Lecture 4: Max flow problem using linear programming

Rajat Mittal

IIT Kanpur

The focus of this lecture note is to learn primal dual methods to solve linear programming problems. To show the approach, we will take the example of the *max-flow problem*, it is closely related to another combinatorial problem called *min-cut*.

In the beginning, we will see the definition of these problems and their formulation as a linear program. Later, we will introduce the idea of primal dual method. Finally, we will solve the max-flow/min-cut problem using primal dual method.

1 Network flow problems

Problems related to flows in networks have been studied widely in optimization and algorithms literature. Few examples of these problems are: matching, s-t shortest path, maximum flow and minimum cost problems. Our focus will be on the last two problems, maximum flow and minimum cut.

1.1 Maximum flow problem

Look at a water distribution network in any country, say India. A water distribution network consists of water pipes of different capacities connecting different nodes. Here, nodes can be source of water, cities needing water or just some transfer nodes. In general, the problem is to transfer maximum amount of water to different cities using this network.

For the sake of simplicity, let us assume that there is only one source (of water) and only one sink (a city requiring water). The goal is to find the maximum amount of water which can be sent from source to sink given the capacities of pipes.

The water sent from source to sink can be represented by amount of water sent between every pair of nodes (flow between the pair). This *flow* should have the following properties.

1. The amount of flow in every pipe is less than its capacity (capacity constraints).
2. At every node, the incoming flow is same as the outgoing flow, except source and sink (flow conservation constraints).
3. The outgoing flow at the source is same as the incoming flow at the sink (same as the amount of water transferred). This is the quantity which needs to be maximized.

The problem can be formulated in terms of a graph, look at Fig. 1. We are given a directed graph $G(V, E)$ with:

- A node called source (s),
- A node called sink (t),
- Capacities $c_{u,v}$ for every edge $(u, v) \in E$ (edge represents a connection by a pipe).

The problem is to find the maximum flow possible in the graph G from s to t. A flow is a positive value $f_{u,v}$ for every edge $(u, v) \in E$.

Exercise 1. Convince yourself that the graph problem is same as the simple water distribution problem. What are the constraints on the flow?

Exercise 2. Can you think of an algorithm for this problem? Don't worry about the time taken by the algorithm. What if you are give the guarantee that the flow will only take integer values?

As you might have suspected, this problem can be converted into a linear program. Suppose, $f_{u,v}$ denotes the flow from vertex u to v. Taking into consideration the conditions for a valid flow, the linear program looks like:

$$\begin{aligned} & \max \sum_{\{s,u\}} f_{s,u} \\ \text{s.t.} \sum_{u:\{u,v\}\in E} f_{u,v} = & \sum_{u:\{v,u\}\in E} f_{v,u} \qquad \forall v \neq s,t \quad \text{(flow conservation)} \\ f_{u,v} \leq c_{u,v} & \quad \forall (u,v) \in E \quad \text{(capacity constraint)} \\ f_{u,v} \geq 0 & \quad \forall (u,v) \in E \end{aligned}$$

Note 1. There is also a condition that the outgoing flow at source is same as incoming flow at the sink. But this condition is redundant given other flow conservation constraints.

Exercise 3. Convince yourself that the linear program above captures the max flow problem. Can you think of an algorithm now to solve max flow?

The flow conservation is present on every vertex except at source and the sink. This can be made uniform by putting an edge from t to s and it can be given ∞ capacity. If that does not make you feel comfortable, any upper bound on maximum flow will work.

Exercise 4. Give an upper bound on maximum flow using capacities.

Then the linear program becomes,

$$\begin{aligned} & \max f_{t,s} \\ \text{s.t.} \sum_{u:\{u,v\}\in E} f_{u,v} \leq & \sum_{u:\{v,u\}\in E} f_{v,u} \quad \text{(flow conservation)} \\ f_{u,v} \leq c_{u,v} & \quad \text{(capacity constraint)} \\ f_{u,v} \geq 0 & \end{aligned} \tag{1}$$

Notice that we have exchanged equality in flow conservation with inequality. You will show in the assignment that this change does not affect the LP.

1.2 Minimum cut problem

Let's look at the problem from a bad guy's perspective. Suppose you need to cut the water supply from the source to the sink. The amount of time taken to cut a pipe is proportional to its capacity. How can you minimize your effort in cutting off the water supply?

