50 cent coins.

(e)

Although both constraints are equivalent, the first constraint is better as it restricts the feasible region into a tighter, smaller confined space which makes it easier for any solver algorithm to terminate. Thus, the first constraint is a stronger formulation and is therefore the preferred constraint.

## Question 2

The following are the feasible regions of  $P_1$ ,  $P_2$ , and  $P_3$ .

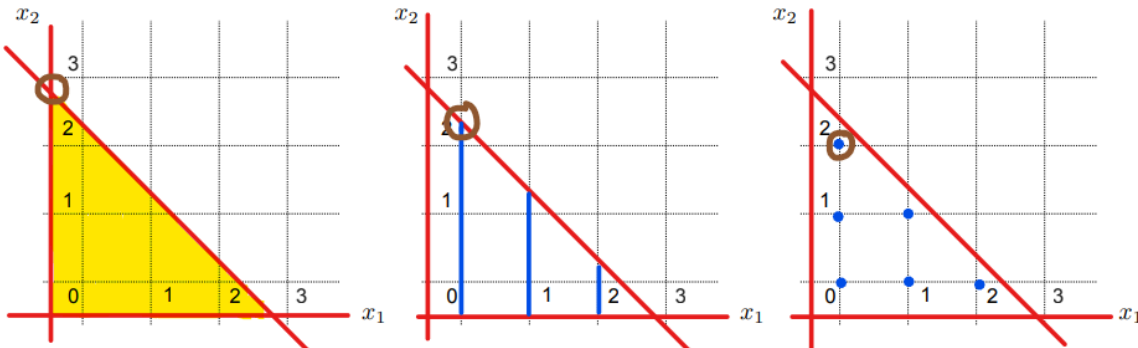


Figure 1: Feasible Region for  $P_1$ ,  $P_2$ , and  $P_3$

By analyzing the direction of the contour of the objective function,  $\nabla z = \nabla(x_1 + 9x_2)$ , we

understand that the objective function increases in the direction of  $(1, 9)$ . With this in mind, the optimal solution for  $P_1$ ,  $P_2$ , and  $P_3$  are the top left cornerpoint of each problem's feasible region.

Thus the optimal solutions for  $P_1$ ,  $P_2$ , and  $P_3$ :

$$\begin{array}{lll} x_{P_1}^* = (-0.5, 2.7) & x_{P_2}^* = (0, 2.2) & x_{P_3}^* = (0, 2) \\ z_{P_1}^* = 23.8 & z_{P_2}^* = 19.8 & z_{P_3}^* = 18 \end{array}$$

The coordinates for  $x_{P_1}^*$  were found by finding the solution for  $x_1 = -0.5$  and  $x_1 + x_2 = 2.2$ , which represent the top left cornerpoint of  $P_1$ 's feasible region.

The coordinates for  $x_{P_2}^*$  were found by finding the solution for  $x_1 = 0$  and  $x_1 + x_2 = 2.2$ , which represent the top left cornerpoint of  $P_2$ 's feasible region.

Finally, the coordinates for  $x_{P_3}^*$  were found by selecting the top left cornerpoint among all available points (which in this case is obviously  $(0, 2)$ .)

## Question 3

(1)

Another formulation of this problem would occur when  $1 \leq t < 2$ .

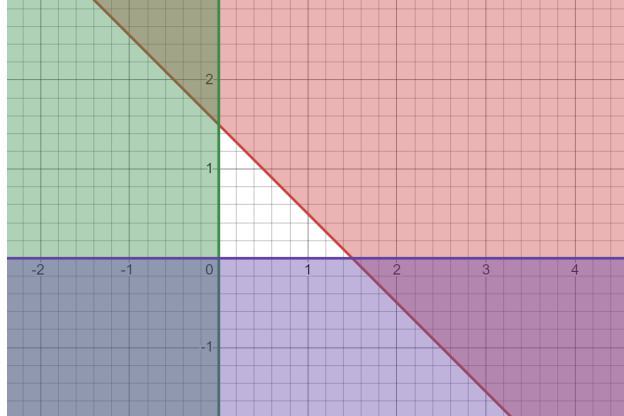
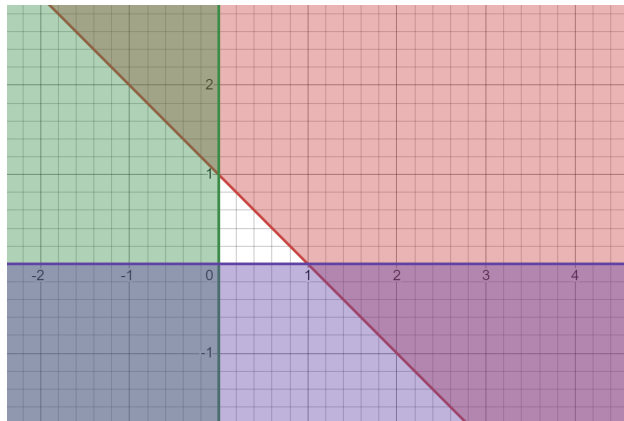
If  $t \geq 2$ , then this formulation would include integer solutions  $(0,2)$ ,  $(2,0)$ ,  $(1,1)$  and potentially other integer solutions in addition to  $(0,0)$ ,  $(0,1)$ , and  $(1,0)$ . Our model should only cover  $(0,0)$ ,  $(0,1)$ , and  $(1,0)$ . Therefore  $t \not\geq 2$ , or in other words  $t < 2$ .

