Improved Approximation Algorithms for Item Pricing with Bounded Degree and Valuation*

Ryoso HAMANE^{†a)}, Nonmember and Toshiya ITOH^{††b)}, Member

SUMMARY When a store sells items to customers, the store wishes to decide the prices of the items to maximize its profit. If the store sells the items with low (resp. high) prices, the customers buy more (resp. less) items, which provides less profit to the store. It would be hard for the store to decide the prices of items. Assume that a store has a set V of n items and there is a set C of m customers who wish to buy those items. The goal of the store is to decide the price of each item to maximize its profit. We refer to this maximization problem as an *item pricing* problem. We classify the item pricing problems according to how many items the store can sell or how the customers value the items. If the store can sell every item i with unlimited (resp. limited) amount, we refer to this as *unlimited* supply (resp. *limited* supply). We say that the item pricing problem is *single-minded* if each customer $j \in C$ wishes to buy a set $e_j \subseteq V$ of items and assigns valuation $w(e_j) \geq 0$. For the single-minded item pricing problems (in unlimited supply), Balcan and Blum regarded them as weighted k -hypergraphs and gave several approximation algorithms. In this paper, we focus on the (pseudo) degree of k -hypergraphs and the valuation ratio, i.e., the ratio between the smallest and the largest valuations. Then for the single-minded item pricing problems (in unlimited supply), we show improved approximation algorithms (for k -hypergraphs, general graphs, bipartite graphs, etc.) with respect to the maximum (pseudo) degree and the valuation ratio.

key words: item pricing, approximation algorithm, pseudodegree, valuation ratio

1. Introduction

1.1 Background

When a store sells items to customers, the store wishes to decide the prices of the items to maximize its profit. If the store sells the items with low (resp. high) prices, the customers buy more (resp. less) items, which provides less profit to the store. So it would be hard for the store to decide the prices of items. Assume that a store has a set V of n items and there is a set C of m customers who wish to buy those items. The goal of the store is to decide the price of each item to maximize its profit. We refer to this maximization problem as an *item pricing* problem. We classify the item

pricing problems according to how many items the store can sell and how the customers value items. If the store can sell every item i with unlimited (resp. limited) amount, we refer to this as *unlimited* supply (resp. *limited* supply). We say that the item pricing problem is *single-minded* [4] if each customer $j \in C$ wishes to buy a set $e_j \subseteq V$ of items and assigns valuation $w(e_j) \geq 0$. We say that the item pricing problem is *unit-demand* [4] if each customer $j \in C$ assigns a valuation $w_j(i)$ to each item $i \in V$ and buys one of the most beneficial items for $j \in C$.

For a single-minded and unlimited supply model, let us consider the case where the store decides the price of an item i by $p_i \geq 0$. Each customer $j \in C$ will buy the set $e_j \subseteq V$ of items at the price of $p(e_j) = \sum_{i \in e_j} p_i$ iff $w(e_j) \geq p(e_j)$. Let $\mathbf{p} = (p_1, p_2, \dots, p_n)$ be an item pricing vector. Then the total profit of the store is given by the sum of the profit from the customer $j \in C$ who bought its set e_j , i.e.,

$$\begin{aligned} \text{PROFIT}(\mathbf{p}) &= \sum_{j \in C: w(e_j) \geq p(e_j)} p(e_j) \\ &= \sum_{j \in C: w(e_j) \geq p(e_j)} \sum_{i \in e_j} p_i. \end{aligned}$$

Otherwise mentioned, the item pricing problems are assumed to be problems for the single-minded and unlimited supply model. It is obvious that an instance of the item pricing problem can be defined by a weighted hypergraph $G = (V, E)$. More precisely, we regard an item $i \in V$ as a vertex $i \in V$ of G and for each customer $j \in C$, we regard a set $e_j \subseteq V$ of items as a hyperedge $e_j \in E$ of G with weight $w(e_j)$.

Then by a weighted hypergraph, we can formulate the item pricing problem as follows:

Definition 1.1: Given a hypergraph $G = (V, E)$ with $|V| = n$ and a set $\{w(e)\}_{e \in E}$ of weight, compute a vertex pricing vector $\mathbf{p} = (p_1, p_2, \dots, p_n)$ that maximizes the total sum of the profit returned by $e \in E$, i.e.,

$$\text{PROFIT}(\mathbf{p}) = \sum_{e \in E: w(e) \geq p(e)} \sum_{i \in e} p_i,$$

where $p(e) = \sum_{i \in e} p_i$ is a sum of prices on $e \in E$.

We say that a hypergraph vertex pricing problem $G = (V, E)$ is a k -hypergraph vertex pricing problem if $|e| \leq k$ for each $e \in E$ and that a hypergraph vertex pricing problem $G = (V, E)$ is a graph vertex pricing problem if $|e| \leq 2$ for each $e \in E$. More specifically, we say that a graph vertex

Manuscript received April 2, 2007.

Manuscript revised June 8, 2007.

