Pricing in the Presence of Peering

Eui-woong Lee, David Buchfuhrer, Lachlan L. H. Andrew, Ao Tang, Steven H. Low

Department of Computer Science, California Institute of Technology, Pasadena, CA 91125, USA

Abstract—This paper examines a simple model of how a provider ISP charges customer ISPs by assuming the provider ISP wants to maximize its revenue while customer ISPs have the possibility of setting up peering connections. It is shown that finding the optimal pricing is NP-complete. Several algorithms are proposed and the corresponding constant approximation ratios are proved. Finally, numerical results are provided to show the average behaviors of the proposed algorithms.

I. INTRODUCTION

The Internet is an aggregation of a large number of networks owned by competing entities, which we generically call Internet service providers (ISPs). It is well known that economic consideration frequently overrides technical factors in the Internet traffic routing. In general, ISPs seek to maximize their own profit. The border gateway protocol (BGP) provides routing information, but the choice of route is determined by financial considerations as well as engineering factors.

The routes available depend on which ISPs are connected by direct links. When two unconnected ISPs want to exchange data, they must each pay a "provider" ISP to relay their data. It has been demonstrated [5] that routing has a "valley free" property, in which ISPs can be divided into tiers, with providers always in a tier above their customers. Topology measurement results [12] show that, among 4200 or so transit ISPs in the Internet today, only 15 of them are more or less fully connected. This set of ISPs are commonly referred to as tier-1 ISPs, and act as global providers. The remaining transit ISPs either directly or indirectly rely on the tier-1 ISPs for global reachability.

There has recently been much attention paid to ecomonic consequences of interdomain routing [1], [2], [8], [11] This paper concerns the optimal prices a (tier-1) provider ISP can charge its (tier-2) customers, taking into account the fact that its customers have the option of "peering" with other tier-2 networks by operating their own links to carry some or all of their traffic. In other words, the provider ISP tries to balance the trade-off between high prices and the number of subscribers.

This problem is of interest in the networking community as it is a simple model that describes the basic relations among ISPs. It serves as a starting point for future extensions to the case of competing tier-1 ISPs. As we will soon see, it also has a rich structure which is of interest to theoretical computer science.

The paper is organized as follows. Section II describes our mathematical model, which captures the optimization problem faced by a provider ISP in deciding what to charge its customers. This problem is shown in Section III to be NP hard. Section IV develops approximation algorithms with guaranteed performance to solve this problem. Numerical tests are provided in Section V to investigate the average behavior of the proposed algorithms.

II. MODEL AND NOTATION

Let $\mathbb{R}^+$ be the set of positive reals, and $\overline{\mathbb{R}}^+ = \mathbb{R}^+ \cup \{0\}$ be the set of non-negative reals, and let $|X|$ denote the cardinality of any set X.

With this notation, consider transit ISPs in two adjacent tiers, modeled as follows. Assume that all ISPs in the top tier cooperate to avoid a price war [11], so that they act as a single provider. This is modeled as a simple undirected graph $G = (V, E)$. Elements of $V = \{v_1, v_2, \ldots, v_{|V|}\}$ are called nodes, vertices or customers, and elements of E are called links or edges. We also assume that the provider is directly connected to every customer $v \in V$. Let $\mu : V \mapsto \overline{\mathbb{R}}^+$ the price function, such that $\mu(v)$ is the price customer v pays the provider when they exchange one unit of traffic, in either direction.

Let $x(u, v)$ be the total traffic volume (per unit time) in both directions between customers u and customer v. If u and v do not peer, then the traffic goes through their providers and they have to pay a total of $x(u, v)(\mu(u) + \mu(v))$, where one term is paid by the sender and the other is paid by the receiver.

For each pair u and v, there is a maximum amount $c(u, v)$ which the provider can charge for traffic sent between u and v, before u and v stop sending traffic via the provider. We will interpret this as the cost per unit time of peering. If the aggregate charge $\mu(u) + \mu(v)$ exceeds the cost of peering $c(u, v)$ then it is to the combined benefit of u and v to peer. This model assumes that in this case they will share the costs in some fair manner and set up the peering link. Thus the revenue to the provider from the edge (u, v) is $x(u, v)(\mu(u) + \mu(v))$ if $\mu(u) + \mu(v) \leq c(u, v)$, and zero otherwise.

The revenue of the provider is then given by the product of its price and the total traffic between all customers that do not peer. For any subset of edges $E' \in E$, let

$$\nu(\mu; E') \equiv \sum_{\substack{(u,v) \in E' \\ \mu(u) + \mu(v) \leq c(u,v)}} x(u, v)(\mu(u) + \mu(v)) \quad (1)$$

giving the total revenue from $G = (V, E)$ as

$$\nu(\mu) \equiv \nu(\mu; E). \quad (2)$$

Remark:

1) The "price to peer", c, could equally well model the total cost to u and v of forgoing their connectivity.
2) That the costs of peering are primarily on-going costs, such as rental of co-location facilities, increased traffic
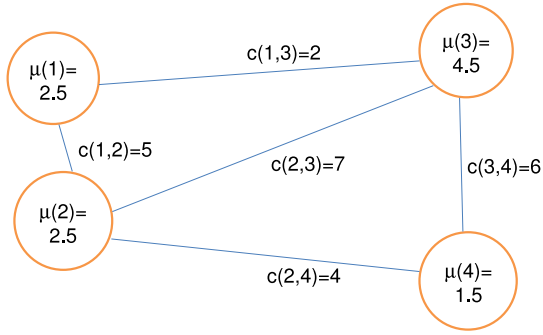
Fig. 1.   An example with four ISPs

within the peers' own networks, the depreciation cost of equipment used for peering, and the extra costs associated with having to track down faults rather than being able to apply pressure to the provider to ensure connectivity.

3) The assumption that ISPs peer if it is in their *combined* interest is an idealization, since in practice peering is usually only established if it is in the interests of both parties individually. The combinatorial nature of the problem remains in both cases.

4) In addition to gaining revenue, the provider incurs additional cost for traffic it carries. This is not included in the current model.

□

The problem we seek to solve is to maximize $\nu(\mu)$ for a graph G:

$$\max_{\mu:V\mapsto\overline{\mathbb{R}}^+} \nu(\mu) \qquad \text{(P1)}$$

with optimal value denoted $opt(G)$.

This notation is shown for a small example network in Figure 1. The circles represent ISPs, and are labeled with their prices, $\mu(v)$, while the edges are labeled with the peering costs. In this example, the provider obtains revenue from edges (1,2), (2,3), (2,4) and (3,4), but not from (1,3) since $\mu(1)+\mu(3) > c(1,3)$. There is negligible traffic between ISPs 1 and 4.

Problem (P1) is related to the problem of finding the maximum cut of a graph. A *cut* of a graph $G = (V, E)$ is a partition of the nodes $\{X, \overline{X}\}$ with $X \cap \overline{X} = \emptyset$, $X \cup \overline{X} = V$. The "size" of the cut of a graph with edge weights c is weighted sum of the "cut edges" with one vertex in each side of the partition, $\sum_{\{(u,v):u\in X, v\in\overline{X}\}} c(u,v)$. This paper uses two NP-complete problems related to maximum cuts. The "simple max-cut" [6], used in Section III, finds the cut of maximum size of an unweighted simple graph ($c(e) = 1$ for all $e \in E$). The "weighted max-cut", used in Section IV-C, finds the maximum cut of an arbitrary weighted graph.

Before ending this section, we now show that the problem can be solved by solving linear programs (LPs). Define a "revenue set" to be a set of edges $R \subseteq E$ from which we require that revenue be gained. Given a revenue set, finding the optimal μ is an LP, and hence in P.

$$\max_{R\subseteq E} M(R) \qquad \text{(P2)}$$

where $M(R)$ is the solution to the linear program

$$\max_{\mu:V\mapsto\overline{\mathbb{R}}^+} \sum_{(u,v)\in R} x(u,v)(\mu(u) + \mu(v))$$
$$s.t. \quad \mu(u) + \mu(v) \leq c(u,v)$$

Note that R is the set of edges from which we *require* revenue; for suboptimal R, the solution to (P2) may yield a solution μ under which revenue is also obtained by edges not in R.

