Vehicle Routing Problem with Time Windows: Dantzig Wolfe Decomposition and Column Generation

Prasangsha Ganguly

January 11, 2021

Background

Let us first recall the Minkowski's Representation Theorem. Let, $\mathbf{X}$ be a polyhedron and let $X^1, ..., X^k$ be the extreme points of the polyhedron and $d^1, ..., d^l$ be the extreme directions of $\mathbf{X}$. Then, according to the representation theorem, any $x \in \mathbf{X}$ can be represented as a convex combination the extreme points and a conic combination of the extreme directions.

Dantzig Wolfe Decomposition

The idea of the Dantzig Wolfe is similar to Lagrange relaxation in a particular way. Consider the following problem: **LP:**

$$Z = Max\ C^T X \tag{1}$$
$$S.T.\ AX \leq b \tag{2}$$
$$DX \leq d,\ d \in \mathbb{Z}^m \tag{3}$$
$$X \geq 0 \tag{4}$$

In Lagrange Relaxation, we identified $AX \leq b$ were relatively easy to handle and what we did was dualize the difficult constraints $DX \leq d$ and put them in the objective function. In Dantzig Wolfe decomposition what we are going to do is to leave the difficult constraints into the problem but, we are going to use the representation theorem and use the extreme points and extreme directions of the polyhedron represented by the easy set $AX \leq b$. That is, say, $\chi = \{X : AX \leq b, X \geq 0\}$. Now, we are going to identify the extreme points (X^i) and extreme directions (d^j) of χ and use that to represent X of the original problem as

$$X = \sum_{i=1}^{k} \lambda_i X^i + \sum_{j=1}^{l} \mu_j d^j$$

For example consider a simple bounded problem without any extreme directions as shown in Fig 1. The green polyhedron is the easy part $\chi : \{X : AX \leq b\}$. We have found the extreme points of the region as X^1, X^2, X^3. The black polyhedron is the $Dx \leq d$ part. The extreme points of the

whole problem are marked as red. Note that as the whole problem is an intersection of the green polyhedron and the black polyhedron, any feasible point of the original problem can be represented as a convex combination of X^1, X^2, X^3. Essentially, any feasible point of the original problem is a point that is a convex combination of X^1, X^2, X^3 and satisfies the constraints $DX \leq d$.

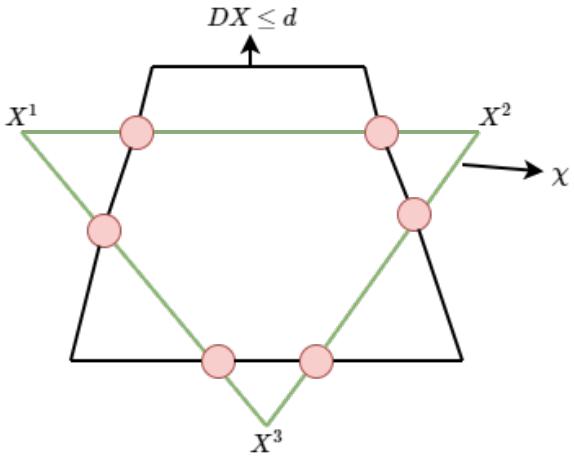

Figure 1: Representation Theorem

So, essentially, the **LP:** can be represented as,

$$Z = Max\ \lambda_1 C^T X^1 + \lambda_2 C^T X^2 + \lambda_3 C^T X^3 \tag{5}$$

$$S.T.\ \lambda_1 DX^1 + \lambda_2 DX^2 + \lambda_3 DX^3 \leq d \tag{6}$$

$$\lambda_1 + \lambda_2 + \lambda_3 = 1 \tag{7}$$

$$\lambda_1, \lambda_2, \lambda_3 \geq 0 \tag{8}$$

That is, we transformed the problem into the extreme points of χ. Note that, now our decision variables are λ_i. The extreme points are known.