[†]The author is with the Department of Information Processing, Tokyo Institute of Technology, Tokyo, 152-8552 Japan.

^{††}The author is with the Global Scientific Information and Computing Center (GSIC), Tokyo Institute of Technology, Tokyo, 152-8552 Japan.

*Research supported in part by Japan Society of the Promotion of Science, Grant-in-Aid for Scientific Research on Priority Areas, 16092205, 2007.

a) E-mail: hamane@dac.gsics.titech.ac.jp

b) E-mail: titoh@dac.gsics.titech.ac.jp

DOI: 10.1093/ietisy/e91-d.2.187

pricing problem $G = (V, E)$ is a *bipartite graph vertex pricing* problem if G is a bipartite graph.

For the bipartite graph vertex pricing problem, Balcan and Blum [1, Theorem 1] presented a $1/2$ -approximation algorithm. Based on the approximation algorithm for the bipartite graph vertex pricing problem, Balcan and Blum derived a $1/4$ -approximation algorithm for the graph vertex pricing problem [1, Theorem 1] and also derived a $O(1/k)$ -approximation algorithm for k -hypergraph vertex pricing problem [1, Theorem 2], which improves the $O(1/k^2)$ -approximation algorithm by Briest and Krysta [2]. The approximation algorithms by Balcan and Blum [1] are derived by simple probabilistic arguments and in the analysis of those algorithms, the approximation ratio would be underestimated (thus those algorithms could be improved). In this paper, we will derive algorithms for those vertex pricing problems to achieve better approximation ratio.

1.2 New Notions

In this paper, we introduce new notions of “pseudodegree” and “valuation ratio.”

For any integer $k \geq 2$, let $G = (V, E)$ be a k -hypergraph vertex pricing problem with a set $\{w(e)\}_{e \in E}$ of weights (note that for $k = 2$, it is a graph vertex pricing problem). The pseudodegree of a vertex $v \in V$ is the number of distinct (hyper) edges in $e \in E$ incident to the vertex $v \in V$ and valuation ratio is the ratio between the smallest and the largest valuations.

The notion of pseudodegree captures the purchase patterns of customers. Let us consider the case where customers wish to buy a set of a computer, a monitor, and a keyboard at the store, and assume that (1) there are a popular set of computers $\{P_1, P_2\}$, a popular set of monitors $\{M_1, M_2\}$, and a popular set of keyboards $\{K_1, K_2\}$; (2) most of the customers wish to buy $\{P, M, K\} \in \{P_1, P_2\} \times \{M_1, M_2\} \times \{K_1, K_2\}$. This is a 3-hypergraph vertex pricing problem $G = (V, E)$. If 300 customers wish to buy the set of P_1, M_1, K_1 , 200 customers wish to buy the set of P_1, M_2, K_1 , and 100 customers wish to buy the set of P_1, M_2, K_2 , then the degree of P_1 is 600, but the pseudodegree of P_1 is 3. In this way, the notion of pseudodegree captures the purchase patterns of customers.

The notion of valuation ratio reflects the case of demanding customers. Assume that many customers wish to get tickets of the world series major league baseball games and/or the world cup football games. The customers know that those tickets are hard to get and low valuations on those tickets may lose the chance to get them. Thus the customers may value them as high as possible. In such a situation, the ratio between the cheapest and the highest valuations is far from zero.

1.3 Main Results

In this paper, we consider vertex pricing problems with bounded pseudodegree or with bounded valuation ratio, and

we show the approximation algorithms for trees, bipartite graphs, k -hypergraphs, etc.

Theorem 3.2: A tree vertex pricing problem $G = (V, E)$ admits an FPTAS.

Theorem 4.1: Let $G = (R \cup L, E)$ be a bipartite graph vertex pricing problem with the maximum pseudodegree d . For any constant $\varepsilon > 0$, there exists a deterministic polynomial time algorithm $\text{BIPARTITE}_\varepsilon^d$ that outputs a vertex pricing vector $\mathbf{p}$ such that

$$\text{PROFIT}(\mathbf{p}) \geq \frac{1 - \varepsilon}{2} \left\{ 1 + \frac{1}{d(d-1) + 1} \right\} \text{OPT}.$$

Theorem 4.2: Let $G = (V, E)$ be a graph vertex pricing problem with the maximum pseudodegree d . For any constant $\varepsilon > 0$, there exists a deterministic polynomial time algorithm $\text{GRAPH}_\varepsilon^d$ that outputs a vertex pricing vector $\mathbf{p}$ such that

$$\text{PROFIT}(\mathbf{p}) \geq \begin{cases} \left(1 + \frac{1}{d} \right) \frac{\text{OPT}}{4} & d \text{ is odd;} \\ \left(1 + \frac{1}{d+1} \right) \frac{\text{OPT}}{4} & d \text{ is even.} \end{cases}$$

Theorem 4.3: For any $k \geq 3$, let $G = (V, E)$ be a k -hypergraph vertex pricing problem with the maximum pseudodegree d . For any constant $\varepsilon > 0$, there exists a deterministic polynomial time algorithm $k\text{-HYPER}_\varepsilon^d$ that outputs a vertex pricing vector $\mathbf{p}$ such that

$$\text{PROFIT}(\mathbf{p}) \geq (1 - \varepsilon) \alpha_k \left\{ 1 + \frac{1/\alpha_k - 1}{d\{d(k-1) - 1\} + 1} \right\} \text{OPT},$$

where $\alpha_k = \frac{1}{k}(1 - 1/k)^{k-1}$ for any integer $k \geq 3$.