This LP interpretation has two important applications. The first is that it allows a heuristic search of revenue sets, as will be described in Section IV-A. It can also be used as a post-processing stage after any approximate algorithm which finds a price function μ. Such price function induces a revenue set, R. Solving the corresponding LP yields a new price function μ no worse than the original, and often significantly better. The value of this approach is demonstrated in the numerical results of Section V.

It is clear that the number of LPs that we need to solve grows exponentially as a function of the number of possible links. This combinatoric structure hints that the problem could be NP-hard, which will be formally shown in the next section.

For the remainder of this paper, it will be assumed for simplicity that $x(e) = 1$ for all $e \in E$. The techniques can easily be generalized.

III. NP-HARDNESS

Before investigating approximate solutions, it is reassuring to know that there is no simple algorithm to find the exact solution. This is the case since the problem of assigning the optimal prices μ is NP-complete. This will now be shown by showing the NP-completeness of the corresponding decision problem "Is there an r such that the total revenue is at least r?"

Clearly this problem is in NP, since it is easy to compute the revenue from a given price allocation. To see that it is NP-hard, it will be shown that the "simple max-cut" problem [6] can be reduced in polynomial time to solving the pricing problem (P1).

If μ were restricted to integers, then the max-cut could be solved simply by assigning weight $c(e) = 1$ to each edge of G and solving the resulting problem (P1). An optimal (integer) solution consists of allocating each node a price $\mu(v) \in \{0, 1\}$, since no revenue can be obtained from edges adjoining any node with price $\mu(v) > 1$. The sets $X = \{v \in V : \mu(v) = 0\}$ and $\bar{X} = V \setminus X$ form the maximum cut, since revenue is obtained exactly from those edges (u, v) with $u \in X$ and $v \in \bar{X}$.

However, the optimal solution to the continuous form of the problem (P1) on G is the degenerate allocation $\mu(v) = 0.5$ for all $v \in V$, which cannot be used to solve the max-cut problem. The remainder of this section describes an augmentation of G which forces the optimal (continuous) prices to be either close to 0 or close to 1, allowing simple max-cut to be reduced to the continuous problem (P1).
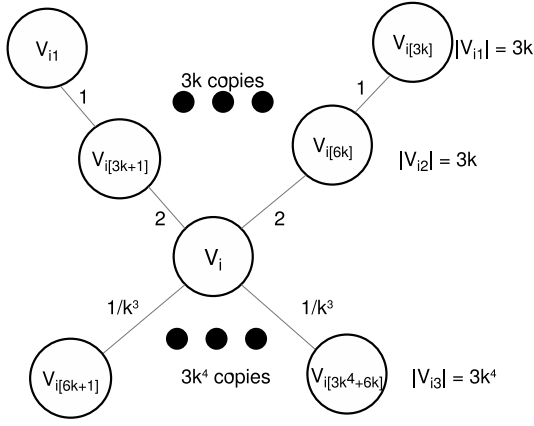
Fig. 2. Construction to force prices μ to lie in $[0, 1/k^3) \cup (1 - 1/k^3, 1]$.
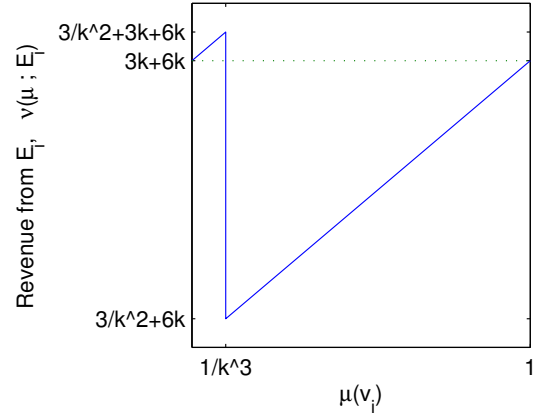

Fig. 3. Maximum revenue from auxiliary graph, E_i, as price $\mu(v_i)$ varies.

Consider an arbitrary simple undirected graph $G = (V, E)$, where $V = \{v_1, v_2, ..., v_{|V|}\}$. Any isolated node v, such that there is no $u \in V$ with $(u, v) \in E$, will not affect the max-cut of G. Thus, without loss of generality, we will assume G has no isolated nodes.

Let $k = 2|E| + 1$. Construct a graph G' from G as follows. For each v_i, define the auxiliary graph G_i as shown in Figure 2.

Formally, let $G_i = (V_i, E_i)$ be a graph, where

$$
\begin{aligned}
V_{i1} &= \{v_{i1}, v_{i2}, ..., v_{i[3k]}\} \\
V_{i2} &= \{v_{i[3k+1]}, v_{i[3k+2]}, ..., v_{i[6k]}\} \\
V_{i3} &= \{v_{i[6k+1]}, v_{i[6k+2]}, ..., v_{i[3k^4+6k]}\} \\
V_i &= \{v_i\} \cup V_{i1} \cup V_{i2} \cup V_{i3} \\
E_{i1} &= \{(v_{ij}, v_{i[j+3k]}) \mid v_{ij} \in V_{i1}, v_{i[j+3k]} \in V_{i2}\} \\
E_{i2} &= \{(v_i, v_{ij}) \mid v_{ij} \in V_{i2}\} \\
E_{i3} &= \{(v_i, v_{ij}) \mid v_{ij} \in V_{i3}\} \\
E_i &= E_{i1} \cup E_{i2} \cup E_{i3}
\end{aligned}
$$

Then $G' = (V', E')$ where

$$
V' = \bigcup_{i=1}^{|V|} V_i, \qquad E' = E \cup \bigcup_{i=1}^{|V|} E_i \tag{3}
$$

with $x(e) = 1$ for $e \in E'$ and weights $c : E' \mapsto \mathbb{R}^+$ such that

$$
c(e) = \begin{cases} 1 & e \in E \cup E_{i1} \\ 2 & e \in E_{i2} \\ 1/k^3 & e \in E_{i3} \end{cases} \tag{4}
$$

Before proving the main results, let us examine some properties of G'.

Lemma 1. *Consider each G_i. The maximum revenue we can get from E_i, $\max\{\nu(\mu; E_i) \mid \mu : V' \mapsto \overline{\mathbb{R}}^+\}$, depends on the value of $\mu(v_i)$, but not on the other nodes in V. Moreover, if $\mu(v_i) \le 1$, the revenues satisfy*

$$
\nu(\mu; E_{i1} \cup E_{i2}) \le 3k(2 + \mu(v_i)) \tag{5}
$$

$$
\nu(\mu; E_{i3}) \le 3k \tag{6}
$$

$$
\max(\nu(\mu; E_i)) = 3k(2 + \mu(v_i) + \mathbf{I}_{\mu(v_i) < 1/k^3})
$$

$$
\le 3k(3 + 1/k^3), \tag{7}
$$

where $\mathbf{I}_A = 1$ if A is true, and 0 otherwise.

The proof of this, and all other lemmas not immediately followed by their proofs, is in the appendix. A plot of $\nu(\mu; E_i)$ against $\mu(v_i)$ is shown in Figure 3.

The following three lemmas show that there exists an optimal solution μ to the problem (P1) with $\mu(v_i)$ very close to either 0 or 1.

From any price function, it is possible to construct another price function yielding at least as much revenue, but with no prices exceeding 1.

Lemma 2. *There is a $\mu : V' \mapsto \overline{\mathbb{R}}^+$ such that $\nu(\mu) = opt(G')$ and $\mu(v_i) \le 1$ for all $v_i \in V$.*

The proof uses Lemma 1 and the fact that the maximum revenue from each auxiliary edge in E_i can be achieved when $\mu(v_i) = 1$.