In general, for a problem with extreme points and extreme directions, the **Dantzig Wolfe Reformulation** can be given as,

$$Z = Max\ \sum_{i=1}^{k} C^T X^i \lambda_i + \sum_{j=1}^{l} C^T d^j \mu_j \tag{9}$$

$$S.T.\ \sum_{i=1}^{k} DX^i \lambda_i + \sum_{j=1}^{l} Dd^j \mu_j \leq d \tag{10}$$

$$\sum_{i=1}^{k} \lambda_i = 1 \tag{11}$$

$$\lambda_i \geq 0\ \ \forall i = 1, ..., k \tag{12}$$

$$\mu_j \geq 0\ \ \forall j = 1, ..., l \tag{13}$$

Where,

$$X = \sum_{i=1}^{k} \lambda_i X^i + \sum_{j=1}^{l} \mu_j d^j$$

. Note that the decision variable for this reformulation are λ_i corresponding to each extreme point of the easy part of the problem and μ_j corresponding to each extreme direction. Essentially, we are convexifying the region which is easy corresponding to the constraint set $AX \leq b$. Actually, the extreme points for $AX \leq b$ are the solution of that part.

Motivation for the Column Generation

The main issue with this formulation is that the number of the extreme points and extreme directions for the part χ grows exponentially. Also, we need to know all the extreme points and directions of χ. Now, let us assume that the $DX \leq d$ has m_D number of constraints. Hence, the DW reformulation represented in (9 to 13) has $m_D + 1$ constraints where m_D constraints are for (10) and 1 constraint for (11).

As argued above, there are exponential number of extreme points and extreme directions of χ. But, as there are m_D constraints for the system $DX \leq q$, the basis is of the size m_D which is much less than the total number of extreme points and extreme directions. As the basis is of the size m_D, there can be only m_D variables at basis with non-zero values and a bunch of variables are non-basic.

Let us consider the dual problem of the DW reformulation. Say,
π_0 = The dual variable for constraint(11) and
π_k = The dual variables for the set of constraints(10) $\forall k = 1, ..., m_D$

$$Z_{dual} = Min \;\; d^T\pi + \pi_0 \tag{14}$$
$$S.T. \;\; \pi^T D X^i + \pi_0 \geq C^T X^i \;\; \forall i = 1, ..., k \tag{15}$$
$$\pi^T D d^j \geq C^T d^j \;\; \forall j = 1, ..., l \tag{16}$$
$$\pi \geq 0 \tag{17}$$

The constraint set (15) is associated with the extreme points X^i or the primal variables λ_i. The constraint set (16) is associated with the extreme directions d^j or the primal variables μ_j. This dual problem has very less number of variables $= m_D + 1$. However, there are exponential number of constraints associated with the exponential number of extreme points and extreme directions $(k + l)$ of the primal problem. According to the strong theorem of duality, $Z^*_{primal} = Z^*_{dual}$, which means in optimal solution the objective value of the primal problem is same as the objective value of the dual problem.

As the primal DW reformulation has exponential number of extreme points to be considered (correspondingly λ's), and in the basis there can be only a few of them, this generates the motivation of the column generation. The **idea** is, only add a few set set of variables λs and μs and later generate new variables if needed.

Column Generation

First, drop some variables λ and μ. That means some of the extreme points and extreme directions (solutions of $AX \leq b$) are not considered. We call this **Restricted Master Problem**. This is same as the primal DW formulation but only considering k' extreme points (or λ) and l' extreme directions (or μ) instead of all k extreme points and l extreme directions.

Restricted Master Problem (RMP)

$$Z_{RMP} = Max \sum_{i=1}^{k'} C^T X^i \lambda_i + \sum_{j=1}^{l'} C^T d^j \mu_j \tag{18}$$

$$S.T. \sum_{i=1}^{k'} DX^i \lambda_i + \sum_{j=1}^{l'} Dd^j \mu_j \leq d \tag{19}$$

$$\sum_{i=1}^{k'} \lambda_i = 1 \tag{20}$$

$$\lambda_i \geq 0 \ \ \forall i = 1, ..., k' \tag{21}$$

$$\mu_j \geq 0 \ \ \forall j = 1, ..., l' \tag{22}$$

Where, $k' \leq k$ and $l' \leq l$. The problem is called the restricted master problem because the solution space is smaller compared to the original primal problem (P). Any feasible solution to RMP is also feasible to the primal problem. Because, we can just take a solution in RMP and set other variables to 0 and that is a solution to the original problem. Hence restricted master problem is the problem with more constraints and hence, smaller solution space. So, $Z_P \geq Z_{RMP}$ as we have a maximization problem. If $Z_P = Z_{RMP}$, then we have the optimal solution.

