MS-E2121 Homework 6

Tuomas Porkamaa

March 2022

Problem 6.1: Knapsack covers

Let

$$S = \left\{ x \in \{0,1\}^5 : 79x_1 + 5x_2 + 53x_3 + 45x_4 + 45x_5 \leq 143 \right\} \tag{1}$$

(a)

Minimal covers for S:

$$\begin{aligned} C_1 &= \{2,3,4,5\} \\ C_2 &= \{1,3,4\} \\ C_3 &= \{1,3,5\} \\ C_4 &= \{1,4,5\} \end{aligned}$$

(b)

Let $\bar{x} = (\frac{1}{5}, \frac{2}{5}, 1, 1, \frac{3}{5})$. The separation problem for finding violated inequality for $\bar{x}$ is

$$\begin{aligned} \text{min.} \quad & \sum_{j \in C} (1 - \bar{x}_j) \\ \text{s.t.} \quad & \sum_{j \in C} a_j > b \end{aligned} \quad \Leftrightarrow \quad \begin{aligned} \text{min.} \quad & \sum_{j \in N} z_j (1 - \bar{x}_j) \\ \text{s.t.} \quad & \sum_{j \in N} z_j a_j > b \\ & z \in \{0,1\}^n \end{aligned} \tag{2}$$

By applying $\bar{x}$ and definition of S to (2), the problem becomes:

$$\begin{aligned} \text{min.} \quad & z_1 \frac{4}{5} + z_2 \frac{3}{5} + z_3 0 + z_4 0 + z_5 \frac{2}{5} \\ \text{s.t.} \quad & z_1 79 + z_2 5 + z_3 53 + z_4 45 + z_5 45 > 143 \\ & z \in \{0,1\}^5 \end{aligned} \tag{3}$$

The objective and constraint indicate that the optimal solution is one of the minimal covers of S given in (a). Because $\frac{3}{5} + \frac{2}{5} > \frac{4}{5}$, this cuts out C_1. From

remaining covers, it can be easily seen that solution is C_2 since for all covers, $z_1 = 1$ and the costs associated with other values of z are zero. Therefore the solution for (3) is $\bar{z} = (1, 0, 1, 1, 0)$, or cover C_2, and objective value is $\frac{4}{5}$.

Furthermore, valid cover inequalities for S are form of

$$\sum_{j \in C} x_j \leq |C| - 1 \Leftrightarrow \sum_{j \in C} (1 - x_j) \geq 1 \tag{4}$$

Therefore, $\bar{x}$ violates (4) associated with C_2 since

$$\sum_{j \in C_2} \bar{x}_j = \bar{x}_1 + \bar{x}_3 + \bar{x}_4 = \frac{1}{5} + 1 + 1 = 2.2 > 2 = |C_2| - 1$$

i.e. , the violated cover inequality is

$$x_1 + x_3 + x_4 \leq 2$$

(c)

Consider cover C_1 from (a). The extended cover $E(C_1)$ equals to $E(C_1) = \{1, 2, 3, 4, 5\}$ and the extended cover inequality becomes $x_1+x_2+x_3+x_4+x_5 \leq 3$. Therefore this inequality cuts off $\bar{x}$ since $\bar{x}_1 + \bar{x}_2 + \bar{x}_3 + \bar{x}_4 + \bar{x}_5 = 3.2$.

Problem 6.2: Improving TSP formulation with valid inequalities

(a)

Inequality (2):

Assume that (2) $\equiv x_{ij} + x_{ji} \leq 1, \forall i, j \in N : i \neq j$ is invalid, i.e., $\exists i, j \in N, i \neq j$ such that $x_{ij} + x_{ji} > 1$. Because $x_{ij} \in \{0, 1\}$, this means that $x_{ij} + x_{ji} = 2$. This indicates that there is a subtour between cities i and j. However, this is contradiction, since constraint (1) prevents all subtours. Therefore (2) is a valid inequality for MTZ.