For μ of the form specified in Lemma 2, there are stricter constraints for the value of $\mu(v_i)$. For values of $\mu(v_i)$ near $1/2$, the solution would forfeit the revenue of up to $9k$ from the very large auxiliary graph G_i, and the revenue could be increased by reducing $\mu(v_i)$ to 0. This gives rise to

Lemma 3. *Consider μ of the form given in Lemma 2. For each $v_i \in V$, $\mu(v_i) \le 1/k^3$ or $\mu(v_i) \in [1 - 1/k^3, 1]$*

In particular, if $\mu(v_i) \le 1/k^3$, revenue is gained from E_{i3} while if $\mu(v_i) \ge 1 - 1/k^3$ then sufficient revenue is gained from $E_{i1} \cup E_{i2}$ to compensate the loss of that revenue. Thus $\mu(v_i)$ should be very close to either 0 or 1.

With these results, we can show that the simple max-cut problem can be reduced to the pricing problem (P1).

Lemma 4. *The max-cut of G has size at least $r \in \mathbb{Z}$ if and only if $opt(G') \ge 9|V|k + r$.*

Theorem 5. *The simple max-cut problem can be reduced in polynomial time to the pricing problem (P1).*

Proof: By Lemma 4, $opt(G') \ge 9|V|k + r$ if and only if the max-cut of G has size at least $r \in \mathbb{Z}$. To see that the reduction can be done in polynomial time, note that

$$
|V'| = |V| + (3k^4 + 6k) \tag{8}
$$

$$
|E'| = |E| + (3k^4 + 6k) \tag{9}
$$

are polynomials of $|V|$ and $|E|$, and to make G', it is only required to add $|V'|$ nodes and $|E'|$ edges. ∎

Corollary 6. *The problem (P1) is NP-complete*

Proof: It is clearly in NP, and the above reduction from simple max-cut shows NP-hardness. ∎

IV. APPROXIMATION ALGORITHMS

Since the problem is NP-complete, it is useful to investigate approximation algorithms. This section presents two algorithms with provable performance bounds, and a more heuristic algorithm which usually gets higher revenue but can sometimes perform arbitrarily badly.

When seeking a bound, an obvious candidate would be the sum of the edge weights, $\sum_{e \in E} c(e)$. However, this is hampered by the fact that for all $r \in \mathbb{R}^+$ however small, there is a G such that $opt(G) \leq r \sum_{e \in E} c(e)$. (This can be seen by considering complete graphs of size k with $c(v_j, v_{[(j+i) \bmod k]}) = 1/i$; $opt(G)$ grows as $O(k)$ while the sum grows as $\Theta(k \log(k))$.)

Instead, bounds are found in terms of a tighter upper bound $F(V)$, defined as follows. For any $v \in V$ and any $t \in \overline{\mathbb{R}}^+$, define $\lambda_{v,t} : V \mapsto \mathbb{R}$, by $\lambda_{v,t}(v) = t$ and $\lambda_{v,t}(u) = 0$ for all $u \neq v$. Then let $f(v)$ be the maximum revenue obtainable from ISP v, and $g(v)$ be the corresponding price charged to v. That is,

$$f(v) = \max\{\nu(\lambda_{v,t}) \mid t \in \overline{\mathbb{R}}^+\} \tag{10}$$

$$g(v) = \min\{t \in \overline{\mathbb{R}}^+ \mid \nu(\lambda_{v,t}) = f(v)\}. \tag{11}$$

Denote the edges from which v could get revenue under μ by

$$E_\mu(v) = \{(u,v) \in E \mid \mu(v) \leq c(u,v)\} \tag{12}$$

and note that, taking $\mu = g$,

$$f(v) = g(v) \, |E_g(v)|. \tag{13}$$

For any subset $U \subseteq V$, let

$$F(U) = \sum_{v \in U} f(v). \tag{14}$$

Lemma 7. *For any graph $G = (V, E)$,*

$$opt(G) \leq F(V). \tag{15}$$

Proof: Let μ be a function such that $\nu(\mu) = opt(G)$. For each $v \in V$, denote the revenue obtained from v by

$$f'(v) = \mu(v) \, |\{\, u \in V \mid \mu(u) + \mu(v) \leq c(u,v), \quad (u,v) \in E \,\}| \tag{16}$$

whence

$$\begin{aligned} opt(G) &= \sum_{v \in V} \sum_{\substack{(u,v) \in E \\ \mu(u)+\mu(v) \leq c(u,v)}} \mu(v) \\ &= \sum_{v \in V} f'(v) \leq \sum_{v \in V} f(v) \end{aligned}$$

using $\mu(u) \geq 0$, and the result follows by (14). ∎

For all graphs, $F(V)/4 \leq opt(G)$, allowing it to be used to prove constant-ratio bounds of Theorems 13 and 15.

The following lemma shows that $f(v)$ and $g(v)$ are easily computable.

Lemma 8. *For each $v \in V$,*

$$g(v) \in \{c(u,v) \mid (u,v) \in E\}. \tag{17}$$

Proof: Clearly, if the right hand side is non-zero,

$$g(v) \leq \max\{c(u,v) \mid (u,v) \in E\} \tag{18}$$

since otherwise $f(v) = 0$ by (13).

Suppose that (17) is false, and let

$$t = \min\{c(u,v) \mid (u,v) \in E, g(v) < c(u,v)\}.$$

Such a t always exists by (18), and the hypothesis that (17) is false. Moreover, with E_g defined by (12),

$$|\{(u,v) \in E \mid t \leq c(u,v)\}| = |E_g(v)|$$

because the existence of a (u,v) such that $g(v) < c(u,v) < t$ would contradict the minimality of t. Thus

$$\begin{aligned} \nu(\lambda(v,t)) &= t \, |\{(u,v) \in E \mid t \leq c(u,v)\}| \\ &> g(v) \, |E_g(v)| \\ &= f(v) \end{aligned}$$

This contradicts the maximality of $f(v)$. ∎

The results in Theorems 13 and 15 show that this problem is approximable up to a constant factor in polynomial time, and thus in complexity class APX [4]. This leave open whether or not it has a polynomial time approximation scheme (PTAS) [4], meaning that for any $\epsilon > 0$ there exists a polynomial time algorithm to find a solution at least $1 - \epsilon$ times the maximum. Note that max-cut on *dense* graphs is in class PTAS [3] and, since ISPs want to be able to connect to all other ISPs, this problem primarily concerns complete graphs.

A. Sequential by link

Since the optimal weights for a given revenue set can be found by solving a LP, an effective heuristic is simply to try a polynomial number of different revenue sets, and take the best result.[1] Two examples are as follows:

Add   Start with a full set of peering links ($R = \emptyset$), and greedily add to R the link which provides the greatest incremental increase in revenue. This algorithm is guaranteed to get revenue from the two edges with the highest cost, $c(e)$, although the optimal solution may not use these edges.

Relax   Start with an empty set of peering links ($R = E$), and greedily add peering links, which relaxes the constraint and forgoes the revenue.

The results of these algorithms can be unboundedly worse than the optimal, as in the case of the following lemma.

Lemma 9. *For a graph $G = (V, E)$, let $\mu_{Add} : V \mapsto \mathbb{R}^+ \cup \{0\}$ be the price function found by algorithm "Add". For all $r \in \mathbb{R}^+$ there exists a G such that*

$$\nu(\mu_{Add}) \leq r \, opt(G).$$

---

[1] The maximum of the linear program, $M(R)$, may be strictly smaller than the revenue $\nu(\mu)$ from the resulting price function. The results in Section V are for algorithms which maximize the increase in $M(R)$ at each step.

