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.

1 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 in a few ways.

1. By pre-processing the dataset such that only time-feasible routes are considered. (polynomial)
2. By solving a problem for each possible route which minimizes the cost of the given route, satisfying the time window constraint. (an IP problem so NP)

One could also choose to model this in the objective, but as it is a hard constraint I see no reason why pre-processing shouldn't be the solution. I will implement the constraint in the pricing problem, in a later task. Running this model on the supplied dataset yields an objective value of 45 and the following columns.

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

2 Task 2

For this task I will assume that the supplied dataset (`data.routes`) is the entire dataset Ω. 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.

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

Running this on all combinations of the possible routes. The result from the above model yields $m = 7$ which is a violation of the original problem, furthermore an objective value of 41.666. If we inspect the produced routes:

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

The non-zero variables:

θ_0	0.33
θ_1	0.66
θ_4	0.66
θ_5	0.33
θ_7	0.33
θ_9	0.33
θ_{10}	0.33

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.

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

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. Finding the specific sequence of customers to visit is simply a shortest path problem on each vehicle route. Of course the shortest-path algorithm should include the time-window constraint as the cost might not be the same as the travel-time.

This solution is not really interesting and just proves again the point that the vehicle limit constraint doesn't matter that much, since both the LP relaxation picks it and the lagrangian relaxation of the vehicle constraint. We can also see this since the gradient decent picks $\lambda = 0$, which indicates that it doesn't need this term to find a good feasible solution. In fact when I remove the vehicle constraint altogether the solution is still the same.

Therefore I also present a model where the other constraint of all customers being active as the relaxed constraint.

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

Clearly the solver has a much less constrained solution space to pick from. The heuristic solution yields -10560 as the objective value, and a $\lambda = 105$ after 10 iterations and $\mu = 2$. Clearly the solution is completely invalid, naturally this must be because the solver is now instructed in simply finding a solution that minimizes cost and minimizes the amount of active customers. From increasing the iterations it seems like it will just keep growing until infinity.

On either my generated dataset and the supplied it keeps growing the constraint to infinity.

3 Task 3

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

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

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 $\hat{\Omega} \subset \Omega$.

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

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 sets of constraints, such that we have two sets of dual variables. One of the sets of 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 $\lambda_{1,r}$. Our reduced cost of a variable θ_r can be realized by $\hat{c}_r = c_r - \lambda_T A_r$. The problem for pricing a column will be the 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 A be the set of edges, V be the set of vertices including the depot or origin point, $N = V \backslash \{0\}$, Q the the max capacity and $x_{i,j}$ be the activation variable of path i, j. Finally let $y_i \in \mathbb{R}$ be the load on the vehicle when leaving vertex $i \in V$.

To express the "path taken" we must accumulate and check all time window constraints for violation.

$$\sum_{0,j \in A} x_{0,j} t_{0,j} + \sum_{j,i \in A} x_{j,i} t_{j,i} \cdots \sum_{s,z \in A} x_{s,z} t_{s,z} \leq l_z \sum_{s,z \in A} x_{s,z} \quad \forall z \in V$$

As one might notice, this is not easily expressible, as the amount of sum's should be equal to the length of the path from 0 to z.

1. Another could be to check the constraint after the problem has been solved, if the constraint is violated then solve it again with the previously chosen path excluded.

2. A solution could also be to introduce an additional variable like y, but for time window's as this can also be viewed as a resource.

I will use item 2, as it seems the most appropriate (but not simple, that would certainly be item 1). To do this I will introduce a new variable t_i, which will be the time spent travelling to position t_i. t_i should be bounded by $0 \leq t_i \leq l_i$,

where l_i is the latest possible time to arrive. This method uses big-M, denoted by M, and is formulated such that the solver must assign the time-spent correctly to the variable. Additionally we still need to account for waiting before the customer "opens" in case that the vehicle is too early, which is $max(e_i, t_i)$. We can model this by.

$$max(e_j, time_{i,j} + t_i) + Mx_{i,j} \leq t_j + M \quad \forall i, j \in A : j \neq 0$$

Once again there is a problem, max is not linear thus we must find a way around this. We can simply introduce a variable to the model which specifies the time of delivery j given we are coming from i, and the will make sure that this constraint is valid.

$$\alpha_{i,j} \geq e_j \wedge \alpha_{i,j} \geq time_{i,j} + t_i \quad \forall i, j \in A : j \neq 0$$

The solver will pick the smallest value to assign to α that is, the maximum of the two. Note that "serving time" can also easily be modelled here if wished by simply adding it to α. The time constraint can now be expressed more formally as.

$$\begin{array}{ll} \alpha_{i,j} \geq e_j \wedge \alpha_{i,j} \geq time_{i,j} + t_i & \forall i, j \in A : j \neq 0 \\ \alpha_{i,j} + Mx_{i,j} \leq t_j + M & \forall i, j \in A : j \neq 0 \\ 0 \leq t_i \leq l_i & \forall i \in V \\ \\ \alpha \in \mathbb{R} & \end{array}$$

The problem is as follows.

