Graphs and Network Flows
IE411

Lecture 17

Dr. Ted Ralphs

Reduced Cost Optimality Conditions

- Let $\pi(i)$ be a *dual variable* associated with node $i \in N$ (the *potential* of node i)
- For a given set of node potentials π, the *reduced cost* of arc (i, j) is $c_{ij}^{\pi} = c_{ij} - \pi(i) + \pi(j)$.

Property 1. [9.2]

(a) *For any directed path P from node k to node l,* $\sum_{(i,j)\in P} c_{ij}^{\pi} = \sum_{(i,j)\in P} c_{ij} - \pi(k) + \pi(l)$.

(b) *For any directed cycle W,* $\sum_{(i,j)\in W} c_{ij}^{\pi} = \sum_{(i,j)\in W} c_{ij}$.

Reduced Cost Optimality Conditions

Theorem 1. [9.3] *A feasible solution* x^* *is an optimal solution of the minimum cost flow problem if and only if some set of node potentials* π *satisfy* $c_{ij}^{\pi} \geq 0 \; \forall (i, j) \in G(x^*)$.

Successive Shortest Paths Algorithm

- The cycle-canceling algorithm maintained feasibility in each iteration and tried to achieve optimality.
- We will now consider two algorithms that maintain optimality conditions and try to achieve feasibility.
- Recall that a *pseudoflow* is a flow satisfying nonnegativity and capacity constraints, but not necessarily flow balance.
- We will start with a pseudoflow and then try to make it into a flow by sending flow along shortest paths.

Successive Shortest Paths Algorithm

Lemma 1. [9.11] *Suppose that a pseudoflow* x *satisfies the reduced cost optimality conditions with respect to some node potentials* π. *Let* $\vec{d}$ *represent the shortest path distances from some node* s *to all other nodes in the residual network* $G(x)$ *with* c_{ij}^{π} *as the length of an arc* (i, j). *Then:*

(a) *The pseudoflow* x *also satisfies the reduced cost optimality conditions with respect to the node potentials* $\pi' = \pi - d$.

(b) *The reduced costs* $c_{ij}^{\pi'}$ *are zero for all arcs* (i, j) *in a shortest path from* s *to every other node.*

Proof:

Successive Shortest Paths Algorithm

Lemma 2. [9.12] *Suppose that a pseudoflow x satisfies the reduced cost optimality conditions and we obtain x' from x by sending flow along a shortest path from s to some other node k. Then x' also satisfies the reduced cost optimality conditions.*

Proof:

Successive Shortest Paths Algorithm

- The previous properties establishes the validity of the following basic algorithm.
- Start with a pseudo flow $x := 0$ and node potentials $\pi := 0$ and perform the following loop until feasibility is achieved.
  - Identify a node s with an excess and a node t with a deficit.
  - Determine shortest path distances d from s to all other nodes in the residual network with respect to the reduced costs c_{ij}^{π}.
  - Send flow along a shortest path from s to t.
  - Update $\pi \to \pi - d$.
- From the previous slide, we know that sending flow along a shortest path and updating node potentials will maintain optimality conditions.
- The algorithm strictly decreases the excess at some node in each iteration.
- The number of iterations is at most nU, where U is the largest supply.

Complementary Slackness Optimality Conditions

Theorem 2. [9.4] *A feasible solution x^* is an optimal solution of the MCFP if and only if for some set of node potentials π, the reduced costs and the flow values satisfy, for every $(i, j) \in A$,*

(a) *If $c_{ij}^{\pi} > 0$, then $x_{ij}^* = 0$.*

(b) *If $0 < x_{ij}^* < u_{ij}$, then $c_{ij}^{\pi} = 0$.*

(c) *If $c_{ij}^{\pi} < 0$, then $x_{ij}^* = u_{ij}$.*

Proof:

1. If node potentials π and flow vector x satisfy the reduced cost optimality conditions, then they must satisfy (a) - (c).

2. If flow vector x and node potentials π satisfy the complementary slackness conditions, then they satisfy the reduced cost optimality conditions.

Linear Programming Duality

Maximize $\sum_{j=1}^{n} c_j x_j$	Minimize $\sum_{i=1}^{m} \pi_i b_i$
$\sum_{j=1}^{n} a_{ij} x_j \leq b_i$	$\pi_i \geq 0$
$\sum_{j=1}^{n} a_{ij} x_j = b_i$	π_i free
$\sum_{j=1}^{n} a_{ij} x_j \geq b_i$	$\pi_i \leq 0$
$x_j \geq 0$	$\sum_{i=1}^{m} \pi_i a_{ij} \geq c_j$
x_j free	$\sum_{i=1}^{m} \pi_i a_{ij} = c_j$
$x_j \leq 0$	$\sum_{i=1}^{m} \pi_i a_{ij} \leq c_j$