Inequality (3):

Let $x_{ij}, x_{ji} \in X$, i.e. $u_i - u_j + (n-1)x_{ij} \leq u_i - u_j + n - 1 \leq n - 1$ holds. Additionally, because constraint (1) prevents subtours, then either $x_{ij} = 0$ & $x_{ji} = 0$, $x_{ij} = 1$ & $x_{ji} = 0$ or $x_{ij} = 0$ & $x_{ji} = 1$ holds. For these value combinations, the following inequality holds

$$u_i - u_j + (n-1)x_{ij} + (n-3)x_{ij} \leq u_i - u_j + n - 1$$

Therefore inequality (3) is also satisfied, i.e., (3) is valid.

Inequality (4):

Let $x_{1j} \in X$, i.e. $(n-2)x_{1j} \leq n-2$. Therefore

$$u_j - 1 + (n-2)x_{1j} \leq u_j - 1 + n - 2 = 2 - 1 + n - 2 = n - 1$$

The second rightmost equality is derived as follows. Without loss of generality, we can select the starting city of TSP tour freely, i.e. city 1 is the first, from which it follows that the j:th city is the second city in the tour meaning that $u_j = 2$.

Therefore inequality (4) is also satisfied, i.e., (4) is valid.

Inequality (5):

Let $x_{i1} \in X$, i.e. $(n-1)x_{i1} \leq n-1$. Therefore

$$1 - u_i + (n-1)x_{i1} \leq 1 - u_i + n - 1 = 1 - n + n - 1 = 0$$

The second rightmost equality is derived as follows. Without loss of generality, we can select the starting city of TSP tour freely, i.e. city 1 is the first, from which it follows that that the i:th city is the last (n:th) in the tour meaning that $u_i = n$.

(b)

The running time for original problem without extra inequalities was 46.29 seconds. After including the inequalities presented in (a) the running time dropped to 0.74 seconds. The performance increase is indeed notable.

Problem 6.3: Vehicle routing problem with time windows

Consider additional parameters to capacitated vehicle routing problem:

- T_j^{LB} : Lower bound, minimal time, for the delivery in client j
- T_j^{LB} : Upper bound, maximal time, for the delivery in client j
- T_{ij} : Transportation time from node i to node j
- S_i : Service time for node i

(a)

The problem can be formulated as follows:

$$
\begin{aligned}
\min \quad & \sum_{(i,j)\in A} C_{i,j} x_{i,j} \\
\text{s.t.:} \quad & \sum_{j\in V, j\neq i} x_{i,j} = 1, \ \forall i \in N \\
& \sum_{i\in V, i\neq j} x_{i,j} = 1, \ \forall j \in N \\
& u_i - D_j \leq u_j + M(1 - x_{i,j}), \ \forall i \in V, j \in N \\
& u_i - D_j \geq u_j - M(1 - x_{i,j}), \ \forall i \in V, j \in N \\
& 0 \leq u_i \leq Q, \ \forall i \in V \\
& u_1 = Q \\
& a_j \geq a_i + S_i + T_{ij} - M(1 - x_{ij}), \ \forall i, j \in N \qquad (5) \\
& a_j \geq a_1 + T_{1j} - M(1 - x_{1j}), \ \forall j \in N \qquad (6) \\
& T_j^{LB} \leq a_j \leq T_j^{UB}, \ \forall j \in N \qquad (7) \\
& a_1 = 0 \qquad (8) \\
& x_{i,j} \in \{0, 1\}, \ \forall i \in V, \forall j \in V
\end{aligned}
$$

In the updated CVRP-formulation given above, a new variable a_i is introduced associated with constraints (5) - (8). Variable a_i models the arrival time to node i. It is assumed that arrival time for depot node is set to 0 (constraint (8)), but practically it can be any non-negative number, since it models the time when cargo is ready to be distributed to clients. For every client node $j \in N$, the global arrival time must be greater than the arrival time to preceding node $i \in V$ (can be either depot or client), plus the service time in that node and the time to travel from node i to node j. This relation between arrival times is splitted to two parts, i.e. constraints (5) and (6), where (5) models the interaction between client nodes, and (6) between depot and client. In the implementation we have not defined the service time for depot node and hence the constraint (6) assumes that the service time for depot node is 0. The big-M notation ensures that the relation between arrival times exists only if there is a connection between nodes. Lastly, for each client node, the arrival time must be in between predefined time window. This behaviour is modelled via constraint (7).