Meanwhile, if  $t < 1$ , then this formulation would exclude the integer solutions  $(0,0)$ ,  $(0,1)$ , and  $(1,0)$ , which are crucial for the formulation to include. As such  $t \not< 1$  or in other words  $t \geq 1$ .

Only when  $1 \leq t < 2$  does a new formulation of the problem occur while keeping exactly  $(0,0)$ ,  $(0,1)$ , and  $(1,0)$  as the only integer solutions, no more no less.

(2)

$t = 1$  creates a stronger formulation than  $t = 1.5$ , since it is an equivalent formulation with the smaller feasible region as shown below:

Figure 2: Feasible region (unshaded) when  $t = 1.5$ Figure 3: Feasible region (unshaded) when  $t = 1$ 

## Question 4

(a)

$P_2$  and  $P_3$  were branched based on the fractional variable  $x_2 = 2.2$ . Since,  $x_2 = 2.2$  is a non-integer, we want to exclude it by creating two new subproblems  $P_2$  and  $P_3$ . We do this by splitting the feasible region of  $P_1$  into two halfspaces  $x_2 \geq 2.2$  ( $P_3$ ) and  $x_2 < 2.2$  ( $P_2$ ). Since  $x_2$  integer, these constraints can be reformulated into  $x_2 \geq 3$  ( $P_3$ ) and  $x_2 \leq 2$  ( $P_2$ ).

(b)

$P_5$  can be derived as follows:

$$\begin{aligned} \max \quad & x_1 + 9x_2 \\ \text{s.t.} \quad & x_1 + x_2 \leq 2.2 \\ & x_1, x_2 \geq 0 \\ & x_2 \leq 2 \\ & x_1 \geq 1 \\ & x_1, x_2 \text{ integers} \end{aligned}$$

$P_4$  and  $P_5$  are the subproblems of  $P_2$ . Evidently,  $P_4$  was formulated by branching  $P_2$  on  $x_1 = 0.2$ . This was done by dividing the feasible region of  $P_2$  using two halfspaces:  $x_1 \leq 0.2$  ( $P_4$ ) and  $x_1 > 0.2$  ( $P_5$ ). Since  $x_1$  integer, these halfspaces can be reformulated into:  $x_1 \leq 0$  ( $P_4$ ) and  $x_1 \geq 1$  ( $P_5$ ).

Therefore, by simply adding the constraint  $x_1 \geq 1$  on the parent subproblem  $P_2$ , we derive the formulation for  $P_5$ .

(c)

To begin, select  $P_4$  as the active subproblem.  $P_4$  has the following feasible region:

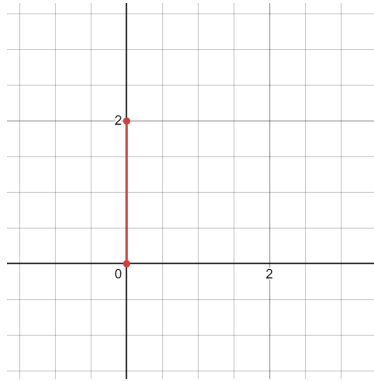
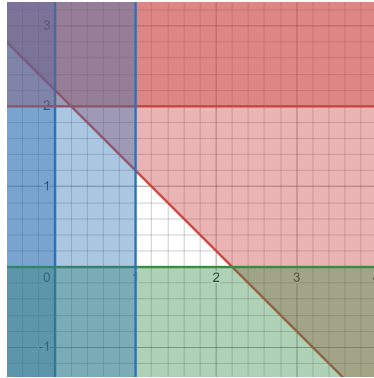


