CE 205A
Transportation Logistics

Lecture 17

Branch and Price for VRPs

Previously on Transportation Logistics

The two-index formulation keeps track of binary variables x_{ij} which is 1 if arc (i, j) is used and is 0 otherwise.

$$
\begin{aligned}
\min \quad & \sum_{(i,j)\in A} c_{ij}x_{ij} & \\
\text{s.t.} \quad & x(\delta^+(0)) = x(\delta^-(0)) = |K| & \\
& x(\delta^+(i)) = 1 & \forall\, i \in N\backslash\{0\} \\
& x(\delta^-(i)) = 1 & \forall\, i \in N\backslash\{0\} \\
& x(\delta^+(S)) \geq r(S) & \forall\, S \subseteq N\backslash\{0\}, S \neq \emptyset \\
& x_{ij} \in \{0, 1\} & \forall\, (i, j) \in A
\end{aligned}
$$

The fourth constraint, also called as the *capacity-cut constraints* (CCC) addresses both capacity limits as well as prevents sub-tours.

Previously on Transportation Logistics

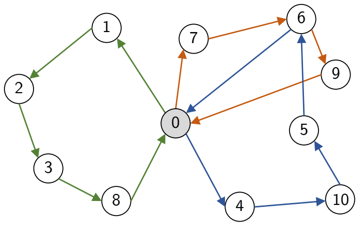
Suppose J is a set of tours which are feasible (satisfy capacity constraints). Let a_{ij} be 1 if tour j visits customer i and is 0 otherwise.

Define c_j as the cost of the tour and y_j as a binary variable which is 1 if tour j is chosen.

$$\begin{aligned}
\min \quad & \sum_{j \in J} c_j y_j \\
\text{s.t.} \quad & \sum_{j \in J} y_j = |K| \\
& \sum_{j \in J} a_{ij} y_j = 1 && \forall\, i \in V \backslash \{0\} \\
& y_j \in \{0, 1\} && \forall\, j \in J
\end{aligned}$$

Previously on Transportation Logistics

If the costs satisfy triangle inequality, the problem can be formulated as a covering problem.



$$\sum_{j \in J} a_{ij} x_j \geq 1 \ \forall \ i \in V \setminus \{0\}$$

The solution can be repaired by adding a short-cut (7,9) or (5,0).

The advantage of using the covering version is that the feasible tours can be replaced with *maximal-feasible tours*. For example, we need not have tours 0-4-0, 0-4-10-0, and 0-4-10-5-0, in the above example.

Further, the dual space is more constrained for the covering problem.

Previously on Transportation Logistics

Let decision variable w_{ik} indicate the start time of service at customer i by vehicle k.

$$
\begin{array}{lll}
\min & \displaystyle\sum_{k\in K}\sum_{(i,j)\in A} c_{ijk}x_{ijk} & \\
\text{s.t.} & \sum_{k\in K} x_k(\delta^+(i)) = 1 & i \in N\setminus\{0, n+1\} \\
& x_k(\delta^+(0)) = 1 & \forall\, k \in K \\
& x_k(\delta^+(i)) = x_k(\delta^-(i)) & \forall\, i \in N\setminus\{0, n+1\}, k \in K \\
& x_k(\delta^-(n+1)) = 1 & \forall\, k \in K \\
& w_{jk} \geq w_{ik} + s_i + t_{ij} - M_{ij}(1 - x_{ijk}) & \forall\, k \in K, (i,j) \in A \\
& a_i \leq w_{ik} \leq b_i & \forall\, k \in K, i \in N \\
& \sum_{i\in N} d_i x_k(\delta^+(i)) \leq C_k & \forall\, k \in K \\
& x_{ijk} \in \{0, 1\} & \forall\, (i,j) \in A, k \in K
\end{array}
$$

What is a good choice of M_{ij}? $b_i + s_i + t_{ij} - a_j$.

Previously on Transportation Logistics

The reduced costs are evaluated using the duals of the master problem. We begin with an initial set of columns J that guarantees feasibility of the restricted master.