B. Sequential by node

An alternative sequential algorithm is to allocate prices sequentially to nodes rather than edges. Algorithm 1 is such an algorithm, which provably gets at least 1/8 of the maximum possible revenue. It incrementally constructs prices μ through a sequence of intermediates μ_i.

Algorithm 1 Sequential by node

Given a graph $G = (V, E)$, and peering costs $c : E \mapsto \mathbb{R}^+$:

1: $\mu_0(v) \leftarrow 0$ for all $v \in V$
2: Compute $f(v)$ and $g(v)$ by (10) and (11)
3: sort V to make $V = \{v_1, v_2, ..., v_{|V|}\}$, such that

$$g(v_1) \leq g(v_2) \leq ... \leq g(v_{|V|}) \qquad (19)$$

4: **for** $i \leftarrow 1$ to $|V|$ **do**
5:    **for** $j \leftarrow 1$ to $|V|$ **do**
6:

$$\mu_i'(v_j) \leftarrow \begin{cases} g(v_i)/2 & j = i \\ \mu_{i-1}(v_j) & j \neq i \end{cases} \qquad (20)$$

7:    **end for**
8:    compute $\nu(\mu_i')$ using (2)
9:    **if** $\nu(\mu_i') - \nu(\mu_{i-1}) \geq \frac{1}{4} f(v_i)$ **then**
10:      $\mu_i = \mu_i'$
11:    **else**
12:      $\mu_i = \mu_{i-1}$
13:    **end if**
14: **end for**

After Algorithm 1 terminates, V is partitioned into two sets A and B such that

$$A = \left\{ v \in V \mid \mu(v) = \frac{1}{2}g(v) \right\} \qquad (21)$$

$$B = \{ v \in V \mid \mu(v) = 0 \} \qquad (22)$$

Algorithm 1 can be summarized as: for each i, we compare μ_{i-1} and μ_i'; if $\nu(\mu_i')$ is sufficiently more than $\nu(\mu_{i-1})$, put v_i into A, making $\mu_i = \mu_i'$.

Since Step (9) ensures the revenue increases by $f(v_i)/4$ whenever $\mu_i \neq \mu_{i-1}$,

Lemma 10. *If $F(A) \geq F(V)/2$ then $\nu(\mu) \geq F(V)/8$.*

Proof:

$$\nu(\mu) = \nu(\mu_{|V|}) = \sum_{i=1}^{|V|} (\nu(\mu_i) - \nu(\mu_{i-1})) \qquad (23)$$

$$= \sum_{v_i \in A} (\nu(\mu_i) - \nu(\mu_{i-1})) \qquad (24)$$

$$\geq \frac{1}{4} \sum_{v_i \in A} f(v_i) \qquad (25)$$

The result follows by (14) and $F(A) \geq F(V)/2$. ∎

To find an analogous result when $F(B) \geq F(V)/2$ will require an additional lemma.

Since the only difference between μ_{i-1} and μ_i' is the value of v_i, the difference $\nu(\mu_i') - \nu(\mu_{i-1})$ depends only on revenue

from edges incident on v_i. Denote the set of edges from which μ_i' obtains revenue by

$$E_{iR} = \{e \in E \mid v_i \in e, \nu(\mu_i', \{e\}) > 0\} \qquad (26)$$

and denote the set of edges from which μ_{i-1} obtains revenue but μ_i' cannot (since $\mu_{i-1}(u) \leq c(e) < \mu_i'(u) + \mu_i'(v_i)$) by

$$E_{iL} = \{e \in E \mid v_i \in e, \nu(\mu_{i-1}; \{e\}) > \nu(\mu_i'; \{e\}) = 0\}. \qquad (27)$$

Note that edges not in $E_{iR} \cup E_{iL}$ do not affect $\nu(\mu_i') - \nu(\mu_{i-1})$, since they yield no revenue under either price function.

The total potential change in revenue at stage i is then

$$\nu(\mu_i') - \nu(\mu_{i-1}) = (\nu(\mu_i'; E_{iR}) - \nu(\mu_{i-1}; E_{iR})) - \nu(\mu_{i-1}; E_{iL}) \qquad (28)$$

Lemma 11. *In stage i of Algorithm 1, the additional revenue available is*

$$\nu(\mu_i'; E_{iR}) - \nu(\mu_{i-1}; E_{iR}) \geq \frac{1}{2} f(v_i) \qquad (29)$$

The potential loss of revenue is

$$\nu(\mu_{i-1}; E_{iL}) = \sum_{\substack{(v_i, v_j) \in E, v_j \in A, j < i, \\ g(v_j) \leq 2c(v_i, v_j) < g(v_i) + g(v_j)}} \frac{g(v_j)}{2} \qquad (30)$$

and, if $v_i \in B$, then the potential loss would have been

$$\nu(\mu_{i-1}; E_{iL}) \geq \frac{1}{4} f(v_i). \qquad (31)$$

Lemma 12. *If $F(B) \geq F(V)/2$ then $\nu(\mu) \geq F(V)/8$.*

Proof: Considering revenue only from edges adjacent to a node with $\mu(v_i) = 0$,

$$\nu(\mu) = \sum_{i=1}^{|V|} \sum_{\substack{(v_i, v_j) \in E, j < i \\ \mu(v_i) + \mu(v_j) \leq c(v_i, v_j)}} \mu(v_i) + \mu(v_j)$$

$$\geq \sum_{v_i \in B} \sum_{\substack{(v_i, v_j) \in E, j < i \\ \mu(v_j) \leq c(v_i, v_j)}} \mu(v_j)$$

Substituting explicit values from (21) and (22), and then limiting the sum to E_{iL}, the right hand side becomes

$$\sum_{\substack{v_i \in B}} \sum_{\substack{(v_i, v_j) \in E, v_j \in A, j < i \\ g(v_j) \leq 2c(v_i, v_j)}} \frac{g(v_j)}{2}$$

$$\geq \sum_{v_i \in B} \sum_{\substack{(v_i, v_j) \in E, v_j \in A, j < i, \\ g(v_j) \leq 2c(v_i, v_j) < g(v_i) + g(v_j)}} \frac{g(v_j)}{2}$$

$$= \sum_{v_i \in B} \nu(\mu_{i-1}; E_{iL})$$

by (30) of Lemma 11. Thus, by (31) of Lemma 11,

$$\nu(\mu) \geq \frac{1}{4} \sum_{v_i \in B} f(v_i)$$

The result follows by (14) and $F(B) \geq F(V)/2$. ∎

Combining Lemmas 10 and 12 gives the main result:

Theorem 13. *The total revenue from Algorithm 1 is bounded by*

$$\nu(\mu) \geq \frac{1}{8} opt(G). \tag{32}$$

Proof: Since $\{A, B\}$ is a partition of V, $F(A) + F(B) = F(V)$. Thus either $F(A) \geq F(V)/2$ or $F(B) \geq F(V)/2$. Applying either Lemma 10 or Lemma 12 gives

$$\nu(\mu) \geq \frac{1}{8} F(V) \tag{33}$$

and the result follows by Lemma 7. ∎

C. Approximate max-cut

A tighter bound can be obtained using an approximate weighted max-cut algorithm. First, consider how prices μ could be allocated if a suitable cut were known.

Given a cut $\{X, \overline{X}\}$ of G, which cuts edges $E' = \{(u, v) \in E \mid u \in X, v \in \overline{X}\}$, define a price function μ_X which sets price to 0 for nodes on one side of the cut, and $g(v)$ for nodes on the other, as follows:
If

$$\sum_{\substack{v \in X \\ g(v) \leq c(u,v)}} \sum_{(u,v) \in E'} g(v) \geq \sum_{\substack{v \in \overline{X} \\ g(v) \leq c(u,v)}} \sum_{(u,v) \in E'} g(v) \tag{34}$$

then

$$\mu_X(v) \leftarrow \begin{cases} g(v) & v \in X \\ 0 & v \in \overline{X} \end{cases} \tag{35}$$

otherwise

$$\mu_X(v) \leftarrow \begin{cases} 0 & v \in X \\ g(v) & v \in \overline{X} \end{cases} \tag{36}$$

Also, define $c' : E \mapsto \overline{\mathbb{R}}^+$ by

$$c'(e) = \sum_{v \in e,\, g(v) \leq c(e)} g(v), \tag{37}$$

which is either 0, $\min(g(u), g(v))$ or $g(u) + g(v)$.

