The Design and Analysis of Algorithms

Lecture 21 Approximation Algorithm II

Zhenbo Wang

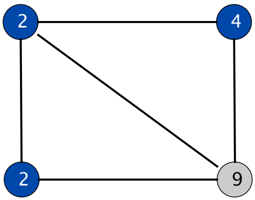
Department of Mathematical Sciences, Tsinghua University

Content

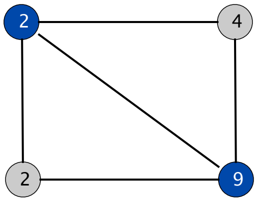
Weighted Vertex Cover

Def. Given a graph $G = (V, E)$, a vertex cover is a set $S \subseteq V$ such that each edge in E has at least one end in S.

- *Weighted Vertex Cover.* Given a graph G with vertex weights, find a vertex cover of minimum weight.



weight = 2 + 2 + 4

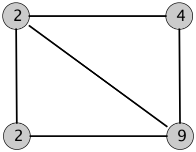


weight = 11

Pricing Method

- *Pricing method.* Each edge must be covered by some vertex. Edge $e = (i, j)$ pays price $p_e \geq 0$ to use both vertex i and j.
- *Fairness.* Edges incident to vertex i should pay $\leq w_i$ in total.

for each vertex i : $\sum_{e=(i,j)} p_e \leq w_i$



Lemma. For any vertex cover S and any fair prices p_e : $\sum_e p_e \leq w(S)$.

Pf.

$$\sum_{e \in E} p_e \leq \sum_{i \in S} \sum_{e=(i,j)} p_e \leq \sum_{i \in S} w_i = w(S). \square$$

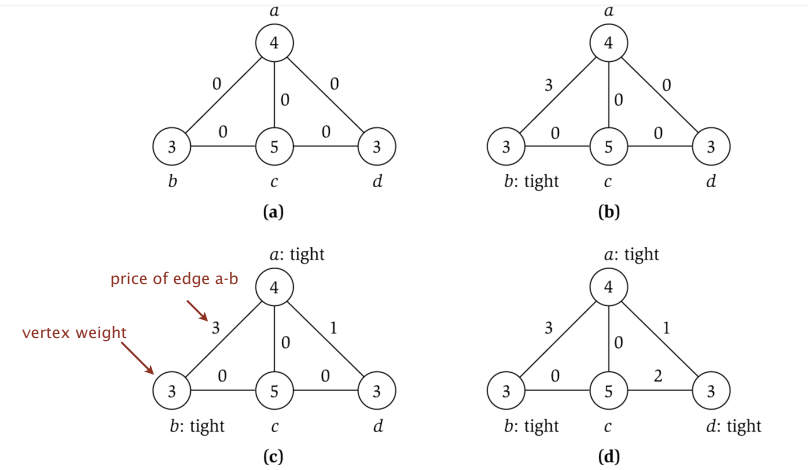
Pricing Method

- Set prices and find vertex cover simultaneously.

WEIGHTED – VERTEX – COVER(G, w)

$S \leftarrow \emptyset$
for each $e \in E$ **do**
    $p_i \leftarrow 0$.
end for
while there exists an edge (i, j) such that neither i nor j is tight) **do**
    Select such an edge $e = (i, j)$.
    Increase p_e as much as possible until i or j is tight.
end while
$S \leftarrow$ set of all tight nodes.
return S.

Pricing Method: Example



Pricing Method: Analysis

Theorem 1

Pricing method is a 2-approximation for WEIGHTED VERTEX COVER.

Pf. Algorithm terminates since at least one new node becomes tight after each iteration of while loop.

Let S = set of all tight nodes upon termination of algorithm.

S is a vertex cover: if some edge (i, j) is uncovered, then neither i nor j is tight. But then while loop would not terminate.