1. Solve the restricted master problem

$$\begin{aligned} \min \; & \sum_{j \in J} c_j x_j \\ \text{s.t.} \; & \sum_{j \in J} \mathbf{A}_{.j} x_j = \mathbf{b} \\ & x_j \geq 0 \, \forall \, j \in J \end{aligned}$$

   Suppose $\mathbf{y}$ represents the optimal dual solution.

2. Solve the sub-problem $z_{sub} = \min_{j \in J^c}(c_j - \mathbf{y}^\mathsf{T}\mathbf{A}_{.j})$. Let j^* be the optimal solution.

3. If $z_{sub} < 0$, $J \leftarrow J \cup \{j^*\}$ and go to Step 1, else terminate.

Previously on Transportation Logistics

Define dual variables w_1 and w_2 for the first and second constraint of the LP relaxation of the restricted version of this master problem.

The first constraint can be written in standard form using a slack but that does not affect the pricing problem.

The reduced cost is therefore

$$\bar{c}_p = c_p - \begin{bmatrix} w_1 & w_2 \end{bmatrix} \begin{bmatrix} t_p \\ 1 \end{bmatrix} = c_p - w_1 t_p - w_2$$

To find a path which minimizes this expression, what are the decision variables? Can we write this as an optimization problem involving link flow variables?

Note that w_1 and w_2 are constants in the following pricing sub-problem.

Previously on Transportation Logistics

Note that the pricing problem does not have the complicating constraint. Hence, standard labeling methods can be used. The arc weights can be negative and hence we should ensure that the path is elementary.

$$z_{sub} = \min \sum_{(i,j) \in A} c_{ij} x_{ij} - w_1 t_{ij} x_{ij} - w_2$$

$$\text{s.t.} \sum_{j:(i,j) \in A} x_{ij} - \sum_{h:(h,i) \in A} x_{hi} = \begin{cases} 1 & \text{if } i = s \\ -1 & \text{if } i = t \\ 0 & \text{otherwise} \end{cases}$$

$$x_{ij} \in \{0, 1\} \, \forall \, (i, j) \in A$$

If $z_{sub} < 0$, then we add the new path formed by the links belonging to the path to the RMP and resolve it to get new dual variables.

Apply this method to find the optimal resource constrained path in the earlier instance.

Lecture Outline

Lecture Outline

VRP Reformulations

VRP Reformulations

Set-partitioning Version

Consider the set-partitioning version of the VRP. This model is generic and can handle capacity and time window constraints since we only pick those tours P which satisfies those constraints.

Suppose, we let some of the vehicles be unused, then we can write the formulation as

$$\begin{aligned} \min \quad & \sum_{p \in P} c_p y_p & \\ \text{s.t.} \quad & \sum_{p \in P} a_{ip} y_p = 1 & \forall\, i \in V \setminus \{0\} \\ & y_p \in \{0,1\} & \forall\, p \in P \end{aligned}$$

Write the pricing problem for this formulation for the CVRP and the CVRPTTW.

VRP Reformulations

Set-cover Version

Now consider a variant in which a customer can be visited more than once. Recall that if triangle inequality is satisfied, we can repair this solution.

Here, a_{ij} indicates the number of times route j visits customer i.

$$\begin{aligned} \min \quad & \sum_{p \in P} c_p y_p && \\ \text{s.t.} \quad & \sum_{p \in P} a_{ip} y_p \geq 1 && \forall\, i \in V \backslash \{0\} \\ & y_p \in \{0, 1\} && \forall\, p \in P \end{aligned}$$

Is the pricing problem different for this variant? Does it have a higher/lower LP relaxation objective compared to the previous partition version?

Lecture Outline

Elementary Shortest Paths with Resource Constraints

ESPPRC

Introduction

For solving LP relaxations of VRPs, it is necessary to find the elementary shortest paths which turns out to be NP-hard.

Also, the underlying graphs for this problem can have negative edge weights (why?), which needs to be factored while designing any algorithm.

ESPPRC

Introduction

Consider a graph $G = (N, A)$, an origin s and destination t. Each arc (i, j) has a cost c_{ij}. No non-negativity constraints on the costs are assumed.

Also assume L resources that are tracked by the traveler. Let $d^l_{ij} \geq 0$ be the consumption of resource l on arc (i, j). The resource vectors are assumed to follow triangle inequality.