Lemma 14. *For an arbitrary cut $\{X, \overline{X}\}$ with cutting edges $E' \subseteq E$, the revenue generated by μ_X is related to the c'-weight of the cut by*

$$\nu(\mu_X) \geq \frac{1}{2} \sum_{e \in E'} c'(e) \tag{38}$$

Proof: Without loss of generality, consider the case that more revenue can be gained from nodes in X than in $\overline{X}$, in the sense of (34). Then, considering revenue only from the cut edges in E',

$$\nu(\mu_X) \geq \sum_{\substack{(v \in X, u \in \overline{X}) \in E' \\ \mu_X(u) + \mu_X(v) \leq c(u,v)}} \mu(u) + \mu(v). \tag{39}$$

By (34), μ is given by (35), whence then summand in (39) becomes $g(v)$, giving

$$\nu(\mu_X) \geq \sum_{\substack{(v \in X, u \in \overline{X}) \in E' \\ g(v) \leq c(u,v)}} g(v)$$

$$= \sum_{v \in X} \sum_{\substack{(u,v) \in E' \\ g(v) \leq c(u,v)}} g(v)$$

$$\geq \frac{1}{2} \left[\sum_{v \in X} \sum_{\substack{(u,v) \in E' \\ c(u,v) \geq g(v)}} g(v) + \sum_{v \in \overline{X}} \sum_{\substack{(u,v) \in E' \\ c(u,v) \geq g(v)}} g(v) \right]$$

where the last step follows from (34), since the sum of two terms is at most twice the maximum term. Thus, since X and $\overline{X}$ partition V,

$$2\nu(\mu_X) \geq \sum_{v \in V} \sum_{\substack{(u,v) \in E' \\ c(u,v) \geq g(v)}} g(v)$$

$$= \sum_{e \in E'} \sum_{\substack{v \in e \\ c(u,v) \geq g(v)}} g(v)$$

and the result follows by the definition of c' in (37). ∎

Algorithm 2 finds a suitable cut, not by performing a max-cut on the graph with the original weights c, but instead with the weights c', which allows the performance to be bounded.

Algorithm 2 Max-cut-based algorithm

Given a graph $G = (V, E)$ and peering costs $c : E \mapsto \mathbb{R}^+$:
1: Compute $f(v)$ and $g(v)$ by (10) and (11)
2: Compute c' according to (37).
3: Use the approximate weighted max-cut algorithm of [10] to get a cut $X, \overline{X}$ of G, with edge weights c', of weight at least $\sum_{e \in E} c'(e)/2$.
4: Let $\mu \leftarrow \mu_X$ given by (34)–(36).

The main result of this section is

Theorem 15. *The μ generated by Algorithm 2 satisfies*

$$\nu(\mu) \geq \frac{1}{4} opt(G) \tag{40}$$

This is a straightforward consequence of the following two lemmas and (7).

Lemma 16. *The weight function c' defined by (37) satisfies*

$$F(V) = \sum_{v \in V} f(v) = \sum_{e \in E} c'(e) \tag{41}$$

Proof: By the definitions of f, g and ν and λ,

$$F(V) = \sum_{v \in V} \nu(\lambda_{v, g(v)})$$

$$= \sum_{v \in V} \sum_{\substack{(u,v) \in E \\ g(v) \leq c(u,v)}} g(v)$$

$$= \sum_{e \in E} \sum_{\substack{v \in e \\ g(v) \leq c(u,v)}} g(v)$$

and the result follows by the definition of c' in (37). ∎

Lemma 17. *The weight of the max-cut satisfies*

$$\sum_{e \in E'} c'(e) \geq \frac{1}{2} F(V) \qquad (42)$$

Proof: Since E' is generated using the algorithm of [10], it is guaranteed that

$$\sum_{e \in E'} c'(e) \geq \frac{1}{2} \sum_{e \in E} c'(e) \qquad (43)$$

The result then follows from Lemma 16. ∎

Note that Algorithm 2 uses a max-cut algorithm which guarantees 1/2 of *sum of edge weights* [10]. It is tempting to think that this result could be tightened by using the celebrated 0.878 approximation algorithm of [7]. However, this is unsuitable here since it guarantees 0.878 of the *true max-cut*, which may be less than 1/2 of the sum of the edge weights.

V. NUMERICAL RESULTS

In this section, numerical testing on random graphs are provided to study typical behaviors of the proposed algorithms.

The random graphs used here are all complete, since typically, each tier-2 ISP wants to communicate with every other tier-2 ISP. The results presented here are for costs $c(e)$ drawn independently from a uniform distribution with a maximum 100 times its minimum, $U[1, 100]$, and for exponentially distributed $c(e)$. Simulations were also performed for correlated costs, and the results are qualitatively similar, except that the max-cut algorithm performs around 20% worse.

The max-cut, sequential-by-node and "Add" version of sequential-by-edge are all run on ensembles of 100 random networks of between 3 and 20 nodes (3 to 190 edges). For comparison, the exact solution is evaluated by exhaustive search over the revenue sets, R, for those networks with up to seven nodes (21 edges).

Figure 4 shows the probability density function (PDF) of the "normalized revenue", measured as the ratio between the revenue obtained using the sequential-by-node and max-cut approximations and the optimal revenue. This is the reciprocal of the standard "approximation ratio". For clarity, the heuristic sequential-by-edge algorithm is not shown, since it finds the optimal prices with probability over 0.9 and simply appears as a spike on the graph. The max-cut and sequential-by-node algorithms achieve around 70% or 80% of the optimum in most runs, which is well above their provable bounds. Note that the ranking of the algorithms is unrelated to their theoretical bounds. The reason that the sequential-by-edge algorithm performs so well is primarily that it performs a more thorough search, examining $O(|V|^2)$ revenue sets, and thus $O(|V|^{5.5})$ time with an $O(|V|^{3.5})$ LP algorithm [9], in contrast to the $O(|V|^2 \log |V|)$ running time for the two approximation algorithms. For these particular examples, the mean running time of the programs are shown in Table I. The fixed overhead of around 48 ms dominates the time for the sequential-by-node and max-cut algorithms.
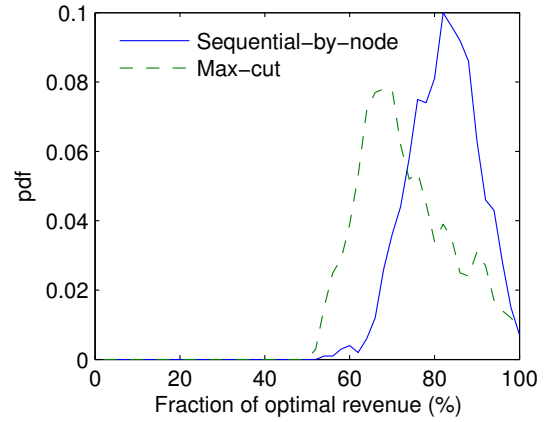

Fig. 4. PDF of the normalized revenue for networks with $|V| = 3, \ldots, 7$ and uniformly distributed peering costs.