Relaxed Dual Problem (RDP)

Now, let us consider the dual of the RMP. This is called the Relaxed Dual Problem. This problem is similar to the original dual problem but has less number of constraints as some variables from the primal problem are removed so corresponding constraints of the dual are also removed. Here we have k' constraints corresponding to selected λs and l' constraints corresponding to selected μs in the RMP.

$$Z_{RDP} = Min \ \ d^T \pi + \pi_0 \tag{23}$$

$$S.T. \ \ \pi^T DX^i + \pi_0 \geq C^T X^i \ \ \forall i = 1, ..., k' \tag{24}$$

$$\pi^T Dd^j \geq C^T d^j \ \ \forall j = 1, ..., l' \tag{25}$$

$$\pi \geq 0 \tag{26}$$

This is called relaxed dual problem because this problem has less number of constraints. So, the solution space is larger compared to the original dual problem. As we have a minimization problem so $Z_{RDP} \leq Z_{dual}$.

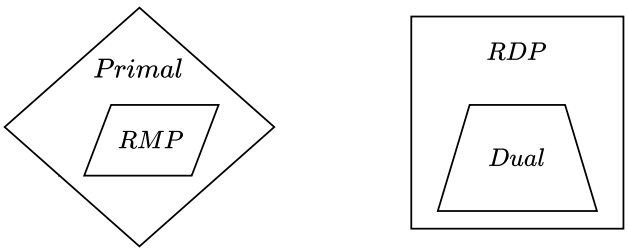

Figure 2: Relation of Primal-RMP and Dual-RDP

If $\pi^*(RDP)$ is the optimal solution of the RDP and if it is feasible for the original $Dual$ problem, then $\pi^*(RDP)$ is optimal to the the original problem.

Actually, we solve the Restricted Master Problem (RMP) and say, $\lambda^*(RMP)$ and $\mu^*(RMP)$ are the optimal solution for the RMP. Then immediately by complementary slackness, we obtain the optimal solution of the Relaxed Dual Problem (RDP), we don't have to solve RDP separately. Let, $\pi^*(RDP)$ and $\pi_0^*(RDP)$ be the optimal solution of the Relaxed dual problem.

Generating Columns

Hence our objective is to prove that $\pi^*(RDP)$ is feasible to the original $Dual$ problem and hence optimal to the original dual problem too. To check that a solution of the RDP is feasible to the original problem, we have to prove that the solution satisfies the constraints that are not in RDP but in the original problem. That is for those constraints in $k'+1...k$ and $l'+1...l$. As, RDP has less number of constraints than the original problem, so those constraints have to be checked which are removed from the original problem to create the RDP. In other words, $\pi^*(RDP)$ and $\pi_0^*(RDP)$ are optimal solution of the original dual problem if they satisfy:

$$\pi^*(RDP)^T DX^i + \pi_0^*(RDP) \geq C^T X^i \quad \forall i = k'+1, ..., k \tag{27}$$

$$\pi^*(RDP)^T Dd^j \geq C^T d^j \quad \forall j = l'+1, ..., l \tag{28}$$

Essentially, if we consider a bounded problem without any extreme directions, then we want to find if there is any extreme point X^i of $AX \leq b$ which violates the constraint (27). Or,

$$\pi^*(RDP)DX^i - \pi_0^*(RDP) < C^T X^i \tag{29}$$

or,

$$(C^T - \pi^*(RDP)D)X^i - \pi_0^*(RDP) > 0 \tag{30}$$

or,

$$(C^T - \pi^*(RDP)D)X^i > \pi_0^*(RDP) \tag{31}$$

However, we are not not going to check one by one all the extreme points X^i not included in RMP if that violates constraint (31) or not. Because as there are exponential number of such extreme points not in RMP, if we iterate over them, then we are not improving anything. Hence the idea is to **identify the maximum violation**. That is in constraint (31), find the maximum value of the left hand side. If that maximum value is less than $\pi_0^*(RDP)$, then no constraint is violated. Otherwise, there is atleast one violation and the corresponding extreme point is added to the master problem. Hence, this results in another optimization problem which is called the *Subproblem* or the slave which optimizes the violation and generates the new extreme point (column) to be added.