Figure 4: Feasible Region (shaded line) of  $P_4$

By the graphical approach, we see that the optimal solution for  $P_4$  is  $x_{P_4}^* = (0, 2)$  with value  $z_{P_4}^* = 18$ . We update our best guess as follows:

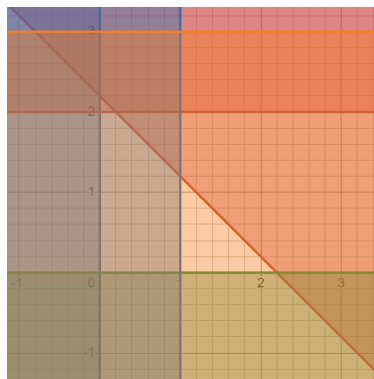
$$\begin{aligned} x_{\text{bg}} &\leftarrow x_{P_4}^* = (0, 2) \\ L &\leftarrow z_{P_4}^* = 18 \end{aligned}$$

We prune  $P_4$  by optimality and select  $P_5$  as the next active subproblem. By the graphical approach, it is clear that the optimal solution for  $P_5$  is  $x_{P_5}^* = (1, 1.2)$  with value  $z_{P_5}^* = 11.8$ . However, since  $z_{P_5} < L$ , we do not update  $x_{\text{bg}}$  and  $L$ . In other words,  $P_5$  cannot produce any more optimal solution than  $P_4$ .

Figure 5: Feasible Region (unshaded area) of  $P_5$ 

We prune  $P_5$  by bound and select  $P_2$  as the next active subproblem. As  $P_2$  was split into two subproblems  $P_4$  and  $P_5$ , and both have been solved,  $P_2$  have been solved too. No need to update our best guess.

We prune  $P_2$  and select  $P_3$  as the next active subproblem. The feasible region of  $P_3$  is as follows:

Figure 6: Feasible Region (unshaded area) of  $P_3$ 

Evidently, the feasible region of  $P_3$  is an empty set. In other words, there are no solutions that satisfy all the constraints in  $P_3$ .

We prune  $P_3$  by infeasibility and select  $P_1$  as the next active subproblem. As  $P_1$  was split into two subproblems  $P_4$  and  $P_5$ , and both have been solved,  $P_1$  have been solved too. We prune  $P_1$  and since there are no more remaining subproblems, the Branch and Bound algorithm terminates here. With that, our current best guess becomes our optimal solution:

$$x^* = (0, 2)$$

$$z^* = 18$$

## Question 5

The following prompt was given to ChatGPT.



**You**

Question 5 (20 points) The board of directors of General Wheels Co. is considering six large capital investments. Each investment can be made only once. These investments differ in the estimated long-run profit (net present value) that they will generate as well as in the amount of capital required, as shown by the following table (in units of millions of dollars):

Investment Opportunity	1	2	3	4	5	6
Estimated profit	15	12	16	18	9	11
Capital required	38	33	39	45	23	27

The total amount of capital available for these investments is 100 million. Investment opportunities 1 and 2 are mutually exclusive, and so are 3 and 4. Furthermore, neither 3 nor 4 can be undertaken unless one of the first two opportunities is undertaken. There are no such restrictions on investment opportunities 5 and 6. The objective is to select the combination of capital investments that will maximize the total estimated long-run profit (net present value).

Express the problem as an integer program.

Figure 7: Prompt for Question 5

The following model was ChatGPT's response:

$$\begin{aligned}
 &\max 15x_1 + 12x_2 + 16x_3 + 18x_4 + 9x_5 + 11x_6 \\
 &\text{s.t. } 38x_1 + 33x_2 + 39x_3 + 45x_4 + 23x_5 + 27x_6 \leq 100 \\
 &\quad x_1 + x_2 \leq 1 \\
 &\quad x_3 + x_4 \leq 1 \\
 &\quad x_1 + x_2 \geq x_3 + x_4 \\
 &\quad x_i \in \{0, 1\}, \quad \forall i = 1, \dots, 6
 \end{aligned}$$

