Multicut and Integer Multicommodity Flow in Trees

Approximation Algorithms,

V. V. Vazirani

Book Chapter 18

Nicolas Karakatsanis, 2008

Introduction

The theory of cuts in graphs occupies a central place not only in the study of exact algorithms, but also approximation algorithms

Primal-Dual schema was used to derive a factor 2 algorithm for the weighted vertex cover problem

For that algorithm, the relaxed dual complementary slackness conditions were automatically satisfied in any integral solution

In this chapter, we will use the primal-dual schema to obtain an algorithm for a generalization of this problem

This time, enforcing relaxed dual complementary slackness conditions will be a non-trivial part of the algorithm

Moreover, the procedure of reverse delete will be introduced (useful for several other primal-dual algorithms

The problems and their LP-relaxations

The following is an important generalization of the minimum $s-t$ cut problem. It also generalizes the multiway cut problem (Chapter 4)

Problem 18.1 (Minimum multicut)

Let $G=(V,E)$ be an undirected graph with nonnegative capacity c_e for each edge $e \in E$. Let $\{(s_1,t_1),\dots,(s_k,t_k)\}$ be a specified set of pairs of vertices, where each pair is distinct, but vertices in different pairs are not required to be distinct. A *multicut* is a set of edges whose removal separates each of the pairs. The problem is to find a minimum capacity multicut in G.

The minimum $s - t$ cut problem is the special case of multicut for $k=1$. Problem 18.1 generalizes multiway cut because separating terminals $s_1,\dots,s_k$ is equivalent to separating all pairs (s_i,s_j), for $1 \le i < j \le 1$. Therefore, the minimum multicut problem is NP-hard even for $k=3$, since the the multiway cut problem is NP-hard for the case of 3 terminals.

The problems and their LP-relaxations

In chapter 20 an $O(\log k)$ factor approximation algorithm will be obtained for the minimum multicut problem.

In this chapter a factor 2 algorithm will be obtained for the special case when $G = (V, E)$ is restricted to be a tree.

Since G is a tree, there is a unique path between s_i and t_i, and the multicut must pick an edge on this path to disconnect s_i from t_i.
The minimum multicut problem is NP-hard even if restricted to trees of height 1 and unit capacity edges.

The minimum multicut problem is NP-hard even if restricted to trees of height 1 and unit capacity edges.

Integer Minimum Multicat

Integer programming formulation

minimize $\sum_{e \in E} c_e d_e$

subject to $\sum_{e \in p_i} d_e \geq 1 \quad i \in \{1, \ldots, k\}$

$d_e \in \{0,1\}, \quad e \in E$

Linear Programming Relaxation

minimize $\sum_{e \in E} c_e d_e$

subject to $\sum_{e \in p_i} d_e \geq 1, \quad i \in \{1, \ldots, k\}$

$d_e \geq 0 \quad e \in E$

Consider d_e as the fractional extent to which edge e is picked.

Fractional multicut: A solution to this LP-relaxation problem

Condition: On each path p_i, the sum of fractions of edges picked is at least 1.

In general, minimum fractional multicut may be strictly cheaper than minimum integral multicut

Fractional multicommodity flow

Problem 18.2 (Fractional mutlicommodity flow)

We will interpret the dual program as specifying a multicommodity flow in ***G***, with a separate commodity corresponding to each vertex pair (s_i, t_i) . Dual variable f_i will denote the amount of this commodity routed along the unique path from s_i to t_i .

Dual program formulation (18.2)

$$
\begin{aligned}
\text{maximize} \quad & \sum_{i=1}^{k} f_i \\
\text{subject to} \quad & \sum_{i: e \in p_i} f_i \le c_e, \quad e \in E \\
& f_i \ge 0 \qquad i \in \{1, \ldots, k\}
\end{aligned}
$$

The commodities are routed concurrently.

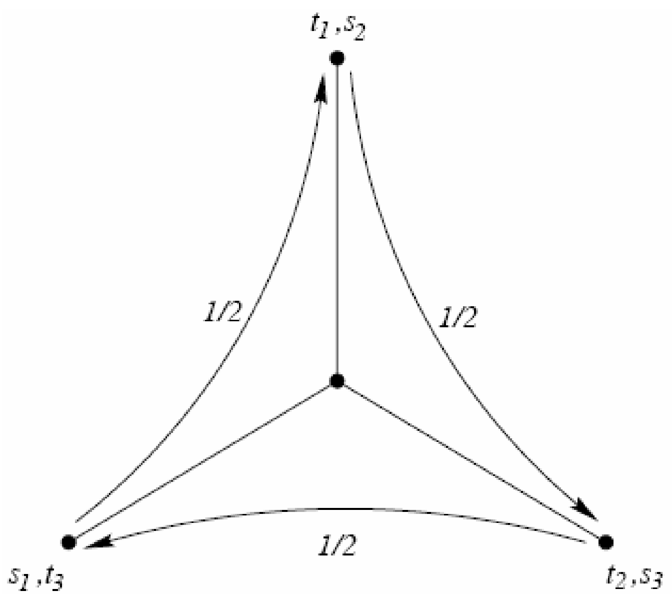
The object is to maximize the sum of the commodities routed, subject to the constraint that the sum of flows routed through an edge is bounded by the capacity of the edge.