Say, the set of nodes still connected to s (after cutting off some pipes) is S. If the water supply is cut then there is no edge going from a vertex in S to a vertex in $\bar{S}$. Here, $\bar{S}$ denotes the remaining vertices.

In graph theoretic framework, this is called a cut between s and t (any subset S containing s and not containing t). A cut can also be represented as $S, \bar{S}$. The size/capacity of the cut is the total capacity of edges going from S to $\bar{S}$. So, our problem is to find the cut with minimum capacity in the given graph.

Note 2. For capacity of the cut $(S, \bar{S})$, we do not include edges going from $\bar{S}$ to S.

Can we write this problem as a linear program? Let us start by defining variables p_u for every vertex and $d_{u,v}$ for every edge. Say, p_u is equal to 1 if it is on *s-side* of the cut and 0 otherwise. Also, $d_{u,v}$ is 1 if the

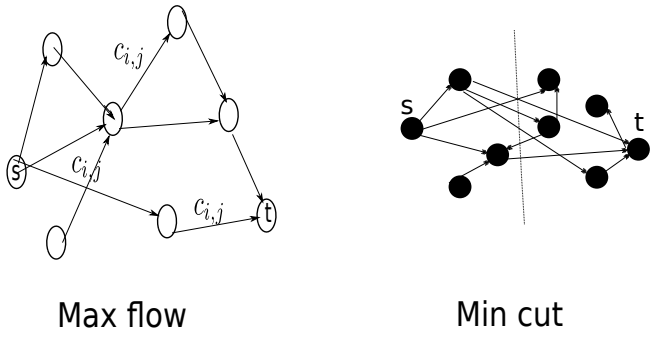

Fig. 1. Network flow problems in a graph

edge is part of the cut and 0 otherwise. An *optimization problem* for minimum cut can be written as,

$$\begin{aligned} \min \sum_{(u,v)\in E} c_{u,v} d_{u,v} & \\ \text{s.t. } d_{u,v} \geq p_u - p_v & \qquad \forall (u,v) \in E \\ p_s - p_t = 1 & \\ p_u,\ d_{u,v} \in \{0,1\} & \quad \forall u \in V,\ (u,v) \in E \end{aligned}$$

Exercise 5. Is this a linear program?

We observe that there is an unfamiliar constraint in the program above, variables are supposed to be integers. Such a constraint is not a linear constraint (feasible region is not a polytope). A linear program with integrality constraint on variables is known as an *integer linear program (ILP)*. We will call the above optimization problem as ILP for min cut.

Exercise 6. Write maximum independent set problem as an ILP.

It turns out that we can even frame NP hard problems as ILP's. This suggests that solving ILP in general might be really hard. A standard way to look at such problems is, relax the integrality constraint and make it a linear program.

$$\begin{aligned} \min \sum_{(u,v)\in E} c_{u,v} d_{u,v} & \\ \text{s.t. } d_{u,v} \geq p_u - p_v & \qquad \forall (u,v) \in E \\ p_s - p_t = 1 & \\ d_{u,v} \geq 0 & \qquad \forall (u,v) \in E \\ p_u,\ d_{u,v} \geq 0 & \quad \forall u \in V,\ (u,v) \in E \end{aligned}$$

Exercise 7. Show that the optimal of this linear program is at most the optimal of the ILP for min cut.

Notice that all solution of ILP will be solutions of LP, these are known as *integral solutions*. There can be other solutions of LP, those are non-integral solutions.

Exercise 8. Show that we can replace $p_s - p_t = 1$ by $p_s - p_t \geq 1$ without changing the objective of the linear program.

Making these two modifications, we get the linear program.

$$\begin{aligned} \min \sum_{(u,v)\in E} c_{u,v} d_{u,v} & \\ \text{s.t. } d_{u,v} \geq p_u - p_v & \quad \forall (u,v) \in E \\ p_s - p_t \geq 1 & \\ d_{u,v} \geq 0 & \quad \forall (u,v) \in E \\ p_u \geq 0 & \quad \forall u \in V \end{aligned}$$

Let us call this the LP for min cut. Remember that the optimal value of this LP is at most the value of min cut (optimal value of ILP for min cut). We will show in Sec. 2 that the two values are exactly equal.