Let S^* be optimal vertex cover. We show $w(S) \leq 2w(S^*)$.

$$w(S) = \sum_{i \in S} w_i = \sum_{i \in S} \sum_{e=(i,j)} p_e \leq \sum_{i \in V} \sum_{e=(i,j)} p_e = 2 \sum_{e \in E} p_e \leq 2w(S^*). \square$$

Local-Ratio Technique

Given: A set of feasibility constraints $\mathcal{F}$ on $x \in \{0,1\}^n$, A nonnegative weight function w.

Goal: Minimize (or Maximize) $w \cdot x$

Theorem 2 (Local-Ratio Theorem)

Let w, w_1, $w - w_1$ be nonnegative weight functions. Then, if x is an r-approx w.r.t. $(\mathcal{F}, w_1)$ and $(\mathcal{F}, w - w_1)$, then x is an r-approx w.r.t. $(\mathcal{F}, w)$.

Intuition:

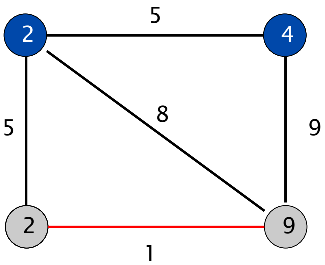
- Find a good weight function w_1.
- Solve recursively for $w - w_1$.
- The Pricing Method for Vertex Cover is actually the Local-Ratio Technique.

Prize Collecting Vertex Cover

Prize Collecting Vertex Cover (PCVC):

Def. Given $G = (V, E)$, $w : V \cup E \to R^+$

Sol. Find $C \subseteq V$ and $F \subseteq E$, such that for $E \setminus F$, C is a vertex cover, and minimize $\sum_{v \in C} w(v) + \sum_{e \in F} w(e)$.



weight = 2 + 4 + 1
= 7

Local Ratio Algorithm for PCVC

LOCAL – RATIO – PCVC(G, w)

while there exists an edge $e = (u, v) \in E$, such that $\min\{w(u), w(v), w(e)\} > 0$ **do**
    $\epsilon \leftarrow \min\{w(u), w(v), w(e)\}$.
    $w(u) \leftarrow w(u) - \epsilon$.
    $w(v) \leftarrow w(v) - \epsilon$.
    $w(e) \leftarrow w(e) - \epsilon$.
end while
return $C = \{u \mid w(u) = 0\}$ and $F = \{e = (u, v) \mid w(u), w(v) > 0\}$.

Local Ratio Analysis for PCVC: A Simple 3-approach

- When the algorithm halts, every edge is covered by C or F, and thus the cost is zero.
- Consider the ith iteration. Let $e = (u_i, v_i)$ be the edge involved, and let ϵ_i be the amount by which the weights are reduced.
- In the ith round, we pay $3\epsilon_i$ and effect a drop of at least ϵ_i in the optimal cost.
- Since every feasible solution must either contain one (or both) of u_i and v_i, or else pay for the edge e_i.
- Hence tht local ratio between our payment and the drop in OPT is at most 3 in every round.
- Thus, a factor 3 is immediate.

Local Ratio Analysis for PCVC: A tight 2-approach

- We must either pay for e_i, or for one or both of u_i and v_i, but never for all three.
- Of the $3\epsilon_i$ we seem to pay, only $2\epsilon_i$ are actually ”consumed”.
- Thus, the approximation ratio is 2.

Linear Programming (LP)

$$(LP) \quad \begin{array}{ll} \min & c^T x \\ s.t. & Ax = b, \\ & x \geq 0, \ x \in R^n. \end{array} \tag{1}$$

- A is a $m \times n$ matrix($m < n$), assume $r(A) = m$.
- Let $A = (B, N)$, where B are m linear independent column vectors of A.
- The m variables corresponding to the m basis vectors, say x_B, are called basic variables; The remaining $n - m$ vectors, say x_N, are called nonbasic variables, and $x = \begin{pmatrix} x_B \\ x_N \end{pmatrix}$.
- Let $x_N = 0$, we have $x = \begin{pmatrix} B^{-1}b \\ 0 \end{pmatrix}$, is called a basic solution, if it satisfies $B^{-1}b \geq 0$, then we call it x a basic feasible solution.