Theorem 5.2: Let $G = (V, E)$ be a graph vertex pricing problem with the valuation ratio r . There exists a deterministic polynomial time algorithm VALUATION_r^* that outputs a vertex pricing vector $\mathbf{p}$ such that

$$\text{PROFIT}(\mathbf{p}) \geq \begin{cases} \frac{2+r}{8} \text{OPT} & 0 \leq r < \frac{2}{3}; \\ \frac{1}{2(2-r)} \text{OPT} & \frac{2}{3} \leq r \leq 1. \end{cases}$$

2. Preliminaries

For any integer $k \geq 2$, let $G = (V, E)$ be a k -hypergraph vertex pricing problem with a set $\{w(e)\}_{e \in E}$ of weights (note that for $k = 2$, it is a graph vertex pricing problem). By definitions, it is immediate that E could be a multiset. For a *multi-set* E of k -hyperedges, define the *set* E' of k -hyperedges by removing all selfloops in E and regarding multiple hyperedges in E as a single hyperedge. Note that E' is not a multiset. Thus we have that $G' = (V, E')$ is a k -hypergraph with neither multiple hyperedges nor selfloops, i.e., $G' = (V, E')$ is a simple k -hypergraph. For a k -hypergraph $G = (V, E)$, we define the notion of *pseudodegree* $d_G(v)$ of a vertex $v \in V$

in $G = (V, E)$ to be the number of hyperedges $e \in E'$ incident to the vertex v in $G' = (V, E')$, i.e., for each $v \in V$, $d_G(v) = |\{e \in E' : v \in e\}|$. The notion of pseudodegree will play an crucial role in Sect. 4 to derive improved approximation algorithms.

For a simple graph $G = (V, E)$ with the maximum degree $d \geq 1$, we can easily color the vertices of G with a set $C = \{c_1, c_2, \dots, c_{d+1}\}$ of $d + 1$ colors (without assigning the same color to any of adjacent vertices). It is immediate to see that the same result holds for a multigraph (not a multi-hypergraph) w.r.t. pseudodegree.

Proposition 2.1: *For a multigraph $G = (V, E)$ with the maximum pseudodegree d , the vertices of V can be colored with $d + 1$ colors without assigning the same color to any adjacent vertices (excluding selfloops).*

For any integer $k \geq 2$, let $G = (V, E)$ be a k -hypergraph vertex pricing problem with a set $\{w(e)\}_{e \in E}$ of weights. For any $e \in E$, if $w(e) = 0$, then $e \in E$ provides no profit under any vertex pricing (even if optimal vertex pricing). So in the rest of this paper, we assume without loss of generality that $w(e) > 0$ for each $e \in E$. The notion of valuation ratio r is given by the ratio between the smallest and the largest valuations, i.e.,

$$r = \frac{\min_{e \in E} w(e)}{\max_{e \in E} w(e)}.$$

From the assumption that $w(e) > 0$ for each $e \in E$, it is immediate that $0 < r \leq 1$.

In the rest of this paper, we assume that the valuation $w(e)$ of an edge $e \in E$ is a positive integer, and we also assume that the price p_v assigned to a vertex $v \in V$ is a nonnegative integer.

3. FPTAS for a Class of Graphs

In this section, we show that a tree vertex pricing problem admits an FPTAS. For this, we first show that a star vertex pricing problem admits an FPTAS.

3.1 FPTAS for Stars

In this subsection, we show that a star vertex pricing problem admits an FPTAS, i.e., for a star vertex pricing problem $G = (V, E)$ with a set $\{w(e)\}_{e \in E}$ of weights and any constant $\varepsilon > 0$, there exists a deterministic algorithm STAR_ε that runs in polynomial in $|G|$ and $1/\varepsilon$ and outputs a vertex pricing vector $\mathbf{p}$ such that $\text{PROFIT}(\mathbf{p}) \geq (1 - \varepsilon) \cdot \text{OPT}$.

For a star vertex pricing problem $G = (V, E)$ with a set $\{w(e)\}_{e \in E}$ of weights, let $V = \{v_1, v_2, \dots, v_n\}$, where $v_1 \in V$ is the center of the star G , and let $m = |E|$ by counting multiple edges and selfloops. Let $h = \max_{e \in E} w(e)$ be the maximum (positive) valuation.