$$\begin{array}{lll} \text{minimize} & \displaystyle\sum_{i,j \in A} (c_{i,j} - \lambda_{1,i}) x_{i,j} - \lambda_0 & \\ \text{subject to} & \displaystyle\sum_{i,k \in A} x_{i,k} = \sum_{k,j \in A} x_{k,j} & \forall k \in N \\ & \displaystyle\sum_{0,j \in A} x_{0,j} = 1 & \\ & \displaystyle\sum_{i,0 \in A} x_{i,0} = 1 & \\ \\ & \alpha_{i,j} \geq e_j \wedge \alpha_{i,j} \geq time_{i,j} + t_i & \forall i, j \in A : j \neq 0 \\ & \alpha_{i,j} + Mx_{i,j} \leq t_j + M & \forall i, j \in A : j \neq 0 \\ & 0 \leq t_i \leq l_i & \forall i \in V \backslash \{0\} \\ \\ & y_i + q_j + Mx_{i,j} \leq y_j + M & \forall i, j \in A : j \neq 0 \\ & 0 \leq y_i \leq Q & \forall i \in N \\ \\ & x_{i,j} \in \{0, 1\} & \forall i, j \in A \\ & \alpha_{i,j} \in \mathbb{R} & \forall i, j \in A \end{array}$$

Notice that we cannot preemptively remove path's to be checked based on time windows, like before. This is because that we actually generate our columns ad-hoc.

4 Task 4

I will begin using a subset of original dataset and then add 4 columns using the path that the pricing problem yields. In practice you would continue until no more reduced cost routes are found, but this may end up being many iterations. Initially $\hat{\Omega} = \{\theta_3, \theta_4, \theta_{11}\}$.

$18\ \theta_3$	$+15\ \theta_4$	$+12\ \theta_{11}$	
	$+\theta_4$		$= 1$
θ_3			$= 1$
	θ_4		$= 1$
θ_3			$= 1$
θ_3			$= 1$
		θ_{11}	$= 1$
		θ_{11}	$= 1$
		θ_{11}	$= 1$
θ_3	$+\theta_4$	θ_{11}	≤ 3
θ_3	$,\theta_4$	θ_{11}	≥ 0

Then we solve the pricing problem with the RMP's duals as parameters and look for a possible column to introduce. We have the initial RMP solution.

$$\theta_3 = 1.00$$
$$\theta_4 = 1.00$$
$$\theta_{11} = 1.00$$
$$cost = 45.00$$
$$\lambda_{1,r} = (15.00, 18.00, 0.00, 0.00, 0.00, 12.00, 0.00, 0.00)$$
$$\lambda_0 = 0.0$$

Solving the pricing problem with the RMP duals, we find the column r with price c_r and $\hat{c} = -18.0$:

$$r = [0, 2, 1, 0]$$
$$c_r = 15$$

λ \ c_r	18	15	12	
15.0	0	1	0	$= 1$
18.0	1	0	0	$= 1$
0.0	0	1	0	$= 1$
0.0	1	0	0	$= 1$
0.0	1	0	0	$= 1$
12.0	0	0	1	$= 1$
0.0	0	0	1	$= 1$
0.0	0	0	1	$= 1$
0.0	1	1	1	≤ 3
θ	1.0	1.0	1.0	

Figure 1: Table before adding the column

λ \ c_r	18	15	12	15	
0.0	0	1	0	1	$= 1$
0.0	1	0	0	1	$= 1$
15.0	0	1	0	0	$= 1$
18.0	1	0	0	0	$= 1$
0.0	1	0	0	0	$= 1$
12.0	0	0	1	0	$= 1$
0.0	0	0	1	0	$= 1$
0.0	0	0	1	0	$= 1$
0.0	1	1	1	1	≤ 3
θ	1.0	1.0	1.0	0.0	

Figure 2: Table after adding column

Figure 3: Notice that we use the **previous** table's λ's (dual variables, that represent the constraints) since the new column's reduced cost is relative to the **current** solution, the interesting cells have been marked in red

The remaining tables will be in subsection 7.1. Next I will introduce the following in respect to previous table in Figure 2.

$$r = [0, 6, 3, 4, 0]$$
$$c_r = 20$$
$$\hat{c} = -25.0$$
$$cost = 45.00$$
$$\lambda_{1,r} = (0.00, 0.00, 15.00, 18.00, 0.00, 12.00, 0.00, 0.00)$$
$$\lambda_0 = 0.0$$

And again.

$$r = [0, 5, 7, 1, 0]$$
$$c_r = 23$$
$$\hat{c} = -22.0$$
$$cost = 45.00$$
$$\lambda_{1,r} = (15.00, 0.00, 0.00, 0.00, 18.00, 0.00, 12.00, 0.00)$$
$$\lambda_0 = 0.0$$

Then.

$$r = [0, 4, 3, 8, 0]$$
$$c_r = 15$$
$$\hat{c} = -68.0$$
$$cost = 45.00$$
$$\lambda_{1,r} = (0.00, 0.00, 15.00, 18.00, 0.00, -13.00, 0.00, 25.00)$$
$$\lambda_0 = 0.0$$

And then all the following from iterating like above.