## Decision Variables

The model defines the decision variables  $x_1, x_2, x_3, x_4, x_5, x_6$ , where:

$$x_i = \begin{cases} 1, & \text{if the } i\text{-th investment opportunity is taken} \\ 0, & \text{otherwise} \end{cases}, \quad \forall i = 1, \dots, 6$$

## Constraints

The model uses the following constraints:

### (i) Total Investment Budget

The total amount of capital available is 100 million. As such, the price of all investments taken must not exceed a value of 100 million, which gives the following constraint:

$$38x_1 + 33x_2 + 39x_3 + 45x_4 + 23x_5 + 27x_6 \leq 100$$



**(ii) Decision Variables**

A certain investment opportunity can either be taken or not taken. One cannot take and not take an opportunity at the same time, and neither can one take an opportunity partially. As such, each decision variable  $x_i$  is constrained to 0 (investment not taken) or 1 (investment taken).

$$x_i = \{0, 1\} \quad \forall i = 1, \dots, 6$$

**(iii) Mutually Exclusive Investments**

Investment opportunities 1 and 2 are mutually exclusive, and so are 3 and 4. In other words, one can only take investment opportunity 1 or 2 but not both. This gives the constraint:

$$x_1 + x_2 \leq 1$$

This constraint is true because it restricts  $x_1 = 1, x_2 = 1$  from the feasible region (as  $1 + 1 = 2 \not\leq 1$ ). Meanwhile, it allows for  $x_1 = 1, x_2 = 0$ ,  $x_1 = 0, x_2 = 1$ , and  $x_1 = 0, x_2 = 0$  to take place, as these are valid values under the constraint.

By a similar logic, we have the following constraint to describe the mutual exclusivity of 3 and 4.

$$x_3 + x_4 \leq 1$$

**(iv) A Special Constraint**

Neither 3 nor 4 can be undertaken unless one of the first two opportunities is undertaken. This gives the constraint:

$$x_1 + x_2 \geq x_3 + x_4$$

This constraint is true because it restricts  $x_3$  and  $x_4$  to 0 when  $x_1 + x_2 = 0$ . In other words, the constraint restricts us from taking 3 and 4 when neither 1 nor 2 is undertaken.

$$\begin{aligned} 0 &\geq x_3 + x_4 \\ \because x_3, x_4 &\in \{0, 1\} \\ \therefore x_3 = x_4 &= 0 \end{aligned}$$

Meanwhile, if either  $x_1$  or  $x_2$  is undertaken, then  $x_1 + x_2 = 1$  and this constraint reduces into the mutual exclusivity constraint:

$$1 \geq x_3 + x_4$$

Finally, if both  $x_1$  and  $x_2$  are undertaken, the constraint turns into the following general constraint:

$$2 \geq x_3 + x_4$$

This constraint is true under any condition since  $x_3, x_4 \in \{0, 1\}$  and thus  $x_3 + x_4$  can never exceed 2.

## Objective Function

The model maximizes the following objective function:

$$z = 15x_1 + 12x_2 + 16x_3 + 18x_4 + 9x_5 + 11x_6$$

This objective function reflects the total estimated profit resulting from the combination of investment opportunities selected. Each component is simply the estimated profit for each investment opportunity multiplied by 0 or 1, depending on whether or not the investment opportunity is undertaken.

## Question 6

The following is the given formulation:

$$\min \sum_{i,j=0}^n c_{(i,j)} x_{(i,j)} \quad (1)$$

$$\text{s.t.} \sum_{i=1}^n x_{(0,i)} \leq M \quad (2)$$

$$\sum_{j=0}^n x_{(i,j)} = 1, \quad \forall i = 1, \dots, n \quad (3)$$

$$\sum_{i=0}^n x_{(i,j)} = 1, \quad \forall j = 1, \dots, n \quad (4)$$

$$x_{(i,j)} \in \{0, 1\}, \quad \forall i, j = 1, \dots, n, \quad i \neq j \quad (5)$$

$$s_0 = 0 \quad (6)$$

$$w_i \leq s_i \leq Q, \quad \forall i = 1, \dots, n \quad (7)$$

$$s_i - s_j + w_j \leq Q(1 - x_{(i,j)}), \quad \forall i, j = 1, \dots, n \quad (8)$$

(a)

Step by step, we will break down the model and explain why it is true.