Note 3. We are lucky in the case of min-cut. In most of the cases, relaxing the integrality constraint gives only a bound on the original value of interest. In those cases, we are interested in showing that the value given by LP approximates the value given by ILP *well*.

First, let us look at the two linear programs, one for max flow and other for min cut. Do you see any relation between them?

Let's construct the dual of the above mentioned linear program for max flow (Eqn. 1). Dual program will have variable $d_{u,v}$ for every edge and p_u for every vertex. Using these variables, the dual can be written as:

$$\begin{aligned} \min \sum_{(u,v)\in E} c_{u,v} d_{u,v} & \\ \text{s.t. } d_{u,v} \geq p_u - p_v & \quad \forall (u,v) \in E \\ p_s - p_t \geq 1 & \\ d_{u,v} \geq 0 & \quad \forall (u,v) \in E \\ p_u \geq 0 & \quad \forall u \in V \end{aligned}$$

Note 4. There is no constraint corresponding to the edge (t, s) (and no variable $d_{t,s}$). Also, the quantities p_u are translation invariant, i.e., all of them can have a constant subtracted or added if that keeps them positive.

It turns out that max flow LP and min cut LP are dual of each other! By strong duality, the LP value of min cut is same as the LP value of max flow. Remember, we still haven't shown that the LP value of min cut is equal to the value of min cut (value of ILP of min cut).

It is interesting to note that the capacity of any cut is an upper bound on the maximum flow possible between s and t. Actually this statement is easy to verify by looking at the cut picture in Fig. 1.

This is a special case of weak duality, when the feasible solution of min cut LP is integral. So, we know that the max flow LP is a lower bound on the value of min cut. Let us show that all these values; max flow, LP value for min cut and ILP value for min cut are same.

2 Max flow and min cut theorem

Let's take a look at the optimal solutions for the primal and dual formulation of max flow. Since $f_{u,v} = 0$ for all edges is a feasible solution for primal and also there is an upper bound on the maximum flow, both primal and dual are feasible and optimal solutions exist.

Say $f^*_{u,v}$ and $d^*_{u,v}, p^*_u$ are optimal solutions for the primal and dual. We can assume that s and t are connected and hence flow is positive. By complementary slackness,

1. $f^*_{u,v} > 0 \Rightarrow d^*_{u,v} = p^*_u - p^*_v$
2. $f^*_{u,v} < c_{u,v} \Rightarrow d_{u,v} = 0$, taking negation, $d^*_{u,v} > 0 \Rightarrow f^*_{u,v} = c_{u,v}$

3. $p^*_s - p^*_t = 1$
4. $p^*_v > 0 \Rightarrow \sum_{u:\{u,v\}\in E} f^*_{u,v} = \sum_{u:\{v,u\}\in E} f^*_{v,u}$

Exercise 9. Why is the third condition true?

We know that the last condition is trivially satisfied, so let us worry about the first three conditions only. If we can show a primal/dual solution which satisfies first three conditions, then it is an optimal solution for our primal/dual combination (by complementary slackness). The following argument creates an *integral* solution for the dual using the optimal solution $f^*_{u,v}$ satisfying complementary slackness.

Consider a new graph with edges (u, v) when $f^*_{u,v} < c_{u,v}$ and edges (v, u) when $f^*_{u,v} > 0$ holds. Notice that the second set of edges will be reversed. The graph with these edges is called the *residual graph* for the flow $f^*_{u,v}$.

Since the flow is optimum, vertices s and t are not connected in the residual graph.

Exercise 10. Show that if the vertices s and t are connected in the residual graph, then we can increase the flow along the path to contradict the optimality of $f^*_{u,v}$.

Suppose, S is the set of vertices connected to s in the residual graph. The dual solution corresponds to the cut $S, \bar{S}$. So, if $u \in S$ then $p_u = 1$ otherwise 0. The edges (u, v) which go from S to $\bar{S}$ are assigned $d_{u,v} = 1$ and others 0.

Looking at complementary slackness conditions for this dual solution, clearly $p_s - p_t = 1$, showing the third condition. For the second condition, $d^*_{u,v} > 0$ implies that the edge (u, v) is part of the cut in residual graph. So, (u, v) is not present in the residual graph, implying $f^*_{u,v} = c_{u,v}$.