Notice that the sum of flows through an edge (u, v) includes flow going in either direction, u to v and v to u

Example

Example 18.2

Consider the followig graph with unit capacity edges and 3 vertex pairs



The arrows show how to send 3/2 units of flow by sending 1/2 unit of each commodity.

Picking an edge to the extent of 1/2 gives a multicut of capacity 3/2 as well.

The above must be optimal solutions to the primal and dual programs

On the other hand, any integral multicut must pick at least two of the three edges in order to disconnect all three pairs. Hence, minimum integral multicut has capacity 2

Maximum integral multicommodity flow is 1, since sending 1 unit of any of the three commodities will saturate two of the edges.

Integer multicommodity flow

Problem 18.3 (Integer multicommodity flow)

Graph $\boldsymbol{G}$ and the source-sink pairs are specified as in the minimum multicut problem; however, the edge capacities are all integral. A separate commodity is defined for each (s_i, t_i) pair. The object is to maximize the sum of commodities routed, subject to the edge capacity constraints and subject to routing each commodity integrally.

Integer program formulation (18.3)

$$\begin{aligned} \text{maximize} \quad & \sum_{i=1}^{k} f_i \\ \text{subject to} \quad & \sum_{i:e \in p_i} f_i \le c_e, \quad e \in E \\ & f_i \in \mathbf{Z}^+ \quad i \in \{1, \ldots, k\} \end{aligned}$$

Let us consider the problem when $\boldsymbol{G}$ is restricted to be a tree.

Construct this integer program formulation by constraining the f_i varaiables to be non-negative

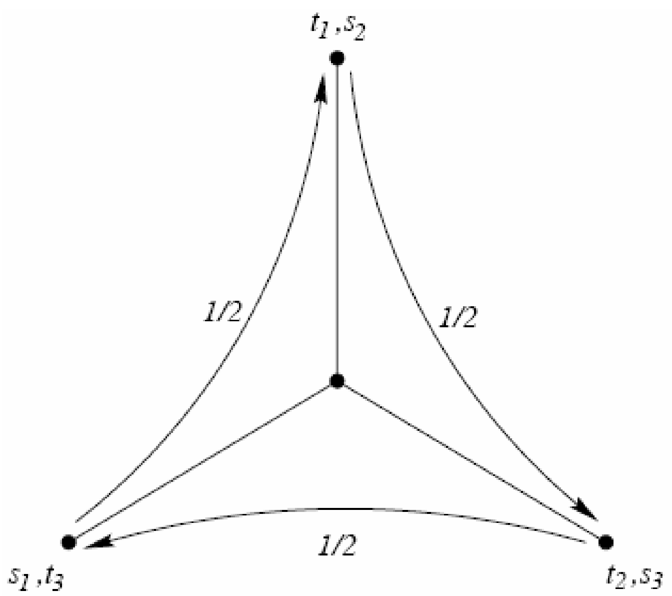
Clearly the objective function value of this problem is bounded by that of the linear program (18.2)

This problem is NP-hard even for trees of height 3 (though the capacity has to be arbitrary)

Example

Example 18.2

Consider the followig graph with unit capacity edges and 3 vertex pairs



The arrows show how to send 3/2 units of flow by sending 1/2 unit of each commodity.

Picking an edge to the extent of 1/2 gives a multicut of capacity 3/2 as well.

The above must be optimal solutions to the primal and dual programs

On the other hand, any integral multicut must pick at least two of the three edges in order to disconnect all three pairs. Hence, minimum integral multicut has capacity 2

Maximum integral multicommodity flow is 1, since sending 1 unit of any of the three commodities will saturate two of the edges.