### Decision Variables

The model involves only the decision variables  $x_{(i,j)}$ , which is defined as:

$$x_{(i,j)} = \begin{cases} 1, & \text{if a truck moves from node } i \text{ to node } j \\ 0, & \text{otherwise} \end{cases}, \quad \forall i, j \in \{0, \dots, n\}, \quad i \neq j$$

The restriction  $i \neq j$  is necessary as no truck should visit and leave from the same client.

## Constraints

### (i) Each Client is Visited Once

Each client is visited once because of the pair of constraints:

$$\sum_{i=0}^n x_{(i,j)} = 1, \quad \forall j = 1, \dots, n$$

$$\sum_{j=0}^n x_{(i,j)} = 1, \quad \forall i = 1, \dots, n$$

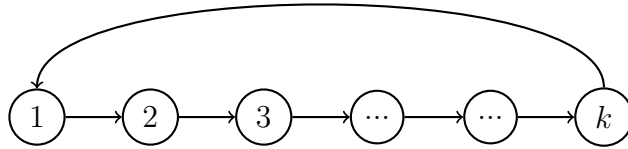
The first constraint ensures that there will only be 1 truck that leaves each node  $i$ . Meanwhile, the second constraint ensures that there will only be 1 truck that visits each node  $j$ . Both constraint ensures that there will only be 1 truck that visits and leaves from each node. As such, each client is visited exactly once.

### (ii) Disallowing Subtours

The constraint

$$s_i - s_j + w_j \leq Q(1 - x_{(i,j)}), \quad \forall i, j = 1, \dots, n$$

ensures that no subtours are formed. This is achieved by preventing the accumulation of weight exceeding the capacity of the vehicle,  $Q$ , when transitioning from node  $i$  to node  $j$ . To understand this, consider a subtour which does not visit the depot, consisting of nodes  $\{1, 2, 3, \dots, k, 1\}$ .



Node 1 points to 2, which points to 3, which points to 4 and so on until we reach node  $k$ , which ultimately points back at 1. As such:

$$x_{(1,2)} = x_{(2,3)} = \dots = x_{(k-1,k)} = x_{(k,1)} = 1$$

Because of the value that these decision variables take, the inequality  $s_i - s_j + w_j \leq Q(1 - x_{(i,j)})$  becomes:

$$\begin{aligned} s_1 - s_2 + w_2 &\leq 0 \\ s_2 - s_3 + w_3 &\leq 0 \\ s_3 - s_4 + w_4 &\leq 0 \\ &\vdots \\ s_{k-1} - s_k + w_k &\leq 0 \\ s_k - s_1 + w_1 &\leq 0 \end{aligned}$$

Adding all these inequalities result in:

$$w_1 + w_2 + \dots + w_{k-1} + w_k \leq 0$$

This, however, cannot be true as  $w_i > 0$  for every  $i$  (each client  $i$  has a parcel of positive weight  $w_i$ ). As such, by proof of contradiction, we conclude that there cannot be a subtour which does not visit the depot.

### (iii) Decision Variables

Following the definition of the decision variables,  $x_{(i,j)}$  is either 0 or 1 for every  $i$  and  $j$ . As such we have the constraint:

$$x_{(i,j)} \in \{0, 1\}, \quad \forall i, j \in \{0, \dots, n\}, \quad i \neq j$$

### (iv) Limit of Trucks

There cannot be more trucks sent than what the company owns. As such, we have the constraint:

$$\sum_{i=1}^n x_{(0,i)} \leq M$$

where  $M$  is the number of trucks the company owns. Essentially, all trucks sent from the depot ( $i = 0$ ) cannot exceed more than the company's number of trucks. As shown in (ii), all trucks have to go through the depot. So by limiting all trucks that go through the depot to at most  $M$ , we are limiting all trucks used overall to at most  $M$ .

### (v) Truck Capacity Constraint

At any point in time after visiting client  $i$ , the accumulated weight of a truck,  $s_i$ , must not exceed its total capacity  $Q$ . In addition, its accumulated weight at  $i$  must at least equal the weight it picked up from client  $i$ . Thus the constraint:

$$w_i \leq s_i \leq Q, \quad \forall i = 1, \dots, n$$

The constraint  $s_i - s_j + w_j \leq Q(1 - x_{(i,j)})$ ,  $\forall i, j = 1, \dots, n$  does not change our conclusion because the constraint produces implications that are consistent with the problem formulation for all  $x_{(i,j)}$ .