Exercise 11. Check that this dual solution satisfies the first condition for complementary slackness given above.

For an optimal primal solution we get a dual solution which satisfies the complementary slackness conditions, showing that the dual solution is also optimal. Notice that the proposed solution works not just for the LP of min cut but also for the ILP of min cut.

Since LP value of min cut is equal to the value of max-flow by strong duality, this implies that the value of min cut (value of ILP for min cut) is not just an upper bound on the value of max flow but exactly equal. We will write out the consequences.

- Max flow = min cut. In a graph, the value of max flow is equal to value of min cut.
- The dual LP for max flow has the same value as the ILP for min cut. In other words, the ILP value of min cut is same as the LP value of min cut.
- If capacities are all integer then min cut is an integer, implying that the max flow is an integer too.

  Note 5. Actually, if capacities are integer then there exists a max-flow for which flow in every edge is an integer. This is called the *integrality theorem* in networks. We will not go through its proof here.

Maximum flow and minimum cut is a great example of the power of duality. Even without solving the concerned LP's, we have derived a lot of information about optimal solutions.

The next section will be devoted to finding the optimal value of these two LP's. You might have seen the algorithm before, the presentation here will be inspired from linear programming. As a bonus, we will learn a general technique which is useful in many other problems.

3 Overview of the primal dual approach

Now, we will look at one of the approaches to solve the linear program for maximum flow (that will solve the min cut problem too). The approach, called *primal dual approach*, is quite general and can be applied to a huge class of linear programs. This section will give a brief overview of this primal dual approach.

Suppose the LP's are given in this standard form. We will assume that $b \geq 0$, it will gives us a feasible solution to start with.

Primal LP

$$\begin{aligned} \max\ & c^T x \\ \text{s.t. } & Ax - b \leq 0 \end{aligned}$$

Dual LP

$$\begin{aligned} \min\ & b^T y \\ \text{s.t. } & A^T y - c = 0 \\ & y \geq 0 \end{aligned}$$

Again, we assume that A is an $m \times n$ matrix. The algorithm starts with a feasible primal solution ($x = 0$ is feasible since $b \geq 0$). At a particular iteration, suppose we are at solution x^*.

The idea is to improve the feasible solution x^* till the complementary slackness conditions get satisfied. In other words, using x^*, we will try to create a dual solution (say y^*) satisfying complementary slackness (though it need not be dual feasible). Obviously, we should try to minimize the violation by the new dual solution y^*. Such an LP to find y^* is called the *restricted dual LP*.

Suppose, S is the index set of all the constraints in primal which are strictly followed by x^*. Then, by complementary slackness, we know that only the corresponding variables in dual can be non-zero. The restricted dual LP is,

Restricted dual LP

$$\begin{aligned} \min\ & \sum_i z_i \\ \text{s.t. } & A^T_{*,i} y + z_i = c_i \quad \forall i \in [n] \\ & y_j \geq 0 \quad \forall j \in S \\ & y_j = 0 \quad \forall j \notin S \\ & z_i \geq 0 \quad \forall i \in [n] \end{aligned}$$

We have used $A_{*,i}$ to denote the i-th column of A; similarly, we will use $A_{i,*}$ to denote the i-th row. Variables z_i are introduced to measure the violation in dual feasibility.

If we can find a solution y^*, z^* of restricted dual such that the objective value is zero, our original x^* is optimal by complementary slackness. Otherwise, we should improve x^*. Next idea is to improve x^* using the solution of the dual of the restricted dual LP (called *restricted primal LP*). Intuitively, a solution of restricted primal LP should give us a direction to satisfy complementary slackness.

We show that the idea works below, let us take a look at the restricted primal LP first.

Restricted primal LP

$$\begin{aligned} \max\ & c^T x \\ \text{s.t. } & A^T_{j,*} x \leq 0 \quad \forall j \in S \\ & x_i \leq 1 \quad \forall i \in [n] \end{aligned}$$

Say x' is the restricted primal solution. We need to show two things,

- First, $x^* + \theta x'$ for a carefully chosen value of θ is feasible for primal with a better objective value. You will show this in the assignment.
- If x^* is not optimal, there exist a feasible solution of restricted primal LP.