TABLE I
RUNNING TIMES (ms) OF DIFFERENT ALGORITHMS

$|V|$	Exhaustive	Sequential-by-edge	Sequential-by-node	Max-cut
5	57	49	49	48
7	59500	56	52	59
20	–	62200	63	72

Clearly the computational complexity of the sequential-by-node and max-cut algorithms scales well. To investigate how well the accuracy scales, Figure 5 plots the performance of the algorithms against the size of the problem (number of nodes) for uniformly distributed peering costs. From Table I it is clear that finding the optimal solution for networks with $|V| > 7$ is prohibitive. For this reason, the performance of all algorithms is normalized to that of the sequential-by-edge algorithm (which yields an upper bound on the true "normalized revenue"). The optimal results are also plotted for $|V| \leq 7$, which shows that the sequential-by-edge algorithm is close to optimal, and hence a suitable reference for comparison when $|V|$ is large.

As mentioned previously, linear programming can be used to fine-tune the solutions obtained by the approximate algorithms. An approximate solution defines a particular revenue set, and it is simple to find the optimal solution corresponding to that revenue set. The curves marked by crosses in Figure 5 show the results of applying this improvement to each of the algorithms. The benefit is clearly significant.

Figure 6 shows analogous results for the exponentially distributed peering costs. Once again, sequential-by-edge performs very well. This time, however, max-cut consistently outperforms sequential-by-node as predicted by the approximation ratios.

VI. CONCLUSION

Motivated by the fact that ISPs can exchange traffic by setting up peering relations, we examine how much revenue a tier-1 ISP can obtain. The problem is formally shown to be NP-hard. We then proceed to provide heuristics and approximation algorithms. In particular, by leveraging an existing weighted max-cut algorithm, we are able to find a solution which is provably at least 1/4 of the optimum. Numerical results are
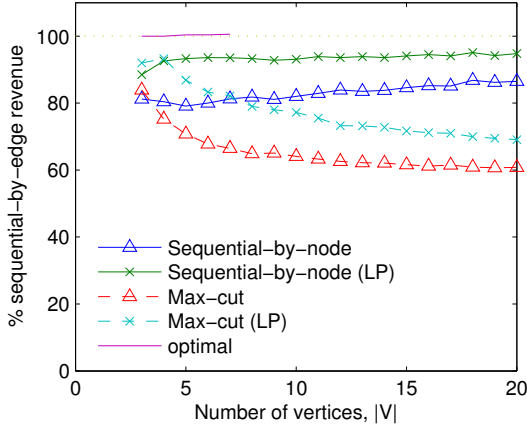
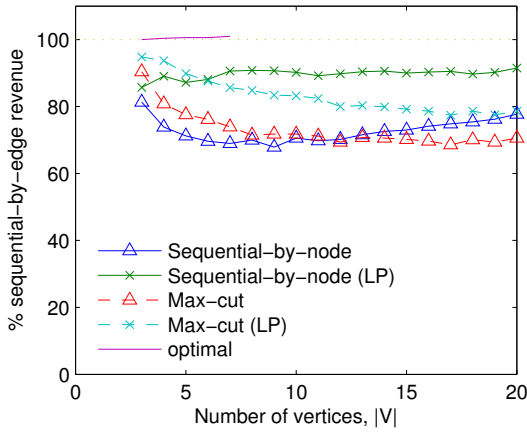Fig. 5. Performance of algorithms relative to sequential-by-edge, uniformly distributed peering costs.

Fig. 6. Performance of algorithms relative to sequential-by-edge, exponentially distributed peering costs.

also provided over randomly generated graphs to show the average behavior of the proposed algorithms.

APPENDIX

Lemma 1 Consider each G_i. The maximum revenue obtainable from E_i depends on the value of $\mu(v_i)$, but not on the other nodes in V. Moreover, if $\mu(v_i) \leq 1$, the revenues satisfy

$$\nu(\mu; E_{i1} \cup E_{i2}) \leq 3k(2 + \mu(v_i)) \qquad (5^*)$$
$$\nu(\mu; E_{i3}) \leq 3k \qquad (6^*)$$
$$\max(\nu(\mu; E_i)) = 3k(2 + \mu(v_i) + \mathbf{I}_{\mu(v_i)<1/k^3})$$
$$\leq 3k(3 + 1/k^3), \qquad (7^*)$$

where $\mathbf{I}_A = 1$ if A is true, and 0 otherwise.

Proof: Consider edges in E_{i1} and E_{i2}.

1) If $\mu(v_i) \leq 1$, we can get at most $3k(2+\mu(v_i))$ by setting

$$\mu(v_{ij}) = \begin{cases} 0 & v_{ij} \in V_{i1} \\ 1 & v_{ij} \in V_{i2} \end{cases} \qquad (44)$$

This establishes (5).

2) If $1 < \mu(v_i) \leq 2$, we can get the maximum $3k \times 3$ by setting

$$\mu(v_{ij}) = \begin{cases} \mu(v_i) - 1 & v_{ij} \in V_{i1} \\ 2 - \mu(v_i) & v_{ij} \in V_{i2} \end{cases} \qquad (45)$$

3) If $2 < \mu(v_i)$, we can get at most $3k$ by setting

$$\mu(v_{ij}) = \begin{cases} 1 & v_{ij} \in V_{i1} \\ 0 & v_{ij} \in V_{i2} \end{cases} \qquad (46)$$

Consider edges in E_{i3}.

1) If $\mu(v_i) \leq 1/k^3$, we can get $1/k^3 \times |E_{i3}| = 3k$ from edges in E_{i3} by setting

$$\mu(u) = \frac{1}{k^3} - \mu(v_i) \qquad u \in V_{i3} \qquad (47)$$

2) otherwise we can get 0 from edges in E_{i3}.

This establishes (6).

Summing revenues from E_{i1}, E_{i2}, and E_{i3} gives (7). ∎

Lemma 2 There is a $\mu : V' \mapsto \overline{\mathbb{R}}^+$ such that $\nu(\mu) = opt(G')$ and $\mu(v_i) \leq 1$ for all $v_i \in V$.

Proof: For any $\alpha : V' \mapsto \overline{\mathbb{R}}^+$ with $\alpha(v_i) > 1$ for some $v_i \in V$, let $\beta : V' \mapsto \overline{\mathbb{R}}^+$ such that

$$\beta(v_i) = 1 \qquad (48)$$
$$\beta(v_{ij}) = \begin{cases} 0 & v_{ij} \in V_{i1} \\ 1 & v_{ij} \in V_{i2} \\ 0 & v_{ij} \in V_{i3} \end{cases} \qquad (49)$$
$$\beta(u) = \alpha(u) \text{ if } u \notin V_i \qquad (50)$$

By Lemma 1, and since the only difference between α and β is for $v \in V_i$, it is enough to consider revenue from E_i and edges in E incident on v_i to compare $\nu(\alpha)$ and $\nu(\beta)$. Consider the revenue $\nu(\cdot; \cdot)$ under each pricing from different sets of links:

1) $\nu(\alpha, E_{i3}) = 0$ since $1/k^3 < 1 < \alpha(v_i)$.
2) $\nu(\beta; E_{i1} \cup E_{i2}) = 3k \times 3$, which is the maximum possible revenue from these edges.
3) From edges in E incident on v_i, α gets no revenue, $\nu(\alpha, \{(u, v_i) : v_i \in V'\}) = 0$, since $\alpha(v_i) > 1$.

Thus $\nu(\beta) \geq \nu(\alpha)$.

Repeating for all v_i such that $\alpha(v_i) > 1$ yields a μ such that $\nu(\mu) \geq \nu(\alpha)$ and $\mu(v_i) \leq 1$ for all $v_i \in V$. Therefore, any function from V' to $\overline{\mathbb{R}}^+$ can be modified to have ≤ 1 for all $v_i \in V$, while getting the equal or more revenue. ∎

Lemma 3 Consider μ of the form given in Lemma 2. For each $v_i \in V$, $\mu(v_i) \leq 1/k^3$ or $\mu(v_i) \in [1 - 1/k^3, 1]$.