Overview of Primal-Dual Schema

Let us consider the following programs, written in standard form

Primal program

minimize $\sum_{j=1}^{n} c_j x_j$

subject to $\sum_{j=1}^{n} a_{ij} x_j \geq b_i, \quad i = 1, \ldots, m$

$x_j \geq 0 \qquad j = 1, \ldots, n$

Dual program

maximize $\sum_{i=1}^{m} b_i y_i$

subject to $\sum_{i=1}^{m} a_{ij} y_i \leq c_j, \quad j = 1, \ldots, n$

$y_i \geq 0, \qquad i = 1, \ldots, m$

Primal complementary slackness conditions

Let $\alpha \geq 1$

For each $1 \leq j \leq n$: either $x_j = 0$ or $c_j / \alpha \leq \sum_{i=1}^{m} a_{ij} y_i \leq c_j$.

Dual complementary slackness conditions

Let $\beta \geq 1$

For each $1 \leq i \leq m$:either $y_i = 0$ or $b_i \leq \sum_{j=1}^{n} a_{ij} x_j \leq \beta \cdot b_i$.

Proposition 15.1: *If* ***x*** *and* ***y*** *are primal and dual feasible solutions satisfying the conditions stated above then:* $\sum_{j=1}^{n} c_j x_j \leq \alpha \cdot \beta \cdot \sum_{i=1}^{m} b_i y_i$

Primal-Dual schema based algorithm

We will use the primal-dual schema to obtain an algorithm that simultaneously finds a multicut and an integer multicommodity flow that are within a factor of 2 of each other, provided the given graph is a tree.
Hence, we get approximation algorithms for both problems, of factor 2 and $1/2$.

Let us define the multicut LP to be the primal program. An edge *e* is *saturated* if the total flow through it equals its capacity.

We will ensure primal complementary slackness conditions, i.e., $\alpha = 1$.
We will relax the dual complementary slackness conditions with $\beta = 2$.

Primal conditions: For each $e \in E, d_e \neq 0 \Rightarrow \sum_{i: e \in p_i} f_i = c_e$
Equivalently, *any edge picked in the multicut must be saturated*

Relaxed dual conditions: For each $i \in \{1, \ldots, k\}, f_i \neq 0 \Rightarrow \sum_{e \in p_i} d_e \leq 2$
Equivalently, *at most two edges can be picked from a path carrying nonzero flow*

Root the tree at an arbitrary vertex. Define the depth of vertex v to be the length of the path from v to the root; the depth of the root is 0.
For two vertices $u, v \in V$, let $lca(u, v)$ denote the *lowest common ancestor* of u and v, i.e. the minimum depth vertex on the path from u to v.

Primal-Dual schema based algorithm

Algorithm 18.4 (Multicut abd integer multicommodity flow in trees)

1. **Initialization:** $f \leftarrow 0; D \leftarrow 0$
2. **Flow Routing:** For each vertex υ, in nonincreasing order of depth, do:
   For each pair (s_i, t_i) such that $lca(s_i, t_i) = \upsilon$, greedily route integral flow from s_i to t_i.
   Add to D all edges that were saturated in the current iteration in arbitrary order.
3. Let $e_1, e_2, \ldots, e_l$ be the ordered list of edges in D.
4. **Reverse delete:** For $j=l$ downto 1 do:
   If $D - \{e_j\}$ is a multicut in G, then $D \leftarrow D - \{e_j\}$.
5. Output the flow and multicut in D.

Primal-Dual schema based algorithm

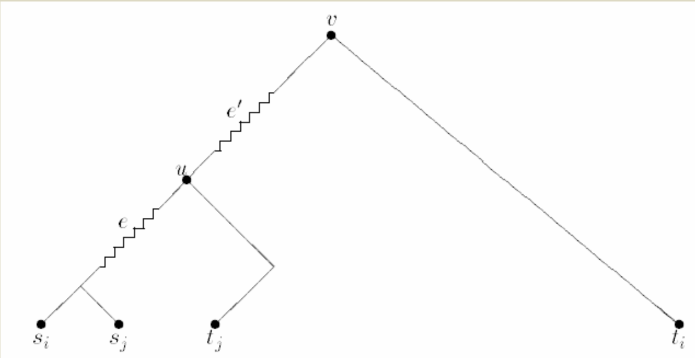
Lemma 18.5

Let (s_i, t_i) be a pair with nonzero flow, and let $lca(s_i, t_i) = \upsilon$. At most one edge is picked in the multicut from each of the two paths, s_i to υ and t_i to υ.

Proof

The argument is the same for each path. Suppose two edges e and e' are picked from the $s_i - \upsilon$ path, with e being the deeper edge. Clearly, e' must be in D all through reverse delete. Consider the moment during reverse delete when edge e is being tested. Since e is not discarded, there must be a pair, say (s_j, t_j), such that e is the only edge of D on the $s_j - t_j$ path. Let u be the lowest common ancestor of s_j and t_j. Since e' does not lie on the path of $s_j - t_j$, then u must be deeper than e' and hence deeper than υ.

