Shortest path problem from the perspective of duality

Reporter: Maocan Song
Date: 8/10/2020

1 Problem statement

Problem statement	Find a least-cost path from origin node to destination node. (**Single commodity** problem, its **quantity** equals 1)
Input	• A network with nodes and links • Link costs an node costs • Origin and destination
Output	The shortest path from origin to destination
Objective function	$\min \sum_{(i,j)\in A} c_{ij} x_{ij}$
Constraints	$\sum_{i:(i,j)\in A} x_{ij} - \sum_{i:(j,i)\in A} x_{ji} = \begin{cases} 1 & i = O \\ -1 & i = D \\ 0 & o.w. \end{cases}$ $\boldsymbol{x_{ij} \geq 0}, \forall ij \in A$
Problem type	Linear program

2. Lagrangian relaxed problem

$$L(\mu) = min \sum_{(i,j)\in A} c_{ij}x_{ij} + \mu_O\left(\sum_{O:(O,j)\in A} x_{Oj} - 1\right) + \sum_{i\in E/\{O,D\}} \mu_i\Big(\sum_{i:(i,j)\in A} x_{ij} - \sum_{i:(j,i)\in A} x_{ji}\Big) + \mu_D\left(-\sum_{D:(D,j)\in A} x_{Dj} + 1\right)$$

$$= (\mu_D - \mu_O) + \min \sum_{(i,j)\in A} c_{ij}x_{ij} + \mu_O \sum_{O:(O,j)\in A} x_{Oj} - \mu_D \sum_{D:(D,j)\in A} x_{Dj} + \sum_{i\in E/\{O,D\}} \mu_i\Big(\sum_{i:(i,j)\in A} x_{ij} - \sum_{i:(j,i)\in A} x_{ji}\Big)$$

$$L_2(\mu) = \min \sum_{(i,j)\in A} c_{ij}x_{ij} + \mu_O \sum_{O:(O,j)\in A} x_{Oj} - \mu_D \sum_{D:(D,j)\in A} x_{Dj} + \sum_{i\in E/\{O,D\}} \mu_i\Big(\sum_{i:(i,j)\in A} x_{ij} - \sum_{i:(j,i)\in A} x_{ji}\Big)$$

$$L_{2-ij}(\mu) = \min c_{ij}x_{ij} + \mu_i x_{ij} - \mu_j x_{ij} = \min(c_{ij} + \mu_i - \mu_j)\, x_{ij}$$

$$L(\mu) = L_1(\mu) + L_2(\mu) = L_1(\mu) + \sum_{(i,j)\in A} \min(c_{ij} + \mu_i - \mu_j)\, x_{ij}, x_{ij} \geq 0$$

Case 1: $c_{ij} + \mu_i - \mu_j < 0\ \exists ij \in A$, $L(\mu) = -\infty$ 无界

Case 2: $c_{ij} + \mu_i - \mu_j \geq 0\ \forall ij \in A$, $L_2(\mu) = 0$

最短路的对偶问题如下：

$$\max \mu_D - \mu_O$$
$$c_{ij} + \mu_i - \mu_j \geq 0, \forall ij \in A$$

The relationship with label-correcting algorithm

5.2 OPTIMALITY CONDITIONS

As noted previously, label-correcting algorithms maintain a distance label $d(j)$ for every node $j \in N$. At intermediate stages of computation, the distance label $d(j)$ is an estimate of (an upper bound on) the shortest path distance from the source node s to node j, and at termination it is the shortest path distance. In this section we develop necessary and sufficient conditions for a set of distance labels to represent shortest path distances. Let $d(j)$ for $j \neq s$ denote the length of a shortest path from the source node to the node j [we set $d(s) = 0$]. If the distance labels are shortest path distances, they must satisfy the following necessary optimality conditions:

$$d(j) \leq d(i) + c_{ij}, \qquad \text{for all } (i, j) \in A. \tag{5.1}$$

These inequalities state that for every arc (i, j) in the network, the length of the shortest path to node j is no greater than the length of the shortest path to node i plus the length of the arc (i, j). For, if not, some arc $(i, j) \in A$ must satisfy the condition $d(j) > d(i) + c_{ij}$; in this case, we could improve the length of the shortest path to node j by passing through node i, thereby contradicting the optimality of distance labels $d(j)$.

These conditions also are sufficient for optimality, in the sense that if each $d(j)$ represents the length of some directed path from the source node to node j and this solution satisfies the conditions (5.1), then it must be optimal. To establish this result, consider any solution $d(j)$ satisfying (5.1). Let $s = i_1 - i_2 - \ldots - i_k = j$ be any directed path P from the source to node j. The conditions (5.1) imply that

$$d(j) = d(i_k) \leq d(i_{k-1}) + c_{i_{k-1}i_k},$$

$$d(i_{k-1}) \leq d(i_{k-2}) + c_{i_{k-2}i_{k-1}},$$

An illustrative example in GAMS

```
*model
variable z;
positive variable x(i,j);

equations
obj
flow_balance_origin(i)
flow_balance_destination(i)
flow_balance_otherwise(i)
;
obj..z =e= sum(i,sum(j$cost(i,j),cost(i,j)*x(i,j)));
flow_balance_origin(i)$origin_node(i).. sum(j$cost(i,j),x(i,j)) =e= 1;
flow_balance_destination(i)$destination_node(i)..sum(j$cost(j,i),x(j,i)) =e= 1;
flow_balance_otherwise(i)$(intermediate_node(i)).. sum(j$cost(i,j),x(i,j))-sum(j$cost(j,i),x(j,i)) =e= 0;

Model SPP /all/;
Solve SPP using LP minimizing z;
```

```
variable z;
variables y(i);

equations
obj
optimal_cost(i,j)
;
obj.. z =e= y("7")-y("1");
optimal_cost(i,j)$cost(i,j).. y(j)-y(i) =l= cost(i,j);

model SPP /all/;
solve SPP using LP maximizing z;
```

```
Tried aggregator 1 time.
LP Presolve eliminated 8 rows and 10 columns.
All rows and columns eliminated.
Presolve time =    0.00 sec.
LP status(1): optimal

Optimal solution found.
Objective :          47.000000
```

```
All rows and columns eliminated.
Presolve time =    0.00 sec.
LP status(1): optimal

Optimal solution found.
Objective :          47.000000
```

3. TSP and the dual problem

Primal problem	
Objective function	$\min \sum_{(i,j)\in A} c_{ij}x_{ij}$
Constraints	$\sum_{i:(i,j)\in A} x_{ij} - \sum_{i:(j,i)\in A} x_{ji} = \begin{cases} 1 & i = O \\ -1 & i = D \\ 0 & o.w. \end{cases}$ $\sum_{i:(i,j)\in A} x_{ij} = 1, i \in A_{task}$ $\boldsymbol{x_{ij} \geq 0}, \forall ij \in A$
Dual problem	
Objective function	$\max \mu_D - \mu_O - \sum_{i\in A_{task}} w_i$
Constraints	$\begin{cases} c_{ij} + \mu_i - \mu_j + w_i \geq 0, & i \in A_{task} \\ c_{ij} + \mu_i - \mu_j \geq 0, & o.w. \end{cases}, \forall ij \in A$

Thanks

- TSP-dual problem
- Space-time network
- In State-space-time network?