Firstly, when  $x_{(i,j)} = 0$ , the constraint transforms into the generally true constraint:

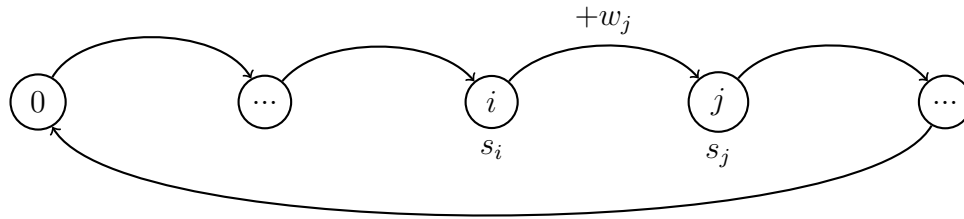
$$\begin{aligned} x_{(i,j)} = 0 &\implies s_i - s_j + w_j \leq Q \\ &\implies s_i \leq Q + (s_j - w_j) \end{aligned}$$

From constraint 7,  $s_j - w_j \geq 0$  and as such the above constraint is simply a relaxation of  $s_i \leq Q$  (also from constraint 7).

Secondly, when  $x_{(i,j)} = 1$ , the truck immediately visits client  $j$  from client  $i$ , and the constraint simplifies into:

$$\begin{aligned} x_{(i,j)} = 1 &\implies s_i - s_j + w_j \leq 0 \\ &\implies s_i \leq s_j - w_j \end{aligned}$$

This is true as the cumulative weight of the truck at  $i$  must not exceed its cumulative weight at  $j$  without the weight picked up at  $j$ . Consider the following illustration.



As such, we know with certainty that the inequality  $s_i - s_j + w_j \leq Q(1 - x_{(i,j)})$  does not contradict the truck capacity constraint.

### (vi) Starting Depot

The truck starts from the depot ( $i = 0$ ) with an empty container. As such, we declare:

$$s_0 = 0$$

This constraint is critical to ensure that any solution produced will be realistic.

### Objective Function

Finally, we have the objective function. The objective of this problem is to minimize the total distance travelled by all vehicles. As such, the objective function computes the total distance travelled by all vehicles and can simply be described as:

$$Z = \sum_{i,j=0}^n c_{(i,j)} x_{(i,j)} \quad \text{for } i \neq j$$

where  $c_{(i,j)}$  denote the distance between client  $i$  and  $j$ .

### (b) Coding Question

The following code was used to formulate the VRP with random data. The data is generated using the Random package with seed 1234 to enable reproducibility.

```
1 using Random
2 seed_value = 1234
3 Random.seed!(seed_value)
```

```

4
5 random_int = rand(1:100)
6 n = rand(5:10)
7 M = rand(2:30)
8 Q = rand(80:200)
9 w = rand(5:100, (1, n+1))
10 c = rand(5:1000, (n+1, n+1))

```

Listing 1: Generating Random Data

Using this seed, we obtain  $n = 8$ ,  $M = 8$ ,  $Q = 188$ , and:

$i$	0	1	2	3	4	5	6	7	8
$w_i$	38	42	96	81	52	76	60	74	5

Table 1:  $w_i$  Generated Using Random Package

$i \setminus j$	0	1	2	3	4	5	6	7	8
0	203	567	369	847	708	717	7	674	198
1	442	257	135	352	445	194	139	482	943
2	684	629	947	770	886	379	534	951	671
3	957	238	577	272	523	568	77	31	893
4	650	129	679	542	943	176	504	350	360
5	997	612	574	297	812	331	908	820	731
6	751	675	77	246	467	634	121	794	404
7	114	763	703	979	978	377	370	968	318
8	494	591	99	854	503	730	882	474	806

Table 2:  $c_{(i,j)}$  Generated Using Random Package

Now, we can formulate the model.

```

1 using JuMP, GLPK
2 # INITIALIZE MODEL
3 m = Model(GLPK.Optimizer)
4
5 # VARIABLES
6 @variable(m, x[1:n+1, 1:n+1], Bin)
7 # Force x[i, j] where i=j to equal 0
8 for i in 1:n+1
9     for j in 1:n+1
10         if i == j
11             @constraint(m, x[i, j] == 0)
12         end
13     end
14 end
15 @variable(m, s[1:n+1])
16
17 # CONSTRAINTS
18 # Constraint 1: Truck Capacity Constraint