Exercise 12. Let x^0 be the optimal solution of the original LP (and x^* is not optimal). Show that a suitable scaling of $x^0 - x^*$ will be a feasible solution of the restricted primal LP with positive objective value.

So, we have the approach (called primal-dual),

1. Start with a feasible solution x of the primal LP.
2. In every iteration, construct the restricted dual and restricted primal LP.
3. If the optimal value of restricted LP's is zero, x is optimal.
4. Otherwise, use the restricted primal LP solution to construct a feasible z (for primal LP) with better objective value of the primal LP.

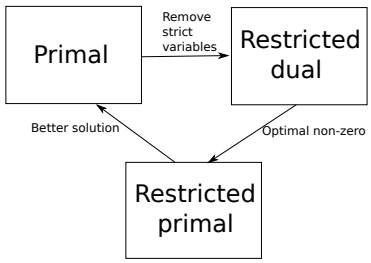

Fig. 2. Primal dual algorithm

The biggest question here is, we wanted to solve LP and now we need to solve restricted LP, what is the advantage?

It turns out that in many cases restricted LP is much simpler to solve as compared to the original LP; for example, the RHS of constraints for restricted LP is zero. The restricted LP can potentially be solved directly (combinatorial algorithm, without using LP solvers). We will see one such application in the next section.

4 Application to max flow

We have already described the LP of max flow for graph $G(V, E)$ before (Eqn. 1).

Let A be the edge incidence matrix ($|V| \times |E|$) for graph G. That means, a_{ie} is $1(-1)$ if i is the tail (head) respectively of the edge e, otherwise $a_{ie} = 0$.

We order the vertices so that the first vertex is s and the second one is t. Let c be the capacity vector, the LP for max flow can be written as,

$$\begin{aligned} & \max \quad v \\ \text{s.t.} \quad Af + dv \leq 0 \quad & d = (+1, -1, 0, \cdots, 0)^T \\ & f \leq c \\ & f \geq 0 \end{aligned}$$

Exercise 13. Show that the LP for maximum flow defined before (Eqn. 1) is equivalent to the one above.

Then, the restricted primal is (with f_e^* as the feasible solution for primal LP),

$$\begin{aligned} & \max \quad v \\ \text{s.t.} \quad & Af + dv \leq 0 \quad d = (+1, -1, 0, \cdots, 0)^T \\ & f_e \leq 0 \quad \forall e \in E : \; f_e^* = c_e \\ & f_e \geq 0 \quad \forall e \in E : \; f_e^* = 0 \\ & f \leq 1, v \leq 1 \end{aligned}$$

Exercise 14. Show that this is the restricted primal.

We only need a feasible solution for this LP with objective value greater than zero. A feasible solution of this LP is a total positive flow from s to t with only constraints being, $f_e \leq 0$ for edges when $f_e^* = c_e$ and $f_e \geq 0$ for edges when $f_e^* = 0$.

The easiest solution for such a feasible flow is to find an *undirected* path P from s to t in G, s.t.,

- If $e \in E$ is a forward edge in path then $f_e^* < c_e$,
- If $e \in E$ is a backward edge (travelled in reverse direction) in path then $f_e^* > 0$.

Such a path P is called an *augmenting path*. Existence of augmenting path tells us that the value of the flow can be increased till some forward edge gets saturated or backward edge gets empty.

Exercise 15. Show that if there is an augmenting path then the value of the flow can be increased.

You will show in the assignment that the restricted primal LP above has value 1 if there is an augmenting path otherwise 0. We give an informal argument. Every augmenting path is a solution to the LP is a simple exercise. For showing that the optimal solution of LP (if optimal value is greater than zero) can be converted into an augmenting path. Start with the least flow value in an edge and construct a path from s to t with that flow value. That will be the augmenting path and LP value will be 1.

So, if the restricted primal has optimal value 0 then complementary slackness conditions are satisfied and the flow is optimal. If not, then we can increase the flow using the augmenting path.

We don't need to solve the LP above (through simplex or some other method) and can directly find an augmenting path if the flow is not optimal. The algorithm to find an augmenting path is known as the *labelling algorithm* and is described in the next section.

4.1 Algorithm to find augmenting path

This algorithm runs on the undirected version of the given directed graph.