Minimum Cost Flow Duality

$$\text{Maximize} \quad w(\pi, \alpha) = \sum_{i \in N} b_i \pi_i - \sum_{(i,j) \in A} u_{ij} \alpha_{ij} \quad (1)$$

$$\text{subject to} \quad \pi_i - \pi_j - \alpha_{ij} \leq c_{ij} \quad \forall (i,j) \in A \quad (2)$$

$$\alpha_{ij} \geq 0 \quad \forall (i,j) \in A \quad (3)$$

$$\pi_i \quad \text{free} \quad \forall i \in N \quad (4)$$

Weak Duality

Theorem 3. [9.5] *Let* $z(x)$ *denote the objective function value of some feasible solution* x *of the MCFP and let* $w(\pi, \alpha)$ *denote the objective function value of some feasible solution* (π, α) *of its dual. Then,* $w(\pi, \alpha) \leq z(x)$.

Strong Duality

Theorem 4. [9.6] *For any choice of problem data, the MCFP always has a solution* x^* *and the dual MCFP has a solution* π *satisfying* $z(x^*) = w(\pi)$.

Theorem 5. [9.7] *If* x *is a feasible flow and* π *is an (arbitrary) vector satisfying* $z(x) = w(\pi)$*, then* (x, π) *satisfy the complementary slackness optimality conditions.*

Consequence of Duality

Property 2. [9.8] *If* x^* *is an optimal solution of the MCFP and* π *is an optimal solution of the dual MCFP, then* (x^*, π) *satisfy the complementary slackness optimality conditions.*

Proof:

Theorem 9.9

Theorem 6. [9.9] *Any linear program that contains (a) at most one +1 and at most one -1 in each column or (b) at most one +1 and at most one -1 in each row can be transformed into a MCFP.*

Implications of Minimum Cost Flow Duality

- Almost all algorithms for solving the primal problem also generate optimal node potentials
- Most algorithms explicitly or implicitly use properties of dual variables
- The primal problem and the dual problem are related closely via basic shortest path problem and maximum flow problem

Computing Optimal Node Potentials

Given an optimal flow x^*, we can obtain optimal node potentials by solving a shortest path problem (with possibly negative arc lengths).

1. What do we know about $G(x^*)$?
2. What does that mean about the shortest path distances $d(\cdot)$ in $G(x^*)$?
3. How can we use the shortest path distances to get node potentials?

Computing Optimal Flows

Given optimal node potentials π, we can obtain an optimal flow x^* by solving a maximum flow problem.

1. What arcs should be in the maximum flow problem?
2. How do we transform the network into a maximum flow problem?

Primal-Dual Algorithm

- Maintains a pseudoflow that satisfies reduced cost optimality conditions.
- Gradually converts pseudoflow to flow by augmenting along shortest paths.
- In contrast to the successive shortest path algorithm, we send flow along multiple shortest paths simultaneously.
- This is done by solving a maximum flow problem.

Primal-Dual Algorithm

Transform minimum cost flow problem into a problem with a single excess node and a single deficit node

- Introduce a source node s and a sink node t
- For each node i with $b(i) > 0$, add a zero cost arc (s, i) with capacity $b(i)$
- For each node i with $b(i) < 0$, add a zero cost arc (i, t) with capacity $-b(i)$
- Set $b(s) = \sum_{i \in N : b(i) > 0} b(i)$, $b(t) = -b(s)$ and $b(i) = 0 \ \forall i \in N$

Primal-Dual Algorithm

Admissible Network $(G^\circ(x))$

- subgraph of residual network $G(x)$ defined with respect to a pseudoflow x that satisfies the reduced cost optimality conditions for some node potentials π
- contains only those arcs in $G(x)$ with zero reduced cost
- residual capacity of arc in $G^\circ(x)$ same as in $G(x)$

Primal-Dual Algorithm

Input: A network $G = (N, A)$, a vector of capacities $u \in \mathbb{Z}^A$, a vector of costs $c \in \mathbb{Z}^A$, and a vector of supplies $b \in \mathbb{Z}^N$

Output: x represents a minimum cost network flow

$x \leftarrow 0$ and $\pi \leftarrow 0$

$e(s) \leftarrow b(s)$ and $e(t) \leftarrow b(t)$

while $e(s) > 0$ **do**

determine $d(\cdot)$ from node s to all other nodes in $G(x)$ with respect to c^π_{ij}

update $\pi := \pi - d$

define the admissible network $G^\circ(x)$

establish a maximum flow from node s to node t in $G^\circ(x)$

update $e(s)$, $e(t)$ and $G(x)$.

end while

Complexity of the Primal-Dual Algorithm

- In each iteration
  - The excess of node s is decreased.
  - The node potential of t is decreased.
- The total number of iterations is $\min nC, nU$.
- In each iteration, we have to solve a max flow problem and a shortest path problem.