Each node i has resource limits $[a^l_i, b^l_i]$ which are to be honored by every feasible path from s to i. These limits could represent time windows or capacities (say $[0, C]$).

Formulate the problem of finding the elementary path with the lowest cost while satisfying resource constraints as an integer program.

ESPPRC
MIP Formulation

Let binary decision variables x_{ij} indicate if arc (i,j) is chosen and let u_i^l be the consumption level of resource l at node i.

$$
\begin{aligned}
\min \quad & \sum_{(i,j)\in A} c_{ij}x_{ij} \\
\text{s.t.} \quad & x(\delta^+(i)) - x(\delta^-(i)) = \begin{cases} 1 & \text{if } i = s \\ -1 & \text{if } i = t \\ 0 & \text{otherwise} \end{cases} && \forall i \in N \\
& u_j^l \geq u_i^l + d_{ij}^l - M_{ij}(1 - x_{ij}) && \forall\, (i,j) \in A, l \in \{1,\dots,L\} \\
& a_i^l \leq u_i^l \leq b_i^l && \forall\, (i,j) \in A, l \in \{1,\dots,L\} \\
& x_{ij} \in \{0,1\} && \forall\, (i,j) \in A
\end{aligned}
$$

The formulation resembles a TSP with time windows and is closer to the prize collecting TSP since we need not visit all customers.

ESPPRC
Preliminaries

Instead of solving the above MIP using branch and bound, one can use labeling methods to find optimal solutions (as well as other elementary paths with negative reduced costs).

The algorithm we will discuss builds on the labeling method for the SPPRC (where cycles are allowed) by Desrochers et al. (1992).

In this method, for each node i, we maintain a list of labels, each of which corresponds to a path from s to i and the corresponding resource utilization vectors.

ESPPRC

Preliminaries

For a given node i, downstream arcs (i, j) are relaxed by updating their labels by extending the labels of node i.

For a path p from s to i, we maintain labels of the form (R_i^p, C_i^p), where $R_i^p = (U_i^{p1}, \ldots, U_i^{pL})$ is a list of resources used and C_i^p is the cost of the path.

What is the problem with this method? This can lead to an exponential number of labels as the number of paths grows exponentially.

ESPPRC

Dominance

To address this issue, dominance rules are applied to prune labels (and paths) that violate resource constraints or might be sub-optimal.

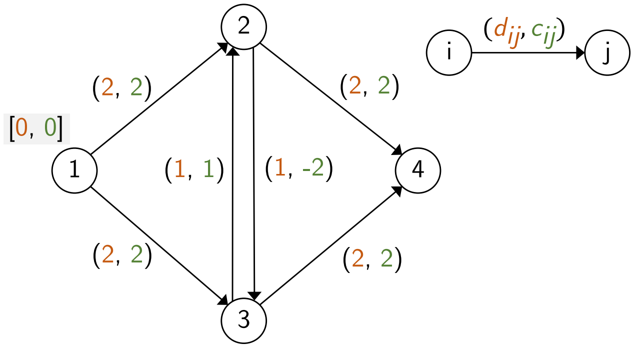
Given two distinct paths p and q to node i, we say that p dominates q iff $U_i^{pl} \leq U_i^{ql}$ $\forall l \in L$ and $C_i^p \leq C_i^q$, and $(R_i^p, C_i^p) \neq (R_i^p, C_i^p)$.

At any stage if any of the resource constraints are violated, we do not extend the label to the successor nodes.

ESPPRC

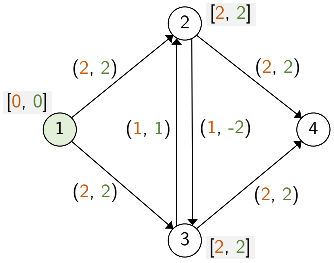
SPPRC – Labeling Algorithm

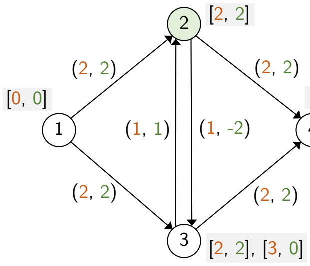
Consider the following network with a single resource. Starting with node 1, update the labels of the successors.