Lemma 3.1: *For a star vertex pricing problem $G = (V, E)$, there exists a deterministic $O(mh^2)$ time algorithm STAR that outputs an optimal vertex pricing vector $\mathbf{q}^*$, i.e., $\text{PROFIT}(\mathbf{q}^*) = \text{OPT}$.*

Proof: Let $d_1 \geq 1$ be the number of selfloops of the center $v_1 \in V$, and for each $2 \leq i \leq n$, let $d_i \geq 1$ be the number of edges incident to the vertex $v_i \in V$ (by counting its selfloops and multiple edges). Notice that $d_1 + d_2 + \dots + d_n = m$. When the price q_1 of v_1 is set to q for the star G , we use $\mathbf{q}_{q_1=q}^* = (q_1, q_2^*, \dots, q_n^*)$ to denote the vertex pricing vector that maximizes the total profit. For each $2 \leq i \leq n$, the price q_i^* can be computed independently by searching all possible (integer) prices $0 \leq q_i \leq h$, which takes $O(d_i h)$ time. Then for each $0 \leq q \leq h$, the vertex pricing vector $\mathbf{q}_{q_1=q}^*$ can be computed in $O(d_2 h) + O(d_3 h) + \dots + O(d_n h) = O(mh)$ time. Thus for the star $G = (V, E)$, the optimal vertex pricing vector $\mathbf{q}^*$ can be computed in $O(mh^2)$. ■

Notice that STAR is a pseudopolynomial time algorithm, because it runs in $O(mh^2)$ time. So for any constant $\varepsilon > 0$, we transform it to an algorithm that deterministically runs in polynomial in m and $1/\varepsilon$.

Theorem 3.1: *A star vertex pricing problem $G = (V, E)$ admits an FPTAS.*

Proof: For a star vertex pricing problem, the algorithm STAR_ε is given in Fig. 1.

Input: star $G = (V, E)$ with $\{w(e)\}_{e \in E}$ and $\varepsilon > 0$.

1. Let $\ell = \lfloor \frac{eh}{m} \rfloor \geq 1$ and for each $e \in E$, let $w'(e) = \lfloor w(e)/\ell \rfloor$.
2. Compute a vertex pricing vector $\hat{\mathbf{p}} = (\hat{p}_1, \hat{p}_2, \dots, \hat{p}_n)$ by running the algorithm STAR on the star $G = (V, E)$ with a set $\{w'(e)\}_{e \in E}$ of weights.
3. For each $1 \leq i \leq n$, let $p_i = \ell \cdot \hat{p}_i$ and output a vertex pricing vector $\mathbf{p} = (p_1, p_2, \dots, p_n)$.

Fig. 1 Algorithm: STAR_ε .

Let $G = (V, E)$ be a star. For a set $\{w(e)\}_{e \in E}$ of weights, we use $\text{PROFIT}(\mathbf{p})$ to denote the total profit by $\mathbf{p}$ and let $W(\mathbf{p}) \subseteq E$ be the set of customers who buy items under $\mathbf{p}$. For a set $\{w'(e)\}_{e \in E}$ of weights defined in step 1 of the algorithm STAR_ε , we use $\text{PROFIT}'(\mathbf{p})$ to denote the total profit by $\mathbf{p}$ and let $W'(\mathbf{p}) \subseteq E$ be the set of customers who buy items under $\mathbf{p}$.

For a vertex pricing vector $\mathbf{r} = (r_1, r_2, \dots, r_n)$, we define $\tilde{\mathbf{r}} = (\tilde{r}_1, \tilde{r}_2, \dots, \tilde{r}_n)$ to be $\tilde{r}_i = \lfloor r_i/\ell \rfloor$ for each $1 \leq i \leq n$. Then the following inequality holds:

$$\text{PROFIT}'(\tilde{\mathbf{r}}) \geq \frac{1}{\ell} \cdot \text{PROFIT}(\mathbf{r}) - m. \quad (1)$$

For each $e \in W(\mathbf{r})$, we have that $\sum_{i \in e} r_i \leq w(e)$. Since $\tilde{r}_i = \lfloor r_i/\ell \rfloor$ and $w'(e) = \lfloor w(e)/\ell \rfloor$,

$$\sum_{i \in e} \tilde{r}_i = \sum_{i \in e} \left\lfloor \frac{r_i}{\ell} \right\rfloor \leq \left\lfloor \frac{1}{\ell} \sum_{i \in e} r_i \right\rfloor \leq \left\lfloor \frac{w(e)}{\ell} \right\rfloor = w'(e),$$

which implies that $W(\mathbf{r}) \subseteq W'(\tilde{\mathbf{r}})$. From the fact that $\tilde{r}_i = \lfloor r_i/\ell \rfloor \geq (r_i/\ell) - 1$, it follows that

$$\begin{aligned} \text{PROFIT}'(\tilde{r}) &= \sum_{e \in W'(\tilde{r})} \sum_{i \in e} \tilde{r}_i \geq \sum_{e \in W(r)} \sum_{i \in e} \tilde{r}_i \\ &\geq \sum_{e \in W(r)} \sum_{i \in e} \left(\frac{r_i}{\ell} - 1 \right) = \frac{1}{\ell} \cdot \text{PROFIT}(r) - m. \end{aligned}$$