The Subproblem (slave)

So, the sub-problem is given by,

$$Max \ \ (C^T - \pi^*(RDP)D)X \tag{32}$$

$$S.T. \ \ AX \leq b \tag{33}$$

$$X \geq 0 \tag{34}$$

Note that the $\pi^*(RDP)$ is not a variable but the optimal solution of the RDP. The variable is X which actually generates new extreme points for $AX \leq b$ part.

Say, the optimal solution of the sub-problem is X^*, then we check if $(C^T - \pi^*(RDP)D)X^* > \pi_0$. Which means the corresponding dual constraint is violated. Hence the corresponding dual constraint or the primal variable (X^*) has to be added. Hence in the RMP, there will be a new $\lambda_{k'+1}$ associated with new found extreme point X^*.

We have only discussed for the extreme points. However, an extreme direction may also get violated i.e., the constraint (28) is violated. In this case, the sub-problem is,

$$Max \ \ (C^T - \pi^*(RDP)D)X \tag{35}$$

$$S.T. \ \ AX \leq b \tag{36}$$

$$X \geq 0 \tag{37}$$

Optimality condition of the sub-problem:

- If the subproblem has an optimal solution or the subproblem is bounded. Then there is no extreme direction, so no violation of the extreme directions too!
- If the optimal solution $(C^T - \pi^*(RDP)D)X^* \leq \pi_0$. Because if the maximum violation doesn't violate the constraint, then we are at optimal.

VRP with Time Window

Let us apply the Dantzig Wolfe decomposition and column generation to solve the vehicle routing problem with time windows. This problem is similar to the traditional vehicle routing problem with some changes. For each edge $(i, j) \in E$, there is an associated time of traversing the edge t_{ij}. Often we consider the cost of the edge is same as the time or $C_{ij} = t_{ij}$. For each node $i \in V$, there is a service time s_i which is the time required to serve the node. Also, there is a time window (l_i, u_i) associated with each node. This denotes that the node i has to be visited at earliest l_i and left latest by the time u_i. The decision variable is,

$$X_{ij}^r = \begin{cases} 1 & \text{if route (vehicle) } r \text{ uses the arc } (i,j) \\ 0 & \text{otherwise} \end{cases}$$

and

$$\kappa_{ir} = \text{The time route (vehicle) } r \text{ visits the node } i$$

Formulation

$$Min \sum_{r=1}^{R} \sum_{(i,j)\in E} C_{ij} X_{ij}^{r} \tag{38}$$

$$S.T. \sum_{r\in R} \sum_{j\in V} X_{ij}^{r} = 1 \ \ \forall i \in V \setminus \{0\} \tag{39}$$

$$\sum_{j\in V} X_{0j}^{r} = 1 \ \ \forall r \in R \tag{40}$$

$$\sum_{r\in R} \sum_{j\in V} X_{ij}^{r} - \sum_{r\in R} \sum_{j\in V} X_{ji}^{r} = 0 \ \ \forall i \in V \tag{41}$$

$$\sum_{i\in V} \delta_i \sum_{j\in V} X_{ij}^{r} \leq Capacity \ \ \forall r \in R \tag{42}$$

$$\kappa_{ir} + C_{ij} + s_i - M(1 - X_{ij}^{r}) \leq \kappa_{jr} \ \ \forall i \in V \setminus \{0\}, j \in V \setminus \{0\}, r \in R \tag{43}$$

$$\kappa_{ir} + C_{i0} + s_i - M(1 - X_{i0}^{r}) \leq u_0 \ \ \forall i \in V \setminus \{0\}, r \in R \tag{44}$$

$$l_i \sum_{j\in V} X_{ij}^{r} \leq \kappa_{ir} \leq u_i \sum_{j\in V} X_{ij}^{r} \ \ \forall i \in V, r \in R \tag{45}$$

$$X_{ij}^{r} \in \{0, 1\} \ \ \forall i \in V, j \in V, r \in R \tag{46}$$

In the formulation, the constraint in (39) denotes that for every vertex we must come out of every vertex i. This essentially means that every vertex other than the depot needs to be visited exactly once. The constraint (40) says that every route must start from the depot. The constraint (41) is the flow conservation constraint, which means that if we enter into a node, we must come out of it. The constraint (42) says that the sum of all the demands must be less than or equal to the capacity of the vehicle in that route. The subtour elimination constraints are given in (43) and (44). Note that, the constraints for eliminating subtours are expressed in terms of time of visiting a particular node. If node j is visited after i, then (43) has to be satisfied which says time of visit of node i + the time of travel + the service time of i will be less than time of visit of j. If a subtour exists then, there will be a pair of nodes i and j where i will be visited after j and j will be visited after i which violates (43). The constraint (45) denotes the time to visit a node must be bounded by the lower limit and the upper limit.