(b)

Implementation and results in *hw6_skeleton.ipynb*

(c)

For the last part, problem given in (a) is solved using larger dataset and different solver configurations. In experiments we use CBC-solver that implements branch-and-cut algorithm for MIP-problems. Following configurations are considered.

1 Default configuration.

2 Presolving turned off.

3 Cuts turned off.

4 Heuristics turned off.

5 Using lots of different cuts and heuristics.

In the default solver setting the presolving methods are employed before solving the LP-relaxation of the problem. Generally the cut generations are also used, but several advanced methods left to CBC to decide if they are needed, instead forcing them on or off. Most of the primal heuristics are also turned off, especially those that are employed in configuration 5.

Regardless on the configuration of the solver, in the first phase the linear relaxation of the problem is solved and initial dual bound is obtained. For each of the configurations 1-5, the obtained dual bound was 3894.8 that is really optimistic compared to the actual optimal value that is given in Table 2. Regardless of the used configuration, the preprocessing step is also performed to reduce the problem size. After these steps, the solver applies Feasibility pump method to search for an initial feasible result. This part is not performed for configuration 4. Lastly, the configuration 5 has several additional heuristics activated, but based on the CBC output log it seems that these extra heuristics didn't provide any improvement on the primal feasible solution.

After the preprocessing steps, the actual Branch-and-Cut procedure to solve the problem begins. The Table 1 gives some central results about the configurations related this.

Table 1:

Configuration	Visited nodes	Solution found	Root node value before cuts	#Cuts	RNV after cuts
1	395	290	20664.4	79	21299.23
2	633	622	20617.8	86	21292.85
3	31903	-	20664.4	0	20664.4
4	434	188	20664.4	76	21305.9
5	162	141	20664.4	94	21341

The *Visited nodes* value gives a rough estimate how fast the solution was obtained from Branch-and-Cut tree. The *Solution found* value tells the number of visited nodes until the optimal solution for the problem was found. Clearly

the configuration 3 has hardest time traveling within the tree, since the cut generations were disabled meaning that the problem formulation was not strengthened at all. This would yield an relatively large problem to solve for Branch and Bound, and the effect can be seen from the visited nodes count. This configuration didn't found the optimal solution either. For configuration 5, some the advanced cutting methods seem to work relatively well since they managed to create a better problem formulations, which eventually leads to the need of visiting less nodes to find the optimal solution. There were, however, also some cuts that certain additional cutting methods were not that useful (can be seen from the log). The disable of presolving also seems to have some negative effect on the count of nodes needed to visit, as it can be seen from the row corresponding to configuration 2. The updated dual bounds presented in the column 'RNV after cuts' also indicates that less nodes needs to be visited if the gap between primal and dual bounds is smaller.

Lastly, the final solving times for each configuration are presented below.

Table 2:

Configuration	Objective value	Solution time
1	21673.79	19.14s
2	21673.79	24.59s
3	21708.31*	298.29s
4	21673.79	19.48s
5	21673.79	288.21s

The configuration where cuts were turned off (3) didn't provide optimal solution, since the time limit exceeded. However, it accomplished to produce interval of $[21569.53, 21708.31]$ where the optimal solution lies. The default configuration (1) was the fastest method for this instance, while configurations without presolving (2) and primal heuristics (4) also finished relatively fast. Among the configurations that succeeded to solve the problem, the configuration that utilizes majority of available heuristics and cuts (5) was slowest one with a large marginal. That may be the case because applying advanced heuristics in preprocessing stage greatly increases the time spent before the Branch-and-Cut algorighm even starts. More effective method may be to rely on simpler or/and less heuristics. Overall, the default configuration seem to have good balance how to organize the whole solving pipeline.