After u has been processed, D must contain an edge from the $s_j - t_j$ path, say e''. Since non-zero flow has been routed from s_i to t_i, e must be added during or after the iteration in which is processed. Since υ is ancestor of u, e is added after υ. So e'' must be in D when e is being tested. This contradicts that at this moment e is the only edge of D on the $s_j - t_j$ path



Primal-Dual schema based algorithm

Theorem 18.6

Algorithm 18.4 achieves approximation guarantees of factor 2 for the minimum multicut problem and factor $1/2$ for the maximum integer multicommodity flow problem on trees.

Proof

The flow found at the end of Step 2 is maximal, and since at this point D contains all the saturated edges, D is a multicut. Since the reverse delete step only discards redundant edges, D is a multicut after this step as well. Thus, feasible solutions have been found for both the flow and the multicut.
Since each edge in the multicut is saturated, the primal conditions are satisfied. By Lemma 18.5, at most two edges have been picked in th multicut from ach path carrying nonzero flow. Therefore, the relaxed dual conditions are also satisfied. Hence, by proposition 15.1, the capacity of the multicut found is within twice the flow.
Since a feasible flow is a lower bound on the optimal multicut, and a feasible multicut flow is an upper bound on the optimal integer multicommodity flow, the claim follows.

Primal-Dual schema based algorithm

Corollary 18.7

On the trees with integer edge capacities, we obtain the following min-max relationship, based on theorem 18.6:

$$\max_{\text{int. flow } F} |F| \le \min_{\text{multicut } C} |C| \le 2 \max_{\text{int. flow } F} |F|,$$

where $|F|$ represents the value of flow function F and $c(C)$ represents the capacity of multicut C.

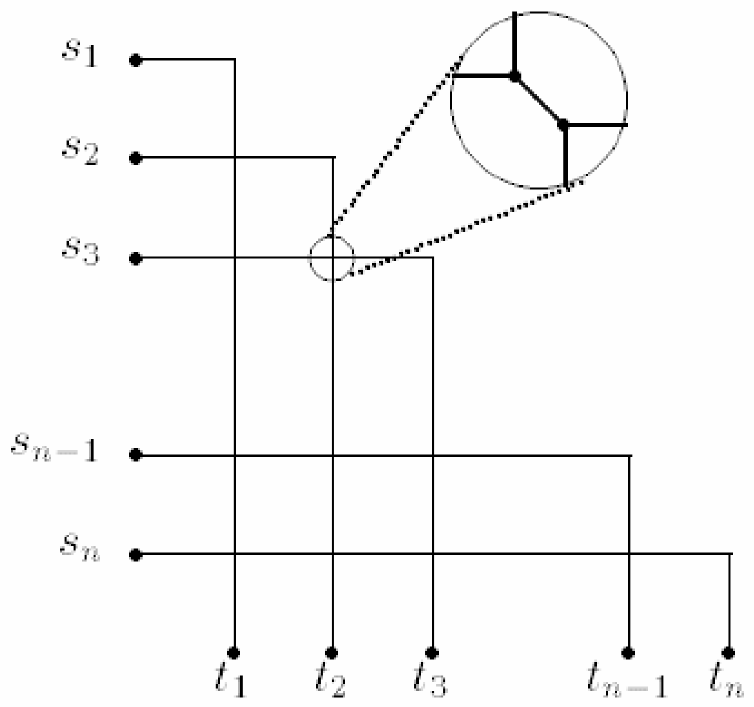
In chapter 20 is presented an $O(\log k)$ factor algorithm for the minimum multicut problem in general graphs; once again, the lower bound used is an optimal fractional multicut.
On the other hand, no notrivial approximation algorithms are known for the integer multicommodity flow problem in graphs more general than trees.

As shown in example 18.8, even for planar graphs, the integrality gap of an LP analogous to (18.2) is lower bounded by $n/2$, where n is the number of source-sink pairs specified

Example

Example 18.8

Consider the following planar graph with n source-sink pairs.



Every edge is of unit capacity.

Any pair of paths between the ith and jth source-sink pairs intersect in at least one unit capacity edge.

The magnified part shows how this is arranged at each intersection.

Thus sending one unit of any commodity blocks all other commodities.

On the other hand, hald a unit of each commodity can be routed simultaneously

Therefore, the integrality gap for this example is $n/2$