```

```
19 @constraint(m, sum(x[1, i] for i in 1:n+1) <= M)
20
21 # Constraint 2: Each Client is Visited Just Once
22 for i in 2:n+1
23     @constraint(m, sum(x[i,:]) == 1)
24 end
25 for j in 2:n+1
26     @constraint(m, sum(x[:,j]) == 1)
27 end
28
29 # Constraint 3: Weight at Depot is 0
30 @constraint(m, s[1] == 0)
31
32 # Constraint 4: Weights & Subtour Elimination Constraint
33 for i in 2:n+1
34     @constraint(m, s[i] >= w[i])
35     @constraint(m, s[i] <= Q)
36     for j in 2:n+1
37         @constraint(m, s[i] - s[j] + w[j] <= Q * (1 - x[i,j]))
38     end
39 end
40
41 # OBJECTIVE FUNCTION
42 @objective(m, Min, sum(c[i, j] * x[i, j] for i in 1:n+1, j in 1:n+1))
43
44 # PRINT THE MODEL
45 println(m)
```

Listing 2: Model Formulation

To solve the problem, run the following code.

```
1 optimize!(m)
2 println("Objective value: ", objective_value(m))
3 println("Optimal solution:")
4 println(value.(x))
5 solution_summary(m)
```

Listing 3: Model Implementation

The following were the results.

```

Objective value: 3691.0
Optimal solution:
[0.0 0.0 0.0 1.0 0.0 0.0 1.0 0.0 1.0;

* Solver : GLPK

* Status
  Result count      : 1
  Termination status : OPTIMAL
  Message from the solver:
  "Solution is optimal"

* Candidate solution (result #1)
  Primal status      : FEASIBLE_POINT
  Dual status        : NO_SOLUTION
  Objective value     : 3.69100e+03
  Objective bound     : 3.69100e+03
  Relative gap        : 1.08372e-03

* Work counters
  Solve time (sec)    : 5.00000e-02

```

Figure 8: Results

$i \setminus j$	0	1	2	3	4	5	6	7	8
0	0	0	0	1	0	0	1	0	1
1	1	0	0	0	0	0	0	0	0
2	1	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	1	0
4	0	0	0	0	0	1	0	0	0
5	0	1	0	0	0	0	0	0	0
6	0	0	1	0	0	0	0	0	0
7	1	0	0	0	0	0	0	0	0
8	0	0	0	0	1	0	0	0	0

Table 3: Optimal Values of  $x_{(i,j)}$ 

$s_0$	$s_1$	$s_2$	$s_3$	$s_4$	$s_5$	$s_6$	$s_7$	$s_8$
0	188	188	81	57	146	60	155	5

Table 4: Optimal Values of  $s_i$ 

### (c) Vehicle Routing Problem with Time Windows

Step by step, we will form a model to minimize the total travelled time by all vehicles. This model will be based on the model formulated in (a) but with several changes:



### Decision Variables

Firstly, the decision variables  $x_{(i,j)}$  remain the same as previously defined:

$$x_{(i,j)} = \begin{cases} 1, & \text{if a truck moves from node } i \text{ to node } j \\ 0, & \text{otherwise} \end{cases}, \quad \forall i, j \in \{0, \dots, n\}, \quad i \neq j$$

Secondly, introduce decision variable  $t_i$  which denote the time that a vehicle starts serving client  $i$ .

### Objective Function

The objective of the problem is to minimize the total travelled time by all trucks. As such, we can simply use the objective function formulated in (a):

$$Z = \sum_{i,j=0}^n c_{(i,j)} x_{(i,j)} \quad \text{for } i \neq j$$

In this case, however,  $c_{(i,j)}$  does not denote the distance between client  $i$  and  $j$ , but rather the time to travel from client  $i$  to  $j$ .

### Constraints

#### (i) Time Window Constraint

Each client  $i$  can only be serviced throughout a time window of  $[a_i, b_i]$ . As such, we have the constraint:

$$a_i \leq t_i \leq b_i \quad \forall i = 1, \dots, n$$

#### (ii) Starting Time

Declare the starting time  $t_0$  as 0.

$$t_0 = 0$$

#### (iii) Eliminating Subtours

Similar to part (a), we need to impose a constraint which eliminates subtours which do not visit the depot. We can do this by creating the following constraint:

$$t_i - t_j + c_{(i,j)} \leq b_i(1 - x_{(i,j)}) \quad \forall i, j = 1, \dots, n$$

