DM872 assignment 2

sagra16
(Valdemar Grange - 081097-2033)

June 2020

It should preemptively be noted that the `ipynb` notebook supplied along side with the code shows results and the way the code was developed.

Task 1

The (capacitated vehicle routing problem) CVRP can be formulated as a set-partitioning problem as follows.

$$\begin{aligned}
\text{minimize} \quad & \sum_{r \in \Omega} c_r \theta_r \\
\text{subject to} \quad & \sum_{r \in \Omega} a_{i,r} \theta_r = 1, \quad \forall i \in N \\
& \sum_{r \in \Omega} \theta_r \leq m \\
& \\
& \theta_r \in \{0, 1\}, \qquad \forall r \in \Omega
\end{aligned}$$

Compared to the tasked problem we need a couple of things to complete the model. We implicitly only allow routes that satisfy the capacity limit by only allowing such sets in the problem.

The "additional" aspect is the time windows, which we can model quite easily by the following constraint.

$$\theta_r \, a_{i,r} \, q_{i,r} \leq l_i, \qquad \forall i \in N, \, \forall r \in \Omega$$

We let $q_{i,r}$ denote the time it takes to reach customer i in route r, where we include accounting for waiting for each customer in the route to be ready (eg the time window beginning). This can simply be implemented by observing `makeQ` from the source code. Observe that we can realize that this constraint can be applied before solving the model, meaning that effectively we can filter out solutions that do not satisfy this constraint. The optimal solution yields an objective value of 41.

Task 2

By introducing linear relaxation we can find a good, but not necessarily optimal solution. We perform linear relaxation by relaxing the binary constraint, such that our model becomes.

$$\begin{aligned}
\text{minimize} \quad & \sum_{r\in\Omega} c_r\theta_r \\
\text{subject to} \quad & \sum_{r\in\Omega} a_{i,r}\theta_r = 1, \quad \forall i \in N \\
& \sum_{r\in\Omega} \theta_r \leq m \\
& \\
& \theta_r \geq 0, \qquad\qquad \forall r \in \Omega
\end{aligned}$$

The result from the above model yields $m = 3$ which is not a violation of the original problem, furthermore an objective value of 41, which is in fact the same as the IP model. Luckily we did reach an integer solution which must mean that our polytope's best value is also the integer solution! I tried to not preprocess the time-window constraint, which yielded an infeasible IP solution such that it needed rounding but luckily the LP solution. If we inspect the produced routes:

$$\begin{bmatrix} [0 & 1 & 0 & 1 & 0 & 0 & 0 & 0] \\ [1 & 0 & 1 & 0 & 0 & 0 & 0 & 1] \\ [0 & 0 & 0 & 0 & 1 & 1 & 1 & 0] \end{bmatrix}$$

Which in fact is exactly the same as the IP solution:

$$\begin{bmatrix} [0 & 1 & 0 & 1 & 0 & 0 & 0 & 0] \\ [1 & 0 & 1 & 0 & 0 & 0 & 0 & 1] \\ [0 & 0 & 0 & 0 & 1 & 1 & 1 & 0] \end{bmatrix}$$

A Lagrangian relaxation might fit better in some more complicated case where LP relaxation would not yield a good enough solution, since lagrangian is both tighter and does not pose so many questions about solution correctness. I will relax the amount of present vehicles.

$$\begin{aligned}
& f(\lambda) = \\
\text{minimize} \quad & \sum_{r\in\Omega} c_r\theta_r - \lambda\left(\sum_{r\in\Omega}\theta_r - m\right) \\
\text{subject to} \quad & \sum_{r\in\Omega} a_{i,r}\theta_r = 1, \qquad\qquad \forall i \in N \\
& \\
& \theta_r \in \{0, 1\}, \qquad\qquad \forall r \in \Omega
\end{aligned}$$

To solve the problem finding the best λ for the lagrangian relaxation, I will use the Held and Karp method.

$$\theta = \mu \frac{z_{LR}(\lambda^k) - z}{\sum_i \lambda_i^2}$$

The solution found from the gradient decent with 10 iterations and $\mu = 2$ was $\lambda = 0$, the code can be viewed for my solution. The actual solution reached an objective of 41 with the same three rows as with the IP and LP relaxed models.