For a vertex pricing vector $\hat{s} = (\hat{s}_1, \hat{s}_2, \dots, \hat{s}_n)$, we define $s = (s_1, s_2, \dots, s_n)$ to be $s_i = \ell \cdot \hat{s}_i$ for each $1 \leq i \leq n$. Then the following inequality holds:

$$\text{PROFIT}(s) \geq \ell \cdot \text{PROFIT}'(\hat{s}). \quad (2)$$

For each $e \in W'(\hat{s})$, it is easy to show that $\sum_{i \in e} \hat{s}_i \leq w'(e)$. Since $s_i = \ell \cdot \hat{s}_i$, $w'(e) = \lfloor w(e)/\ell \rfloor$, we have that

$$\sum_{i \in e} s_i = \sum_{i \in e} \ell \cdot \hat{s}_i \leq \ell \cdot w'(e) = \ell \cdot \left\lfloor \frac{w(e)}{\ell} \right\rfloor \leq w(e),$$

which implies that $W'(\hat{s}) \subseteq W(s)$. From the fact that $s_i = \ell \cdot \hat{s}_i$, it follows that

$$\begin{aligned} \text{PROFIT}(s) &= \sum_{e \in W(s)} \sum_{i \in e} s_i = \sum_{e \in W(s)} \sum_{i \in e} \ell \cdot \hat{s}_i \\ &\geq \sum_{e \in W'(\hat{s})} \sum_{i \in e} \ell \cdot \hat{s}_i = \ell \cdot \text{PROFIT}'(\hat{s}). \end{aligned}$$

We are ready to show the theorem. For a star $G = (V, E)$ with a set $\{w(e)\}_{e \in E}$ of weights, let $q^* = (q_1^*, q_2^*, \dots, q_n^*)$ be the optimal vertex pricing vector and define the vertex pricing vector $\tilde{q}^* = (\tilde{q}_1^*, \tilde{q}_2^*, \dots, \tilde{q}_n^*)$ to be $\tilde{q}_i^* = \lfloor q_i^*/\ell \rfloor$. From Inequality (1), it follows that

$$\begin{aligned} \text{PROFIT}'(\tilde{q}^*) &\geq \frac{1}{\ell} \cdot \text{PROFIT}(q^*) - m \\ &= \frac{1}{\ell} \cdot \text{OPT} - m. \end{aligned} \quad (3)$$

From Lemma 3.1, it is obvious that the vertex pricing vector $\hat{p}$ in step 2 of STAR_ε is optimal for a star $G = (V, E)$ with the set $\{w'(e)\}_{e \in E}$ of weights, which implies that $\text{PROFIT}'(\hat{p}) \geq \text{PROFIT}'(\tilde{q}^*)$. For the vertex pricing vector p in step 3 of the algorithm STAR_ε , it follows from Inequality (2) that $\text{PROFIT}(p) \geq \ell \cdot \text{PROFIT}'(\hat{p})$. Then from Inequality (3), we have that

$$\begin{aligned} \text{PROFIT}(p) &\geq \ell \cdot \text{PROFIT}'(\hat{p}) \geq \ell \cdot \text{PROFIT}'(\tilde{q}^*) \\ &\geq \ell \left(\frac{1}{\ell} \cdot \text{OPT} - m \right) = \text{OPT} - \ell m \\ &\geq \text{OPT} - \varepsilon h \geq (1 - \varepsilon) \cdot \text{OPT}, \end{aligned}$$

where the fourth inequality follows from the definition of $\ell = \lfloor \frac{\varepsilon h}{m} \rfloor \leq \frac{\varepsilon h}{m}$ and the last inequality follows from the fact that $\text{OPT} \geq h$, which is guaranteed by the definition of h . Note that $\lfloor x \rfloor \geq x/2$ for any real $x \geq 1$. Then it is obvious that

$$\begin{aligned} h_* &= \max_{e \in E} w'(e) = \max_{e \in E} \left\lfloor \frac{w(e)}{\ell} \right\rfloor \leq \left\lfloor \frac{h}{\ell} \right\rfloor \\ &= \left\lfloor \frac{h}{\lfloor \frac{\varepsilon h}{m} \rfloor} \right\rfloor \leq \left\lfloor \frac{h}{\frac{\varepsilon h}{2m}} \right\rfloor = \left\lfloor \frac{2m}{\varepsilon} \right\rfloor \leq \frac{2m}{\varepsilon}. \end{aligned}$$

Thus from Lemma 3.1, it follows that for a star $G = (V, E)$ $\{w(e)\}_{e \in E}$ and any constant $\varepsilon > 0$, the algorithm STAR_ε runs in $O(mh_*^2) = O(m^3/\varepsilon^2)$. ■

Input: tree $G = (V, E)$ with $\{w(e)\}_{e \in E}$.