Firstly, when  $x_{(i,j)} = 0$ , the constraint transforms into the generally true constraint:

$$\begin{aligned} x_{(i,j)} = 0 &\implies t_i - t_j + c_{(i,j)} \leq b_i \\ &\implies t_i \leq b_i + (t_j - c_{(i,j)}) \end{aligned}$$

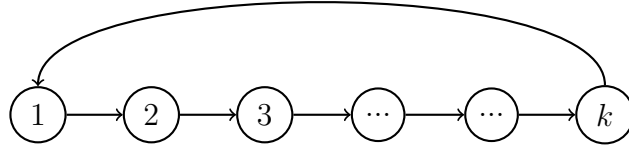
This constraint is generally true as  $t_i \leq b_i$  and  $t_j - c_{(i,j)} \geq 0$  for all  $i$  and  $j$ . So  $t_i \leq b_i + (t_j - c_{(i,j)})$  is simply a relaxation of this constraint.

Secondly, when  $x_{(i,j)} = 1$ , the truck immediately visits client  $j$  from client  $i$ , and the constraint simplifies into:

$$\begin{aligned} x_{(i,j)} = 1 &\implies t_i - t_j + c_{(i,j)} \leq 0 \\ &\implies t_i \leq t_j - c_{(i,j)} \end{aligned}$$

This is true because if the truck visits client  $j$  immediately after serving client  $i$ , the time at which the truck serves customer  $i$  must not be after  $t_j - c_{(i,j)}$ , which is the time exactly at which the truck proceeds to the next customer from client  $i$ .

To prove that this inequality accurately eliminates all subtours, we can derive a similar proof as in part (a). Consider a subtour which does not visit the depot, consisting of nodes  $\{1, 2, 3, \dots, k, 1\}$ .



In this particular subtour:

$$x_{(1,2)} = x_{(2,3)} = \dots = x_{(k-1,k)} = x_{(k,1)} = 1$$

and as such the inequality  $t_i - t_j + c_{(i,j)} \leq b_i(1 - x_{(i,j)})$  becomes:

$$\begin{aligned} t_1 - t_2 + b_2 &\leq 0 \\ t_2 - t_3 + b_3 &\leq 0 \\ t_3 - t_4 + b_4 &\leq 0 \\ &\vdots \\ t_{k-1} - t_k + b_k &\leq 0 \\ t_k - t_1 + b_1 &\leq 0 \end{aligned}$$

Adding all these inequalities result in:

$$b_1 + b_2 + \dots + b_{k-1} + b_k \leq 0$$

This, however, cannot be true as  $b_i > 0$  for every  $i$  (since a timing such as 4 PM cannot be negative). As such, by proof of contradiction, we conclude that there cannot be a subtour which does not visit the depot under the inequality  $t_i - t_j + c_{(i,j)} \leq b_i(1 - x_{(i,j)})$ .

#### (iv) Other Constraints

The rest of the constraints from the model in (a) will reappear in this model due to the

similarity of both problems. As such, we have the following constraints:

$$\begin{aligned} \text{s.t. } & \sum_{i=1}^n x_{(0,i)} \leq M \quad (\text{where } M \text{ is the number of trucks available}) \\ & \sum_{j=0}^n x_{(i,j)} = 1, \quad \forall i = 1, \dots, n \\ & \sum_{i=0}^n x_{(i,j)} = 1, \quad \forall j = 1, \dots, n \end{aligned}$$

The first constraint ensures that the number of trucks sent from the depot do not exceed  $M$ , the maximum number of available trucks. The second and third constraint ensures that each client is visited exactly once.

### Formulation

That being said, we have the following as our model formulation for the VRPTWs:

$$\begin{aligned} \min & \sum_{i,j=0}^n c_{(i,j)} x_{(i,j)} \\ \text{s.t. } & \sum_{i=1}^n x_{(0,i)} \leq M \\ & \sum_{j=0}^n x_{(i,j)} = 1, \quad \forall i = 1, \dots, n \\ & \sum_{i=0}^n x_{(i,j)} = 1, \quad \forall j = 1, \dots, n \\ & x_{(i,j)} \in \{0, 1\}, \quad \forall i, j \in \{0, \dots, n\}, \quad i \neq j \\ & t_0 = 0 \\ & a_i \leq t_i \leq b_i, \quad \forall i = 1, \dots, n \\ & t_i - t_j + c_{(i,j)} \leq b(1 - x_{(i,j)}), \quad \forall i, j = 1, \dots, n \end{aligned}$$