The Properties of LP

- If the feasible domain exists, then it is a convex polyhedra (may be unbounded).
- If it has optimal solution, then it can be achieved at the vertex of the feasible domain (may be unbounded).
- A basic feasible solution corresponds to a vertex of the feasible domain (extreme point).
- Simplex method is a well-used algorithm for linear programming. Its basic idea is to shift from one vertex to the next, such that the objective value decreases.
- Ellipsoid method and interior point method can solve linear programming in polynomial time.

The primal problem (P) and dual problem (D) of LP

$$(P)\quad \begin{array}{l} \min\ c^T x \\ s.t.\ Ax \geq b, \\ x \geq 0,\ x \in R^n. \end{array} \qquad (D)\quad \begin{array}{l} \max\ b^T y \\ s.t.\ A^T y \leq c, \\ y \geq 0,\ y \in R^m. \end{array}$$

- *Weak duality*. If x is a feasible solution to (P), and y a feasible solution to (D), then $c^T x \geq b^T y$.
- *Corollary*. If (P) is unbounded, then (D) is infeasible.
- *Strong duality*. If (P) and (D) are feasible, then both of them have optimal solutions, and the optimal values are equal.
- *Complementary slackness*. Let x be the feasible solutions to (P), y be the feasible solution to (D), then x and y are the optimal solution respectively iff

$$x^T(c - A^T y) = 0,$$
$$y^T(Ax - b) = 0.$$

LP Relaxation

- Many combinatorial optimizations can be represented by an integer linear programming (ILP):

$$(ILP) \quad \begin{array}{ll} \min & c^T x \\ s.t. & Ax = b, \\ & x \geq 0,\ x \in Z^n. \end{array} \tag{2}$$

- Assume that all the parameters are integers.
- If we replace $x \in Z^n$ by $x \in R^n$, we obtain a LP model:

$$(LP) \quad \begin{array}{ll} \min & c^T x \\ s.t. & Ax = b, \\ & x \geq 0,\ x \in R^n. \end{array} \tag{3}$$

- (3) is called the linear programming relaxation of (2).

Primal Dual Method

- In general, it is hard to obtain the optimal solution by solving the LP relaxation, even a feasible solution.
- Primal dual method uses the duality theory of LP, to obtain the optimal solution or approximation solution for some specific problems.
- Primal dual method is introduced by Kuhn in 1950s, he designed the Hungarian algorithm for assignment problem.
- We use the weight vertex cover as an example, to introduce the primal dual method.

D.P. Williamson. The primal dual method for approximation algorithms, Math. Program. 91 (2002) 447–478.

Primal Dual Method for Weighted Vertex Cover

The ILP of the weighted vertex cover problem is:

$$\begin{aligned} \min \quad & \sum_{i \in V} w_i x_i \\ s.t. \quad & x_i + x_j \geq 1 \quad \forall (i,j) \in E \\ & x_i \in \{0,1\}, \quad \forall i \in V. \end{aligned} \tag{4}$$

- Relax the decision variables as $x_i \geq 0$, and we obtain the dual problem of the linear programming:

$$\begin{aligned} \max \quad & \sum_{(i,j) \in E} y(i,j) \\ s.t. \quad & \sum_{k:(i,k) \in E} y(i,k) \leq w_i \quad \forall i \in V \\ & y(i,j) \geq 0 \qquad\qquad \forall (i,j) \in E. \end{aligned} \tag{5}$$

Primal Dual Method

- First set $y = 0$, that is a dual feasible solution; let $x = 0$, it is not a primal feasible solution.
- As long as x is infeasible, there is an uncovered edge (i, j), such that $x_i + x_j = 0$.
- Increase the value of dual variable $y(i, j)$ as large as possible, simultaneously satisfy the dual feasibility.
- The i or j constraint of the dual constraints (5) must satisfy the equality constraint (maybe simultaneously).
- If $\sum\limits_{k:(i,k)\in E} y(i, k) = w_i$, let $x_i = 1$;
- If $\sum\limits_{k:(j,k)\in E} y(j, k) = w_j$, let $x_j = 1$.