Dantzig Wolfe Decomposition

Note that, the solutions of the VRP-TW problem is a set of routes. For the DW reformulation, first we have to identify the constraint set $DX \leq d$ and $AX \leq b$. We consider, the constraint (39) ad our $DX \leq d$, which enforces that all the nodes are to be visited. The remaining constraints form the set $AX \leq b$ which actually generates new routes with all the characteristics of the routes like, flow conservation, capacity constraint, no subtour, time window restriction etc. Note that the solution of the $AX \leq b$ is a feasible route, but it may not visit all the nodes. Hence we create the following master and subproblem.

Master Problem

The master problem considers that the easy part $AX \leq b$ is already convexified and the extreme points are already obtained. Then, it enforces the constraint $DX \leq d$ onto it. For VRP-TW,

the extreme points are the feasible routes. The master problem checks if the routes visit all the customers or not. Hence, let us assume that R is the set of all the extreme points or feasible routes.

The decision variable,

$$Y_r = \begin{cases} 1 & \text{if route } r \in R \text{ is used} \\ 0 & \text{otherwise} \end{cases}$$

The parameters are,

$$C_r = \text{The cost of the route } r$$

and,

$$a_{ir} = \begin{cases} 1 & \text{if route } r \text{ visits customer } i \\ 0 & \text{otherwise} \end{cases}$$

We can now create the master formulation where we need to ensure that all the vertices are visited.

$$Min \ \sum_{r \in R} C_r Y_r \tag{47}$$

$$S.T. \ \sum_{r \in R} a_{ir} Y_r = 1 \ \ \forall i \in V \setminus \{0\} \tag{48}$$

$$\sum_{r \in R} Y_r \leq T \tag{49}$$

$$Y_r \in \{0, 1\} \ \ \forall r \in R \tag{50}$$

There are two issues in the formulation as described below.

- The above formulation has n customers, hence n constraints. However, as the number of extreme points or the feasible routes can increase exponentially, the number of decision variables Y_r can grow exponentially. We can use column generation to solve this issue as discussed before.
- Also, the decision variable Y_r is integer. However, till now we discussed about a real variable. So, to resolve this, we have to do Branch and Bound on top of column generation. This is called Branch and Price. However, we will discuss that later. For now, we consider the linear relaxation.

The Restricted Master Problem

In the restricted master problem, we consider only a few extreme points (feasible routes) $R' \subset R$. We define the linear relaxation.

$$Z_{RMP} = Min \ \sum_{r \in R'} C_r Y_r \tag{51}$$

$$S.T. \ \sum_{r \in R'} a_{ir} Y_r = 1 \ \ \forall i \in V \setminus \{0\} \tag{52}$$

$$\sum_{r \in R'} Y_r \leq T \tag{53}$$

$$Y_r \geq 0 \ \ \forall r \in R' \tag{54}$$

The Relaxed Dual Problem

The relaxed dual problem is the dual of the restricted master problem.

$$Z_{RDP} = Max \sum_{i \in V \setminus \{0\}} \pi_i - \pi_0 T \tag{55}$$

$$S.T. \sum_{i \in V \setminus \{0\}} a_{ir}\pi_i - \pi_0 \leq C_r \ \ \forall r \in R' \tag{56}$$

$$\pi_i \text{ are free } \ \forall i \in V \setminus \{0\} \tag{57}$$

$$\pi_0 \geq 0 \tag{58}$$

We want to find for the routes not in R' which violates the constraint (56). We rewrite the constraint (56) as,

$$C_r - \sum_{i \in V \setminus \{0\}} a_{ir}\pi_i \geq -\pi_0$$

The left hand side of the constraint is the reduced cost of the constraint (56). Hence we want to identify the violation of the constraint. Hence we want to check,

$$C_r - \sum_{i \in V \setminus \{0\}} a_{ir}\pi_i < -\pi_0 \tag{59}$$

To identify this, we minimize $C_r - \sum_{i \in V \setminus \{0\}} a_{ir}\pi_i$ and if that is positive then no constraint is violated. Otherwise, there is a potential route with a reduced cost and that has to be added.