The idea of the algorithm is to label vertices starting from s. Every vertex $x \in V(G)$ is assigned label $(s(x), f(x))$. First entry tell us how to reach x (from which vertex) and second tell us how much flow can be put into it. If we reach t then we have an augmenting path.

Let us label s by (s, ∞). At every step, algorithm will pick a labelled vertex and try to label its neighbours (called scanning from that vertex). If the neighbor is already labelled, we will ignore it. It can label a neighbour in these ways:

- Suppose x is labelled and $(x, y) \in E$ is an edge, s.t., $f_{x,y} < c_{x,y}$, then $s(y) = x$ and $f(y) = \min(s(x), c_{x,y} - f_{x,y})$.
- Suppose x is labelled and $(y, x) \in E$ is an edge, s.t., $f_{y,x} > 0$, then $s(y) = -x$ and $f(y) = \min(s(x), f_{y,x})$.

To make sure that we don't scan a vertex more than once, we will keep a list of labelled but unscaned vertices. It will initially contain only s. At every step, labelling algorithm picks an element from this list, scans it and deletes it from the list. Then, we include all the newly labelled vertices in the list of this round.

The algorithm can terminate in two ways, either we will label t or the list of labelled but unscanned vertices become empty.

Exercise 16. Show that this algorithm will terminate in finite steps.

If t is labelled then we have the augmenting path. In the other case, we can construct a cut with labelled vertices and unlabeled vertices. This cut satisfies the complementary slackness conditions with the given flow. Hence, both the flow as well as the cut are optimal.

Exercise 17. Show that the cut obtained between labelled and unlabelled vertices satisfies complementary slackness conditions.

The complete algorithm to find max flow is known as Ford-Fulkerson algorithm. It terminates in finite iterations if all capacities are integer. It is known that in some cases (when capacities are irrational) the algorithm can run for infinite iterations.

5 Assignment

Exercise 18. Show that exchanging equality in flow conservation with inequality does not change the LP.

Exercise 19. Show that if the flow is optimal then s and t are not connected in the *residual graph.*

Exercise 20. Show that if x' is the solution for restricted primal (with x^* as the feasible primal), then $x^* + \theta x'$ will be feasible for primal with a better objective value (for a carefully chosen θ). Think about, what conditions should θ satisfy? Why positive θ increases the objective value?

Exercise 21. Show that the restricted primal LP for max-flow has value 1 if there is an augmenting path otherwise 0.

Exercise 22. Write the Ford-Fulkerson algorithm in steps.

Exercise 23. A spanning tree T of an undirected graph $G = (V, E)$ is a subset of edges E, s.t., the edges form a tree and any two vertices are connected by a path in T. Given a cost c_e for every edge in E, *minimum spanning tree* is the problem to find the spanning tree with minimum cost. Here, the cost of a tree is the sum of cost of all the edges in the tree.

Remember that a cut is specified by a subset $S \subseteq V$ and its complement. Look at the following linear program,

$$\begin{aligned} \min \quad & \textstyle\sum_e c_e x_e \\ \text{s.t.} \quad & \textstyle\sum e \text{ crosses } \Pi x_e \geq 1 \quad \text{for all cuts } \Pi \\ & x_e \geq 0 \\ & x_e \leq 1 \\ & x_e \in \mathbb{Z}. \end{aligned}$$

1. Show that the above ILP calculates the cost of minimum spanning tree.
2. Removing the integrality constraint, it becomes a linear program. Find the dual of that LP.

A partition Π of a set S by sets $A_1, A_2, \cdots, A_n$ means,

- $A_1, A_2, \cdots, A_n$ are mutually disjoint.
- Union of $A_1, A_2, \cdots, A_n$ is set S.

The size of partition, $|\Pi|$, is n in this case.

Exercise 24. Difficult: Show that even the following ILP characterizes the value of minimum spanning tree.

$$\begin{aligned} \min \quad & \sum_e c_e x_e \\ \text{s.t.} \ \sum e \text{ crosses } \Pi & x_e \geq |\Pi| - 1 \quad \text{for all partitions } \Pi \text{ of } V \\ & x_e \geq 0 \\ & x_e \leq 1 \\ & x_e \in \mathbb{Z}. \end{aligned}$$

References