ESPPRC

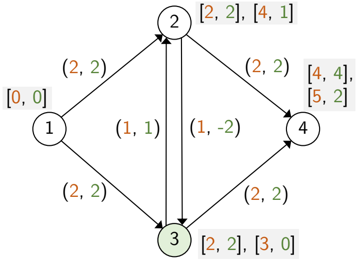
SPPRC – Labeling Algorithm

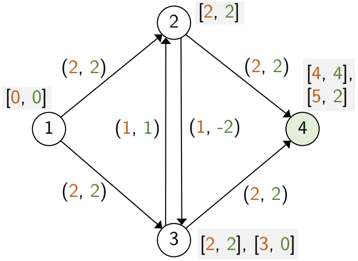




ESPPRC

SPPRC – Labeling Algorithm



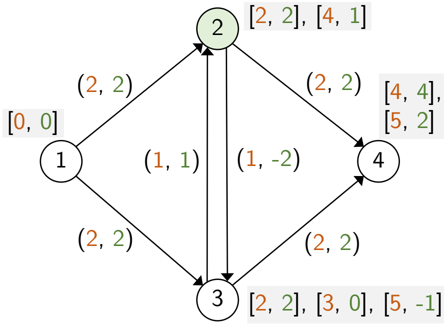


Do we terminate?

ESPPRC

SPPRC – Labeling Algorithm

Let us relax the edges going out of node 2 again. Why don't we add a label [5, 3] to node 4?



The algorithm will converge if we have resource constraints at the nodes and might cycle a few times between nodes 2 and 3, else, it might be unbounded.

ESPPRC
Labeling Algorithm

To extend this method to the case where we do not allow cycles between nodes, we add more labels representing a unit *visitation* resource associated with each node V_i^{pj}.

In addition, we also keep track of the nodes visited by a path p that connects s to i using m_i^p, which helps us check the dominance rules faster.

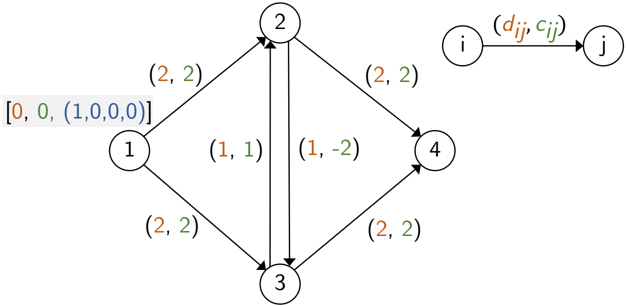
Given two distinct paths p and q from s to i, with labels (R_i^p, C_i^p) and (R_i^q, C_i^q). We say p dominates q iff $C_i^p \leq C_i^q$, $m_i^p \leq m_i^q$, $U_i^{pl} \leq U_i^{ql}\ \forall\ l \in L$, $V_i^{pj} \leq V_i^{qj}\ \forall\ j = 1, \ldots, n$ and $(R_i^p, C_i^p) \neq (R_i^q, C_i^q)$.

This is a weaker version and can be further strengthened to limit the number of labels generated.

ESPPRC

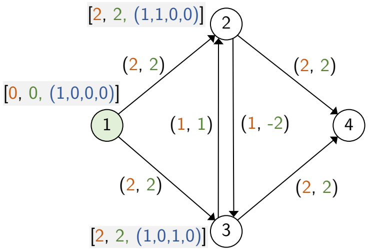
Labeling Algorithm

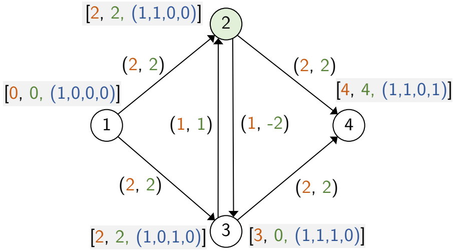
Let us initialize the labels as shown below. The resource limits are not explicitly added in this example. One could prune a few labels if they were present.