The Subproblem

The subproblem essentially produces the new routes satisfying all the desired constraints of the routes like, flow conservation, capacity, time, no subtour etc. The objective of the subproblem is, to minimize $C_r - \sum_{i \in V \setminus \{0\}} a_{ir}\pi_i$ for the routes $r \in R \setminus R'$. However, there are two important points regarding this.

- Note that, the first term in the objective is C_r which is the cost of the route r. Though for the master problem, the route r is already available, for the subproblem, the route is not available. On the other hand, the subproblem essentially constructs the route. Hence for the subproblem, though we want to minimize $C_r - \sum_{i \in V \setminus \{0\}} a_{ir}\pi_i$, but it has to represented using edges used in a route not using the route r itself.

- The second term is $\sum_{i \in V \setminus \{0\}} a_{ir}\pi_i$. Here, a_{ir} is 1 if the route r visits the vertex i and 0 otherwise. Now, note that π_i is a free or unrestricted variable. Which means when π_i is $(-)ve$, then $C_r - \sum_{i \in V \setminus \{0\}} a_{ir}\pi_i$ is more than C_r and otherwise if π_i is $(+)ve$, then it is less than C_r. In a sense, π_i are some weights which tell how much we have to visit a vertex and not visit some vertex. As we are minimizing, if π_i is $(-)ve$ for some vertex i, then we will avoid visiting the vertex i.

Now, as mentioned in the first bullet point above, we have to convert the routes in terms of edges used for the subproblem. The subproblem decision variable is

$$\omega_{ij} = \begin{cases} 1 & \text{if we go from vertex } i \text{ to } j \\ 0 & \text{otherwise} \end{cases}$$

Using this decision variable, the cost the route C_r can be expressed as

$$C_r = \sum_{(i,j)\in E} C_{ij}\omega_{ij} \tag{60}$$

Furthermore, consider a_{ir}, which denotes if the route r visits the vertex i or not. This parameter also needs to be represented by ω_{ij}. Note that, if we visit a vertex i, then we also have to come out of that vertex. Hence,

$$a_{ir} = \sum_{\{j:(i,j)\in E\}} \omega_{ij} \tag{61}$$

Hence, by combining (60) and (61), we reformulated the objective of the subproblem as,

$$\sum_{\{j:(i,j)\in E\}} C_{ij}\omega_{ij} - \sum_{i\in V\setminus\{0\}} \pi_i \sum_{\{j:(i,j)\in E\}} \omega_{ij} \tag{62}$$

which is same as,

$$\sum_{\{(i,j)\in E\}} (C_{ij} - \pi_i)\omega_{ij} \tag{63}$$

By this way, we have converted the expression in terms of edges. We have to minimize it. The whole formulation of the subproblem is given as follows.

The decision variable κ_i denotes the time at which the node i is served.

$$Z_{sp} = Min \sum_{\{(i,j)\in E\}} (C_{ij} - \pi_i)\omega_{ij} \tag{64}$$

$$S.T. \sum_{j\in V} \omega_{0j} = 1 \tag{65}$$

$$\sum_{j\in V} \omega_{ij} - \sum_{j\in V} \omega_{ji} = 0 \ \ \forall i \in V \tag{66}$$

$$\sum_{i\in V} \delta_i \sum_{j\in V} \omega_{ij} \leq Capacity \tag{67}$$

$$\kappa_i + C_{ij} + s_i - M(1-\omega_{ij}) \leq \kappa_j \ \ \forall i \in V\setminus\{0\}, j \in V\setminus\{0\} \tag{68}$$

$$\kappa_i + C_{i0} + s_i - M(1-\omega_{ij}) \leq u_0 \ \ \forall i \in V \tag{69}$$

$$l_i \leq \kappa_i \leq u_i \ \ \forall i \in V \tag{70}$$

$$\omega_i \in \{0,1\} \tag{71}$$

Let us say Z^*_{sp} be the optimal solution of the subproblem. If it is less than 0, then a new route $Y_{r'}$ is created an added in the master problem. If Z^*_{sp} is non-negative then the optimal solution is reached.