Primal Dual Method

- Repeat the procedure, until obtain a primal feasible solution x, it must satisfy:

$$\begin{aligned} \sum_{i \in V} w_i x_i &= \sum_{i \in V} \Big(\sum_{k:(i,k) \in E} y(i,k) \Big) x_i \\ &= \sum_{(i,j) \in E} (x_i + x_j) y(i,j) \\ &\leq 2 \sum_{(i,j) \in E} y(i,j). \end{aligned} \tag{6}$$

- The dual objective value $\sum_{(i,j) \in E} y(i,j)$ is a lower bound of the LP relaxation of the primal problem.
- Therefore, primal dual method is an 2-approximation algorithm for the weighted vertex cover problem.

Primal Dual Method for PCVC

The LP relaxation of PCVC is as follows:

$$
\begin{aligned}
\min \quad & \sum_{i \in V} w_i x_i + \sum_{(i,j)\in E} p_{(i,j)} z(i,j) & & \\
s.t. \quad & x_i + x_j + z(i,j) \geq 1 & \forall (i,j) \in E & \\
& x_i, z(i,j) \geq 0, & \forall i \in V, (i,j) \in E. &
\end{aligned}
\tag{7}
$$

The dual problem is:

$$
\begin{aligned}
\max \quad & \sum_{(i,j)\in E} y(i,j) & \\
s.t. \quad & \sum_{k:(i,k)\in E} y(i,k) \leq w_i & \forall i \in V \\
& y(i,j) \leq p(i,j) & \forall (i,j) \in E \\
& y(i,j) \geq 0 & \forall (i,j) \in E.
\end{aligned}
\tag{8}
$$

Comments

- Each primal constraint now consists of 3 variables.
- A similar argument as for the vertex cover problem can give a 3-approximation algorithm.
- We describe an extension of the primal dual algorithm that we presented for Vertex Cover as follows.

Primal Dual Algorithm for PCVC

- Start with the integer infeasible primal solution $x = 0$, $z = 0$, and the feasible dual solution $y = 0$.
- Repeat while some primal constraint is unsatisfied:

  Increase all (unfrozen) variables $y(i,j)$ until some dual constraint becomes tight.

  We have two types of dual constraints, one of which could have become tight:

1. $\sum_{k:(i,k)\in E} y(i,k) = w_i$, we set $x_i = 1$, and for all $(i,j) \in E$ incident to i, we set $z(i,j) = 0$.
2. Otherwise, $y(i,j) = p(i,j)$, we set $z(i,j) = 1$ and set $x_i = x_j = 0$.

Analysis

- For the solution x, z generated by the above algorithm, the following holds:

$$\begin{aligned}
& \sum_{i\in V} w_i x_i + \sum_{(i,j)\in E} p_{(i,j)} z(i,j) \\
= & \sum_{i\in V} \Big(\sum_{k:(i,k)\in E} y(i,k) \Big) x_i + \sum_{(i,j)\in E} y(i,j) z(i,j) \\
= & \sum_{(i,j)\in E} (x_i + x_j + z(i,j)) y(i,j) \\
\leq & \ 2 \sum_{(i,j)\in E} y(i,j).
\end{aligned} \tag{9}$$

- By weak duality, $\sum_{(i,j)\in E} y(i,j) \leq OPT$.
- Thus, the solution generated is upper bounded by $2OPT$.

Homework

- Read Chapter 11 of the textbook.
- Exercises 4 & 7 in Chapter 11.