ESPPRC

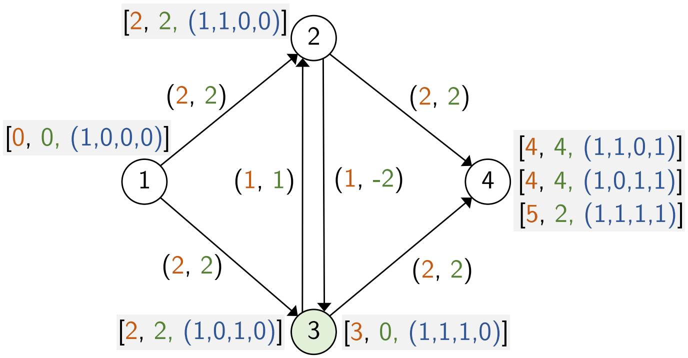
Labeling Algorithm





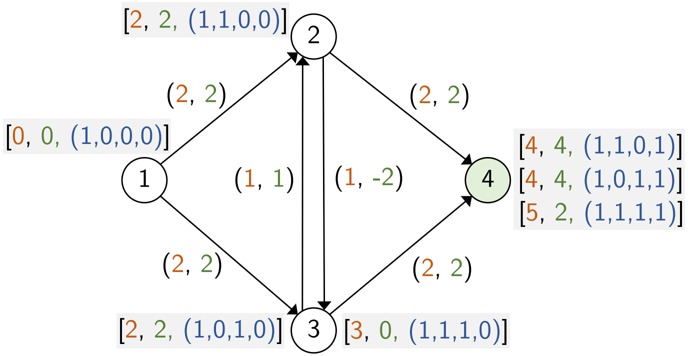
ESPPRC

Labeling Algorithm



ESPPRC

Labeling Algorithm



ESPPRC

Labeling Algorithm

Algorithm 1: ESPPRC(s)

```
Step 1: Initialization
$\Lambda_s \leftarrow ((0, \ldots, 0), 0, (0, \ldots, 0)\}$ ;
$\Lambda_i \leftarrow \emptyset \ \forall \ i \in N \setminus \{s\}$ ;
$SEL \leftarrow \{s\}$

Step 2: Extend labels
while $SEL \neq \emptyset$ do
    Remove an element $i$ from SEL ;
    for $j \in \delta^+(i)$ do
        $F_{ij} \leftarrow \emptyset$;
        for $\lambda_i \in \Lambda_i$ do
            if $j$ has not been visited then
                $F_{ij} \leftarrow F_{ij} \cup \{\textsc{Extend}(\lambda_i, j)\}$;
                $\Lambda_j \leftarrow \textsc{RemoveNonDominatedLabels}(F_{ij} \cup \Lambda_j)$ ;
            end
        end
    end
    if $\Lambda_j$ was updated then
        Add $j$ to SEL if it is not already present
    end
end
```

ESPPRC
Labeling Algorithm

This ESPPRC algorithm was proposed by Feillet et al. (2004) along with additional pruning steps that can be applied while extending the labels.

More sophisticated methods involving bi-directional search have been proposed which are faster. The `cspy` library has a repository of these algorithms.

The VRP problem involves tours that start at a depot and end at a depot. So, can we apply these algorithms directly?

These methods only help solve the LP relaxations of the VRP problem. For finding integer solutions, we would still have to branch on the fractional variables.

ESPPRC

Branch and Price

This method can be combined with BnB where at each BnB tree node, we use column-generation to solve the set partitioning version of the VRP.

Several options exist for deciding how to branch. Empirically, branching on the link variables x_{ij} is better than path/tour variables y_p.

We can also generate these paths at the root node and let the solvers use them to find integer solutions (Price and Branch). This method is easy to implement but can lead to sub-optimal solutions.

Several families of TSP-based valid inequalities are known to be facet-defining for VRPs. These can also be added to design branch, cut, and price algorithms.

Your Moment of Zen

ON THE ROAD TO SUCCESS, THERE ARE NO SHORTCUTS.
JOIN OUR TEAM
800-669-0322
shaffertrucking.com
SHAFFER TRUCKING
OUR MOST VALUABLE RESOURCE SITS 63 FEET AHEAD.
C 240