1. For each $0 \leq i \leq d$, compute $\text{PROFIT}_v^i(p)$ according to the following steps (a), (b), and (c): For each $v \in V_i$
 - (a) If $v \in V_i$ is a leaf, then do nothing.
 - (b) If each of the sons of $v \in V_i$ is a leaf, then for each $0 \leq q \leq h$, compute the maximum total profit $\text{PROFIT}_u^{i-1}(q)$ for the subtree $T(u) = (V(u), E(u))$ by pricing u with q .
 - (c) For each $0 \leq p \leq h$, compute

$$\begin{aligned} \text{PROFIT}_v^i(p) &= \sum_{u \in V_{i-1}(v)} \max_{0 \leq q \leq h} \{ \text{PROFIT}_u^{i-1}(q) \\ &\quad + \text{PROFIT}_{v,u}(p, q) \}. \end{aligned}$$

2. For the root $v_1 \in V$, compute the price p_1^* of v_1 satisfying that $\text{PROFIT}_{v_1}^d(p_1^*) = \max_{0 \leq p \leq h} \text{PROFIT}_{v_1}^d(p)$.
3. For each $u \in V_{i-1}(v)$, compute the price p_u^* of u such that

$$\begin{aligned} &\text{PROFIT}_u^{i-1}(p_u^*) + \text{PROFIT}_{v,u}(p_v^*, p_u^*) \\ &= \max_{0 \leq q \leq h} \{ \text{PROFIT}_u^{i-1}(q) + \text{PROFIT}_{v,u}(p_v^*, q) \}, \end{aligned}$$

and output a vertex pricing vector $p^* = (p_1^*, p_2^*, \dots, p_n^*)$.

Fig. 2 Algorithm: TREE.

3.2 FPTAS for Trees

In this subsection, we generalize Theorem 3.1 to the tree vertex pricing problems. Let $G = (V, E)$ be a tree with a set $\{w(e)\}_{e \in E}$ of weights. Let $V = \{v_1, v_2, \dots, v_n\}$ and $|E| = m$. Let $v_1 \in V$ be the root of the tree G (if the root of the tree G is not given). For an internal vertex $v \in V$, we use $T(v) = (V(v), E(v))$ to denote the subtree of $G = (V, E)$ rooted at v . For each $0 \leq i \leq d$, let $V_i \subseteq V$ be a set of vertices of height i , i.e., $V_d = \{v_1\}$ is the root of G . Note that for each $v \in V$, if each of the sons of v is a leaf, then a subtree $T(v) = (V(v), E(v))$ is a star. For each $v \in V_i$, let $V_{i-1}(v)$ be a set of sons of v , and for each $u \in V_{i-1}(v)$, let $\text{PROFIT}_{v,u}(p, q)$ be the profit returned from the self-loops of v and the edges between v and u by pricing v and u with p and q , respectively. Let $h = \max_{e \in E} w(e)$.

Lemma 3.2: For a tree vertex pricing problem $G = (V, E)$, there exists a deterministic $O(mh^2)$ time algorithm TREE that outputs an optimal vertex pricing vector q^* , i.e., $\text{PROFIT}(q^*) = \text{OPT}$.

Proof: The algorithm TREE is given in Fig. 2, which is based on dynamic programming. It is obvious that for the tree vertex pricing problem $G = (V, E)$, the algorithm TREE outputs an optimal vertex pricing vector p^* , i.e., $\text{PROFIT}(p^*) = \text{OPT}$. In a way similar to the proof of Lemma 3.1, we can show that the deterministic algorithm TREE runs in $O(mh^2)$. ■

Theorem 3.2: A tree vertex pricing problem $G = (V, E)$ admits an FPTAS.

Proof: In a way similar to the proof of Theorem 3.1, we

Input: tree $G = (V, E)$ with $\{w(e)\}_{e \in E}$ and $\varepsilon > 0$.

1. Let $\ell = \lfloor \frac{gh}{2m} \rfloor \geq 1$ and for each $e \in E$, let $w'(e) = \lfloor w(e)/\ell \rfloor$.
2. Compute a vertex pricing vector $\hat{p} = (\hat{p}_1, \hat{p}_2, \dots, \hat{p}_n)$ by running the algorithm TREE on the tree $G = (V, E)$ with a set $\{w'(e)\}_{e \in E}$ of weights.
3. For each $1 \leq i \leq n$, let $p_i = \ell \cdot \hat{p}_i$ and output a vertex pricing vector $p = (p_1, p_2, \dots, p_n)$.

Fig. 3 Algorithm: TREE_ε.

can show that for the tree $G = (V, E)$ with a set $\{w(e)\}_{e \in E}$ of weights and any constant $\varepsilon > 0$, the algorithm TREE_ε (see Fig. 3) deterministically runs in $O(m^3/\varepsilon^2)$ and outputs a vertex pricing vector p such that $\text{PROFIT}(p) \geq (1 - \varepsilon) \cdot \text{OPT}$. ■

4. Bounded Pseudodegree Graphs

4.1 Bipartite Graphs with Bounded Pseudodegree

For bipartite vertex pricing problems, Balcan and Blum [1, Theorem 1] showed a $1/2$ -approximation algorithm ALG_{BB}^{Bpt} as shown in Fig. 4. In this subsection, we consider a bipartite vertex pricing problem with the maximum pseudodegree d .