Solving the $0/1$ problem

The problem with the last formulation is the master variable Y_r is a continuous variable rather than a binary variable. That is, we solved the linear relaxation not the integer program itself. If $Y_r = 0.6$ say, then it means that the route r is used 0.6 fractionally, which doesn't mean anything in reality. But, how to solve the integer problem?

Heuristic: Early Branching

One way is to switch all $Y_r \geq 0$ to 1 and solve the problem. However, this is a heuristic and there is no guarantee of optimality.

Branch and Price

The exact method with guaranteed optimal solution for the integer problem is to use Branch and Price. In this method, column generation and branch and bound is done. Essentially, for every node of the branch and bound tree, the column generation is done and the linear relaxation solution is obtained. Then from this node branching is done to obtain the integer solution. The Fig 3 depicts the underlying concept of the branch and price algorithm.

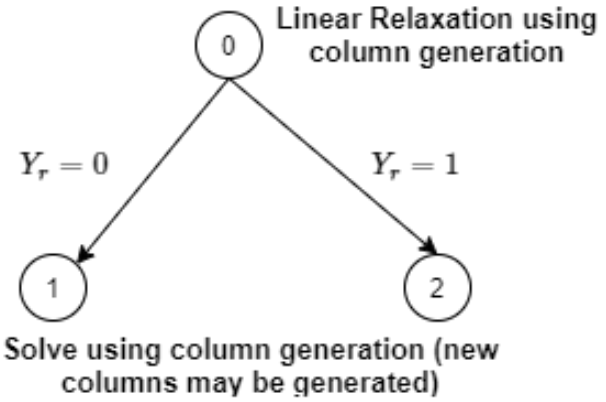

Figure 3: Branch and Price

However, for VRP, with a 0/1 branching there are two problems.

- Consider the branch $Y_r = 1$. When $Y_r = 1$, then the route r is selected and the vertices on the route r is already visited. So, the node will result in a VRP with fewer number of nodes. Which in turn fix a lot of other variables (Y_r) to 0. This is because for those customers that are on the route $Y_r = 1$ are already visited once and can't be visited again. So, all those other routes that could visit those particular customers are immediately forced to 0. So, making one of the variable to 1, a lot of other variables are forced to 0. On the other hand, when $Y_r = 0$, then it means one route is not selected. But, there are exponential number of other routes. So, the branch and bound tree becomes very unbalanced. When $Y_r = 1$, the problem size is reduced a lot by forcing a lot of other variables to 0, so there will be a few branches there. When $Y_r = 0$, the problem size is reduced by a very small portion, so, a lot of branching will be there. Hence we won't be able to generate the bounds efficiently.

- The second most important problem with the 0/1 branching is that we need to ensure that the subproblem doesn't produce the same routes again and again. Note that the master problem and the subproblem are disconnected. The subproblem has only the dual information of the master problem The dual information is through the π_i which indicates if a vetex is not visited already then a new route generated by the subproblem may try to visit the vertex. But, the subproblem doesn't know which routes and how many routes are there in the master problem. This is a issue. The subproblem only produces new better routes but ignorant about what routes are there in the master problem. So, in the branch $Y_r = 0$, the

subproblem may produce a duplicate route setting the same variable to 1. This is because when we are enforcing $Y_r = 0$, then we are adding more constraint, so the objective will deteriorate i.e. go up by a amount say , Δ. Now, the next time, the subproblem will try to minimize the objective and if it produces the same route, then the linear relaxation objective will again go back to the root node or reduced by Δ.
To overcome this, the master problem has to pass the information to the subproblem not to produce the same route as before. We have to generate some constraints which will enforce that the same route is not generated again. One such constraint can be to enforce that those edges that had not been used in the previous routes must be used atleast once in a new route. So, the two routes will differ. That is,

$$\sum_{\{(i,j)\in E:\ \omega_{ij}=0\}} \omega_{ij} \geq 1$$

Actually, this constraint can be improved. We can enforce that the new route either uses some edge not used before or if previously used edges are used then a fewer used edges are used.

$$\sum_{\{(i,j)\in E:\ \omega_{ij}=0\}} \omega_{ij} + (1 - \sum_{\{(i,j)\in E:\ \omega_{ij}=1\}} \omega_{ij}) \geq 1$$

There is another alternative. That is to use another type of branching rule which is called Ryan Foster Branching and is typically suitable for VRP.