Proof: First, it will be established that

$$\mu(v_i) \in [0, 1/k^3] \cup [2/3, 1]. \qquad (51)$$

Suppose instead there is some $v_i \in V$ such that $1/k^3 < \mu(v_i) < 2/3$. Let $\mu' : V' \mapsto \overline{\mathbb{R}}^+$ such that

$$\mu'(v_i) = 0 \qquad (52)$$

$$\mu'(v_{ij}) = \begin{cases} 0 & v_{ij} \in V_{i1} \\ 1 & v_{ij} \in V_{i2} \\ 1/k^3 & v_{ij} \in V_{i3} \end{cases} \qquad (53)$$

$$\mu'(u) = \mu(u) \text{ if } u \notin V_i \qquad (54)$$

Again, it is enough to consider revenue from E_i and E to compare $\nu(\mu)$ and $\nu(\mu')$.

First, we establish that the total revenue under μ is $\nu(\mu) < 9k$, as follows. Note that $\nu(\mu; E_{i3}) = 0$ since $\mu(v_i) > 1/k^3$, and $\nu(\mu; E_{i1} \cup E_{i2}) < 3k(2 + 2/3) = 8k$ since $\mu(v_i) < 2/3$. Thus, $\nu(\mu; E_i) < 8k$. Also, from edges in E, $\nu(\mu; E) \leq |E| \times 1 \leq k$, since $c(e) = 1$ for all $e \in E$.

However, from edges in E_i alone, μ' gets $\nu(\mu'; E_i) = 9k$, which is greater than $\nu(\mu) < 9k$. Thus $\nu(\mu') > \nu(\mu)$, which contradicts the hypothesis $\nu(\mu) = opt(G')$ and establishes (51).

It remains to show that there is no $v_i \in V$ such that $2/3 \leq \mu(v_i) < 1 - 1/k^3$. Suppose instead that there is such a v_i. Then let $\mu' : V' \mapsto \overline{\mathbb{R}}^+$ such that

$$\mu'(v_i) = 1 - 1/k^3 \qquad (55)$$

$$\mu'(v_{ij}) = \begin{cases} 0 & v_{ij} \in V_{i1} \\ 1 & v_{ij} \in V_{i2} \\ 0 & v_{ij} \in V_{i3} \end{cases} \qquad (56)$$

$$\mu'(u) = \mu(u) \text{ if } u \notin V_i \qquad (57)$$

Once again, it is enough to consider revenue from E_i and E to compare $\nu(\mu)$ and $\nu(\mu')$.

First, we will see that μ' derives more revenue from E_i. From E_{i3}, μ gets $\nu(\mu; E_{i3}) = 0$ since $2/3 > 1/k^3$. By (5), the revenue from E_{i1} and E_{i2} is

$$\begin{aligned} \nu(\mu; E_{i1} \cup E_{i2}) &\leq 3k(2 + \mu(v_i)) \\ &< 3k(2 + \mu'(v_i)) \\ &= \nu(\mu'; E_{i1} \cup E_{i2}). \end{aligned}$$

Thus $\nu(\mu'; E_i) > \nu(\mu; E_i)$.

For $(u, v_i) \in E$, noting that $\mu(u) = \mu'(u)$ leaves two cases:

1) $\mu(u) \leq 1/k^3$: $\mu(u) + \mu(v_i) \leq \mu'(u) + \mu'(v_i) \leq 1$, which means μ' gets more revenue than μ from (u, v_i).
2) $\mu(u) \geq 2/3$: μ cannot get any revenue from (u, v_i) since $\frac{2}{3} + \frac{2}{3} > 1$.

Thus, $\nu(\mu'; E) \geq \nu(\mu; E)$.

Thus $\nu(\mu') > \nu(\mu)$, which contradicts the hypothesis $\nu(\mu) = opt(G')$ and completes the proof. ∎

Lemma 4 The max-cut of G has size at least $r \in \mathbb{Z}$ if and only if $opt(G') \geq 9|V|k + r$.

Proof: **"If" part**

Let r' be the size of max-cut of G, and suppose $r' < r$. By Lemmas 2 and 3, we know that there is a $\mu : V' \mapsto \overline{\mathbb{R}}^+$ such that

$$\nu(\mu) = opt(G') \qquad (58)$$

$$\mu(v_i) \leq \frac{1}{k^3} \text{ or } \mu(v_i) \geq 1 - \frac{1}{k^3} \qquad \forall v_i \in V. \qquad (59)$$

Let A and B be the cut of G such that

$$A = \{v_i \mid \mu(v_i) \leq 1/k^3\} \qquad (60)$$

$$B = \{v_i \mid \mu(v_i) \geq 1 - 1/k^3\} \qquad (61)$$

For each $v_i \in V$, the maximal revenue we can get from E_i is $9k + 3/k^3$, when $\mu(v_i) = 1/k^3$, by the second part of Lemma 1. Thus we can get at most $|V|(9k + 3/k^3)$ from all E_is.

For each edge $(u, v) \in E$, there are three possible cases.

1) $(u, v) \in A \times A$: we can get $\mu(u) + \mu(w) \leq 2/k^3$
2) $(u, v) \in A \times B \cup B \times A$: we can get at most 1 since $c(u, v) = 1$.
3) $(u, v) \in B \times B$: we can get nothing from (u, v) since $\mu(u) + \mu(w) \geq 2 \times (1 - 1/k^3) > 1$

Since there are at most $|E|$ instances of case 1, and r' of case 2, we can get at most $2|E|/k^3 + r'$ from E. Thus

$$\begin{aligned} \nu(\mu) &\leq |V|(9k + 3/k^3) + 2|E|/k^3 + r' \\ &= 9|V|k + r' + (3|V| + 2|E|)/k^3. \end{aligned}$$

But $|V| \leq |E| + 1$, $k = 2|E| + 1$ and $|E| \geq 1$ imply $(3|V| + 2|E|)/k^3 < 1$, giving

$$\nu(\mu) < 9|V|k + r' + 1.$$

Since r' and r are both in $\mathbb{Z}$ and $r' < r$, this implies $opt(G') = \nu(\mu) < 9|V|k + r$, which is a contradiction. Therefore, $r' \geq r$, which completes the proof of the "if" part.

"Only if" part:

Let A, B be a cut of G such that the number of cut edges is at least r. Let $\mu : V' \mapsto \overline{\mathbb{R}}^+$ such that

$$\mu(v_i) = \begin{cases} 1 & v_i \in A \\ 0 & v_i \in B \end{cases} \qquad (62)$$

$$\mu(v_{ij}) = \begin{cases} 0 & v_{ij} \in V_{i1} \\ 1 & v_{ij} \in V_{i2} \\ 1/k^3 & v_{ij} \in V_{i3} \end{cases} \qquad (63)$$

Clearly $\nu(\mu) \geq 9|V|k + r$, since

1) For each $v_i \in A$, we can get $9k$ from the auxiliary edges E_i ($3k$ from E_{i1}, $6k$ from E_{i2} and 0 from E_{i3})
2) For each $v_i \in B$, we can get $9k$ from the auxiliary edges E_i ($3k$ from E_{i1}, $3k$ from E_{i2} and $3k$ from E_{i3})
3) From the original edges in E, we can get at least r (1 from each crossing edge)