Input: bipartite graph $G = (R \cup L, E)$ with $\{w(e)\}_{e \in E}$.

1. Set the prices of all vertices in L to 0 and compute a vertex pricing vector p_R that maximizes the total profit. Let $R_{\max} = \text{PROFIT}(0, p_R)$.
2. Set the prices of all vertices in R to 0 and compute a vertex pricing vector p_L that maximizes the total profit. Let $L_{\max} = \text{PROFIT}(p_L, 0)$.
3. Output a vertex pricing vector p such that

$$p = \begin{cases} (p_L, 0) & L_{\max} \geq R_{\max}; \\ (0, p_R) & L_{\max} < R_{\max}. \end{cases}$$

Fig. 4 Algorithm: ALG_{BB}^{Bpt}.

For a bipartite graph $G = (L \cup R, E)$, we define a graph $H = (R, F)$ as described in Fig. 5.

Input: bipartite graph $G = (R \cup L, E)$.

1. For each vertex $u \in R$, let $N_L(u) = \{v \in L : (v, u) \in E\}$ be a neighbor of u .
2. For each pair of distinct vertices $s, t \in R$, $(s, t) \in F$ iff $N_L(s) \cap N_L(t) \neq \emptyset$.

Fig. 5 Definition of a Graph $H = (R, F)$.

Lemma 4.1: For a bipartite graph $G = (L \cup R, E)$ with the maximum pseudodegree d , let $\Delta(H)$ be the maximum degree of a graph $H = (R, F)$ as defined in Fig. 5. Then $\Delta(H) \leq d(d - 1)$.

Proof: Let $G = (L \cup R, E)$ be a bipartite graph with the maximum pseudodegree d . For each $u \in R$ and each $v \in L$, let $N_L(u) = \{v \in L : (v, u) \in E\}$ and $N_R(v) = \{u \in R : (v, u) \in E\}$ be neighbors of u and v , respectively. Since the maximum pseudodegree of the bipartite graph $G = (L \cup R, E)$ is d , we have that $|N_L(u)| \leq d$ for each $u \in R$ and $|N_R(v)| \leq d$ for each $v \in L$. Let $H = (R, F)$ be a graph as defined in Fig. 5. For each $s \in R$, let $d_H(s)$ be the degree of s and let $N(s) = \{t \in R : (s, t) \in F\}$ be neighbors of s in H . From the definition of the graph H , it follows that for each $s \in R$,

$$N(s) = \bigcup_{v \in N_L(s)} \{N_R(v) - \{s\}\}. \quad (4)$$

From the definitions of $N_L(\cdot)$ and $N_R(\cdot)$, it is obvious to see that for each $s \in R$, if $v \in N_L(s)$, then $s \in N_R(v)$. Thus it follows from Eq. (4) that for each $s \in R$,

$$\begin{aligned} d_H(s) &= |N(s)| = \left| \bigcup_{v \in N_L(s)} \{N_R(v) - \{s\}\} \right| \\ &\leq \sum_{v \in N_L(s)} |N_R(v) - \{s\}| \\ &\leq \sum_{v \in N_L(s)} (d - 1) = |N_L(s)| \cdot (d - 1) \leq d(d - 1), \end{aligned}$$

and for the maximum degree $\Delta(H)$ of the graph H , this implies that $\Delta(H) \leq d(d - 1)$. ■

For the bipartite graph $G = (L \cup R, E)$ with a set $\{w(e)\}_{e \in E}$ of weights, let p^* be the optimal vertex pricing vector, i.e., $\text{PROFIT}(p^*) = \text{OPT}$. For the optimal vertex pricing vector p^* of the bipartite graph G , we use $\text{PROFIT}_L(p^*)$ and $\text{PROFIT}_R(p^*)$ to denote the parts of the total profit $\text{PROFIT}(p^*)$ contributed from the L -side and from the R -side, respectively, i.e.,

$$\text{PROFIT}(p^*) = \text{PROFIT}_L(p^*) + \text{PROFIT}_R(p^*).$$

For the subsequent discussions below, assume without loss of generality that $\text{PROFIT}_L(p^*) \geq \text{PROFIT}_R(p^*)$, which implies that $\text{PROFIT}_L(p^*) \geq \text{OPT}/2$ (for the case where $\text{PROFIT}_L(p^*) < \text{PROFIT}_R(p^*)$, we regard the bipartite graph $G = (L \cup R, E)$ as a bipartite graph $G' = (R \cup L, E')$ by interchanging R and L , and apply Lemma 4.1 and Theorem 4.1).