$$
\begin{aligned}
r_1 &= [0, 4, 1, 0] \\
c_{r_1} &= 14 \\
r_2 &= [0, 2, 4, 7, 0] \\
c_{r_2} &= 17 \\
r_3 &= [0, 5, 3, 0] \\
c_{r_3} &= 13 \\
r_4 &= [0, 5, 6, 8, 0] \\
c_{r_4} &= 13 \\
r_5 &= [0, 4, 1, 3, 0] \\
c_{r_5} &= 15 \\
r_6 &= [0, 5, 7, 8, 0] \\
c_{r_6} &= 13
\end{aligned}
$$

This solution is not optimal, as we can still find solutions that yield reduced cost, but for now these are the variables (θ) after adding the above column.

$$
\begin{aligned}
\theta_0 &= 0.50 \\
\theta_1 &= 0.00 \\
\theta_2 &= 1.00 \\
\theta_3 &= 0.50 \\
\theta_4 &= 0.00 \\
\theta_5 &= 0.00 \\
\theta_6 &= 0.00 \\
\theta_7 &= 0.00 \\
\theta_8 &= 0.00 \\
\theta_9 &= 0.50 \\
\theta_{10} &= 0.00 \\
\theta_{11} &= 0.50
\end{aligned}
$$

The first couple of tables can be found in the appendix. I stop at a non-integer solution, to solve task 6 and 7.

5 Task 5

I will first give a quick description of what the code does and then give a short example. First the `column_generation_RMP` function is invoked, which keeps iterating, since an internal return will stop the loop. In every iteration I solve the master problem, and find the duals. I then solve the pricing problem with these duals, which will yield an assignment of variables that represent a path $0 \to \cdots \to 0$. After finding this new path with the most negative reduced cost, the path with its cost is added to the original dataset and cost set, and the algorithm iterates. If the pricing problem finds only a positive reduced cost, then the current set (without the just found path with non-negative reduced cost), is the optimal set. Keep in mind much of the code is simply accounting

for things such as "off by 1" and loop base-cases for the implementations.

Since the lagrangian solution was very unfriendly, I will use the result of the LP relaxed solution of the initially supplied dataset. I will initial routes $\hat{\Omega} = \{0, 1, 4, 5, 7, 9, 10\}$ (where 0 is the customer 0 and not the depot). The found solution is.

θ	r
0.33	[0, 0, 0, 0, 1, 1, 0, 1]
0.33	[0, 0, 0, 0, 1, 0, 1, 1]
0.33	[0, 1, 1, 1, 0, 0, 0, 0]
0.33	[1, 1, 0, 1, 0, 0, 0, 0]
0.33	[0, 0, 0, 0, 0, 1, 1, 1]
0.33	[0, 0, 0, 0, 1, 1, 1, 0]
0.33	[1, 1, 1, 0, 0, 0, 0, 0]
0.33	[1, 0, 1, 1, 0, 0, 0, 0]

5.1 Acknowledgements

The solution is not completely correct, since I had a bug which I could not solve. The first couple of iterations are successful, but at some point I keep finding the same column to add every iteration. To get around this I terminate if a duplicate is found with reduced cost, but I acknowledge that this is not how the algorithm should be implemented.

6 Task 6

7 Appendix

7.1 CG Tables

λ \ c_r	18	15	12	15	20	
15.0	0	1	0	1	0	$= 1$
0.0	1	0	0	1	0	$= 1$
0.0	0	1	0	0	1	$= 1$
0.0	1	0	0	0	1	$= 1$
18.0	1	0	0	0	0	$= 1$
0.0	0	0	1	0	1	$= 1$
12.0	0	0	1	0	0	$= 1$
0.0	0	0	1	0	0	$= 1$
0.0	1	1	1	1	1	≤ 3
θ	1.0	1.0	1.0	0.0	0.0	

λ \ c_r	18	15	12	15	20	23	
0.0	0	1	0	1	0	1	$= 1$
0.0	1	0	0	1	0	0	$= 1$
15.0	0	1	0	0	1	0	$= 1$
18.0	1	0	0	0	1	0	$= 1$
0.0	1	0	0	0	0	1	$= 1$
-13.0	0	0	1	0	1	0	$= 1$
0.0	0	0	1	0	0	1	$= 1$
25.0	0	0	1	0	0	0	$= 1$
0.0	1	1	1	1	1	1	≤ 3
θ	1.0	1.0	1.0	0.0	0.0	0.0	

λ \ c_r	18	15	12	15	20	23	15	
15.0	0	1	0	1	0	1	0	$= 1$
0.0	1	0	0	1	0	0	0	$= 1$
0.0	0	1	0	0	1	0	1	$= 1$
13.0	1	0	0	0	1	0	1	$= 1$
5.0	1	0	0	0	0	1	0	$= 1$
7.0	0	0	1	0	1	0	0	$= 1$
3.0	0	0	1	0	0	1	0	$= 1$
2.0	0	0	1	0	0	0	1	$= 1$
0.0	1	1	1	1	1	1	1	≤ 3
θ	1.0	1.0	1.0	0.0	0.0	0.0	0.0	