Therefore, $opt(G') \geq \nu(\mu) \geq 9|V|k + r$, which completes the proof. ∎

Lemma 9 For a graph $G = (V, E)$, let $\mu_{\text{Add}} : V \mapsto \mathbb{R}^+ \cup \{0\}$ be the price function found by algorithm "Add". For all $r \in \mathbb{R}^+$ there exists a G such that

$$\nu(\mu_{\text{Add}}) \leq r \, opt(G).$$

Proof: Let k be such that

$$\sum_{i=1}^{k} \frac{1}{i} > \frac{12 - 4r + 1/k^2}{r} \qquad (64)$$

which exists since the right hand side is decreasing in k. Let $G = (V, E)$ such that

$$V = \{v_0, v_1, ..., v_k, v_{k+1}, v_{k+2}\} \qquad (65)$$

$$E_1 = \{v_0, v_i \mid 1 \leq i \leq k\} \qquad (66)$$

$$E_2 = \{v_i, v_{i+1} \mid 1 \leq i \leq k-1\} \qquad (67)$$

$$E = E_1 \cup E_2 \cup \{(v_0, v_{k+1}), (v_{k+1}, v_{k+2})\} \qquad (68)$$

and let $c : E \mapsto \mathbb{R}^+$ such that

$$c(e) = \begin{cases} 1/i & e = (v_0, v_i) \in E_1 \\ 1/k^3 & e \in E_2 \\ 10 & e = (v_0, v_{k+1}) \\ 2 & e = (v_{k+1}, v_{k+2}) \end{cases} \qquad (69)$$

Let $\mu' : V \mapsto \mathbb{R}^+ \cup \{0\}$ such that

$$\mu'(v_i) = \begin{cases} 1/i & 1 \leq i \leq k \\ 0 & i = 0 \text{ or } k+2 \\ 2 & i = k+1 \end{cases} \qquad (70)$$

Then, $\nu(\mu'; E_1) = \sum_{i=1}^{k} 1/i$, $\nu(\mu'; E_2) = 0$ and $\nu(\mu'; E \setminus (E_1 \cup E_2)) = 4$, giving

$$\nu(\mu') = 4 + \sum_{i=1}^{k} \frac{1}{i}. \qquad (71)$$

The sequential algorithm first adds (v_0, v_{k+1}) and (v_{k+1}, v_{k+2}), which give the most revenue 12. After that the algorithm adds all the edges in E_2 before adding any edge in E_1, since adding edges in E_1 would decrease revenue by adding a constraint to v_0 and losing revenue from (v_0, v_{k+1}). After adding all the edges in E_2, the algorithm adds edges in E_1, but we can get at most $1 + \frac{1}{k^2}$ from edges in E_1.

(No matter what $\mu(v_0)$ is, $1 = 2 \times \frac{1}{2} = ... = k \times \frac{1}{k}$, so edges in E_1 can get at most 1 from v_0. And since for $1 \leq i \leq k$, $v_i \leq 1/k^3$ by edges in E_2, so edges in E_1 can get at most $\frac{1}{k^3} \times k = 1/k^2$ from v_is.)

Therefore, the algorithm finds the local optimum when it gets revenue from only $E - E_1$, which gives $12 + 1/k^2$. Then

$$\nu(\mu_{\text{Add}}) = 12 + \frac{1}{k^2} < r\nu(\mu') \leq r\, opt(G) \qquad (72)$$

as required. ∎

Lemma 11 In stage i of Algorithm 1, the additional revenue available is

$$\nu(\mu'_i; E_{iR}) - \nu(\mu_{i-1}; E_{iR}) \geq \frac{1}{2} f(v_i) \qquad (29*)$$

The potential loss of revenue is

$$\nu(\mu_{i-1}; E_{iL}) = \sum_{\substack{(v_i, v_j) \in E, v_j \in A, j < i, \\ g(v_j) \leq 2c(v_i, v_j) < g(v_i) + g(v_j)}} \frac{g(v_j)}{2} \qquad (30*)$$

and, if $v_i \in B$, then the potential loss would have been

$$\nu(\mu_{i-1}; E_{iL}) \geq \frac{1}{4} f(v_i). \qquad (31*)$$

Proof: Since nodes $j > i$ have not yet been processed, $\mu'_i(v_j) = 0$. For $j < i$, (19) implies $\mu'_i(v_j) \leq \mu'_i(v_i)$, and

by (20), $\mu_i(v_i) \leq c(v_i, v_j)/2$ for $(v_i, v_j) \in E_g(v_i)$, with $E_g(v_i)$ defined in (12). Thus

$$\mu'_i(v_j) + \mu'_i(v_i) \leq c(e), \qquad e = (v_j, v_i) \in E_g(v_i).$$

Hence, using (13),

$$\begin{aligned} \nu(\mu'_i; E_g(v_i)) - \nu(\mu_{i-1}; E_g(v_i)) &= \mu'_i(v_i) |E_g(v_i)| \\ &= \frac{1}{2} g(v_i) |E_g(v_i)| = f(v)/2. \end{aligned}$$

From only edges from $E_g(v_i)$, μ'_i gets $f(v)/2$ more revenue than μ_{i-1}.

To prove (29), it suffices to show that this gain is not canceled by $e \notin E_{iR}$. This is the case since, for any $e = (u, v_i) \in E_{iR}$,

$$\nu(\mu'_i; \{e\}) > \nu(\mu_{i-1}; \{e\})$$

since

$$\mu_{i-1}(u) + \mu_{i-1}(v_i) < \mu'_i(u) + \mu'_i(v_i) \leq c(u, v_i)$$

whence $(\nu(\mu'_i; E_i \setminus E_g(v_i)) - \nu(\mu_{i-1}; E_i \setminus E_g(v_i))) \geq 0$.

To show (31), note that $v_i \in B$ only if $\nu(\mu'_i) - \nu(\mu_{i-1}) < f(v_i)/4$. Substituting this and (29) into (28) gives (31).

At stage $i - 1$, $\mu_{i-1}(v_i) = 0$, and hence by (1),

$$\begin{aligned} \nu(\mu_{i-1}; E_{iL}) &= \sum_{(v_i, v_j) \in E_{iL}} \mu_{i-1}(v_j) \\ &= \sum_{(v_i, v_j) \in E_{iL}, v_j \in A, j < i} \frac{g(v_j)}{2} \end{aligned}$$

since the non-zero terms are $g(v_j)/2$ by (21) and (22). Noting that E_{iL} consists of those links which give revenue under μ_{i-1} but not μ'_i gives (30). ∎

REFERENCES

[1] D. Acemoglu and A. Ozdaglar. Price Competition in Communication Networks. In *Proc. IEEE INFOCOM*, pages 1–12, 2006.

[2] E. Anshelevich, B. Shepherd and G. Wilfong. Strategic network formation through peering and service agreements. in *Proc. IEEE Foundations of Computer Science (FOCS)*, 2006, pages 77–86.

[3] S. Arora, D. Karger and M. Karpinski. Polynomial time approximation schemes for dense instances of NP-hard problems. In *Proc. ACM Symposium on the Theory of Computing* (Las Vegas, NV), pp. 284-293, May 29–June 1, 1995.

[4] G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, and M. Protasi. *Complexity and Approximation: Combinatorial optimization problems and their approximability properties*, Springer-Verlag, 1999.

[5] L. Gao and J. Rexford. Stable Internet routing without global coordination. *IEEE/ACM Transactions on Networking*, 9(6):681–692, December 2001.

[6] M. R. Garey, D. S. Johnson, L. Stockmeyer. Some simplified NP-complete problems. In *Proceedings of the sixth annual ACM symposium on Theory of computing*, 1974, pp 47-63.

[7] M. X. Goemans, and D. P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 42(6):1115–1145, Nov. 1995.

[8] R. Johari and J. Tsitsiklis. Routing and peering in a competitive Internet. In *Proc. IEEE Conf. Decision and Control*, 2004.

[9] N. Karmarkar. A new polynomial time algorithm for linear programming, *Combinatorica*, 4(4):373–395, 1984.

[10] S. Sahni and T. Gonzalez, P-Complete approximation problems, *Journal of the ACM*, 23(3):555–565, July 1976.

[11] S. Shakkottai and R. Srikant, Economics of network pricing with multiple ISPs, *IEEE Trans. Networking*, 14(6):1233-1244, Dec. 2006.

[12] B. Zhang, R. Liu, D. Massey, and L. Zhang. Collecting the internet AS-level topology. *ACM Computer Communication Review*, 35(1):53–61, Jan. 2005.