Task 3

Like in task 2, the master problem is actually just the linear relaxation of the original problem.

$$\begin{aligned} \text{minimize} \quad & \sum_{r\in\Omega} c_r\theta_r \\ \text{subject to} \quad & \sum_{r\in\Omega} a_{i,r}\theta_r = 1, \quad && \forall i \in N \\ & \sum \theta_r \leq m \\ & \theta_r \geq 0, && \forall r \in \Omega \end{aligned}$$

The restricted master problem is a subset of the master problem where we do not yet consider a majority of the available problem. We denote this arbitrary subset by $\hat{\Omega} \subset \Omega$.

$$\begin{aligned} \text{minimize} \quad & \sum_{r\in\hat{\Omega}} c_r\theta_r \\ \text{subject to} \quad & \sum_{r\in\hat{\Omega}} a_{i,r}\theta_r = 1, \quad && \forall i \in N \\ & \sum \theta_r \leq m \\ & \theta_r \geq 0, && \forall r \in \hat{\Omega} \end{aligned}$$

We also need to develop a pricing model for evaluating the sub-problem. The sub-problem should minimize the reduced cost, where the negative reduction means that the column we are inspecting will enter the basis. In fact if a non-negative reduced cost is found, the solution is optimal. We must construct the dual to our problem, to determine the reduced cost. The dual variables can be observed by inspecting the primal tableau. In the problem we have two constraints, such that we have two sets of dual variables. One of the constraints is a singleton (the $\leq m$ constraint), which I will denote λ_0. The second set of constraints is over $i \in N$, so I will denote them $\lambda_{2,r}$. Our reduced cost of a variable θ_r can be realized by $\hat{c}_r = c_r - \lambda_T A_r$. In fact, constraints will be a valid pathing and with the reduction in cost it becomes a shortest path problem. Every vertex must exactly consume one edge and produce one, or rather every vertex must be the sink of exactly one edge and the source of one. Additionally we use big-M to model capacity constraints, and include the time window constraint. Let q'_i be the function that determines the accumulated time for some vertex i, then we can determine the time-window constraint by multiplying with the path activation variable $x_{i,j}$. Finally let A be the set of edges, V be the set of vertices, Q the the max capacity and $x_{i,j}$ be the activation variable of path i, j.

$$\begin{array}{lll}
\text{minimize} & \displaystyle\sum_{i,j \in A} (c_{i,j} - \lambda_{2,i}) x_{i,j} - \lambda_0 & \\
\text{subject to} & \displaystyle\sum_{i,k \in A} x_{i,k} = \sum_{k,j \in A} x_{k,j} & \forall k \in N \\
& \displaystyle\sum_{0,j \in A} x_{0,j} = 1 & \\
& \displaystyle\sum_{i,0 \in A} x_{i,0} = 1 & \\
& \displaystyle\left(\sum_{(i,j) \in A} x_{i,j} \right) q'_i \leq l_i & \forall i \in V \\
& & \\
& y_i + q_j + M x_{i,j} \leq y_j + M & \forall i, j \in A : j \neq 0 \\
& 0 \leq y_i \leq Q & \forall i \in N \\
& x_{i,j} \in \{0, 1\} & \forall i, j \in A
\end{array}$$

Task 4

Initially we have the subset $\hat{\Omega} \subset \Omega$.

$$\begin{bmatrix}
[0 & 0 & 0 & 1 & 0 & 1 & 1 & 0] \\
[0 & 0 & 0 & 0 & 1 & 1 & 0 & 1] \\
[1 & 0 & 0 & 0 & 1 & 0 & 0 & 0] \\
[0 & 1 & 0 & 1 & 1 & 0 & 0 & 0] \\
[1 & 0 & 1 & 0 & 0 & 0 & 0 & 0] \\
[1 & 1 & 0 & 0 & 0 & 0 & 1 & 0] \\
[0 & 1 & 1 & 0 & 0 & 0 & 0 & 1] \\
[0 & 1 & 0 & 1 & 0 & 0 & 0 & 0] \\
[0 & 0 & 1 & 0 & 1 & 0 & 1 & 0] \\
[0 & 0 & 0 & 0 & 1 & 0 & 1 & 1] \\
[0 & 1 & 1 & 1 & 0 & 0 & 0 & 0] \\
[0 & 0 & 0 & 0 & 0 & 1 & 1 & 1]
\end{bmatrix}$$