Theorem 4.1: Let $G = (R \cup L, E)$ be a bipartite graph vertex pricing problem with the maximum pseudodegree d . For any constant $\varepsilon > 0$, there exists a deterministic polynomial time algorithm BIPARTITE_ε^d that outputs a vertex pricing vector p such that

$$\text{PROFIT}(p) \geq \frac{1 - \varepsilon}{2} \cdot \left\{ 1 + \frac{1}{d(d - 1) + 1} \right\} \cdot \text{OPT}.$$

Proof: We show the algorithm BIPARTITE_ε^d in Fig. 6.

From the definition of the graph $H = (R, F)$, we have that for any pair of distinct $s, t \in R$, $(s, t) \in F$ iff $N_L(s) \cap N_L(t) \neq \emptyset$. Notice that for each $1 \leq i \leq B + 1$, every $u \in R_i$ is colored with the same $c_i \in C$ in step 2 of the algorithm

Input: bipartite graph $G = (L \cup R, E)$ with the maximum pseudodegree d .

1. Define a graph $H = (R, F)$ and let $B = \Delta(H)$ be the maximum degree of the graph H .
2. Color the vertices of the graph $H = (R, F)$ with the set $C = \{c_1, c_2, \dots, c_{B+1}\}$ of $B+1$ colors and partition R into $R_1, R_2, \dots, R_{B+1}$ according to $c_i \in C$.
3. For each $1 \leq i \leq B+1$, let $\tilde{R}_i = R - R_i$ and $\tilde{L}_i = L - N_L(R_i)$, where $N_L(R_i) = \bigcup_{u \in R_i} N_L(u)$, and define an induced bipartite graph $\tilde{G}_i = (\tilde{L}_i \cup \tilde{R}_i, \tilde{E}_i)$.
4. For each $1 \leq i \leq B+1$, set the prices of all vertices of $\tilde{R}_i$ to 0 and compute a vertex pricing vector $\tilde{p}_i$ that maximizes the total profit of the bipartite graph $\tilde{G}_i = (\tilde{L}_i \cup \tilde{R}_i, \tilde{E}_i)$.
5. For each $1 \leq i \leq B+1$ and each $u \in R_i$, define an induced star $S_i^u = (\{u\} \cup N_L(u), E_i^u)$ and compute a vertex pricing vector p_i^u by running the algorithm STAR_ε on S_i^u .
6. For each $1 \leq i \leq B+1$, let $p_i = \{\tilde{p}_i\} \cup \{p_i^u\}_{u \in R_i}$ and output the vertex pricing vector p such that $\text{PROFIT}(p) = \max_{1 \leq i \leq B+1} \text{PROFIT}(p_i)$.

Fig. 6 Algorithm: $\text{BIPARTITE}_\varepsilon^d$.

$\text{BIPARTITE}_\varepsilon^d$, which implies that $(s, t) \notin F$ for any pair of distinct $s, t \in R_i$. So for each $1 \leq i \leq B+1$ and any pair of distinct $s, t \in R_i$, we have that $N_L(s) \cap N_L(t) = \emptyset$. Thus for each $1 \leq i \leq B+1$, it follows that the set $\{S_i^u\}_{u \in R_i}$ of stars are disjoint.

For the bipartite graph $G = (L \cup R, E)$ with a set $\{w(e)\}_{e \in E}$ of weights, let p^* be the optimal vertex pricing vector, i.e., $\text{PROFIT}(p^*) = \text{OPT}$. For each $1 \leq i \leq B+1$, we use $\text{PROFIT}_{R_i}(p^*)$ to denote the total profit returned from the vertices in R_i . So for any constant $\varepsilon > 0$, it follows from Theorem 3.1 that

$$\text{PROFIT}(p_i) \geq (1 - \varepsilon) \cdot \{\text{PROFIT}_L(p^*) + \text{PROFIT}_{R_i}(p^*)\}$$

for each $1 \leq i \leq B+1$. For the vertex pricing vector p in step 6 of the algorithm $\text{BIPARTITE}_\varepsilon^d$, we have that

$$\begin{aligned} \text{PROFIT}(p) &\geq \frac{1}{B+1} \sum_{i=1}^{B+1} \text{PROFIT}(p_i) \\ &\geq \frac{1-\varepsilon}{B+1} \sum_{i=1}^{B+1} \{\text{PROFIT}_L(p^*) + \text{PROFIT}_{R_i}(p^*)\} \\ &= (1-\varepsilon) \left\{ \text{PROFIT}_L(p^*) + \frac{\text{PROFIT}_{R_i}(p^*)}{B+1} \right\} \\ &= (1-\varepsilon) \left\{ \text{PROFIT}_L(p^*) + \frac{\text{OPT} - \text{PROFIT}_L(p^*)}{B+1} \right\} \\ &= (1-\varepsilon) \left\{ \frac{1}{B+1} \text{OPT} + \frac{B}{B+1} \text{PROFIT}_L(p^*) \right\} \\ &\geq (1-\varepsilon) \left\{ \frac{1}{B+1} \text{OPT} + \frac{1}{2} \frac{B}{B+1} \text{OPT} \right\} \\ &= \frac{1-\varepsilon}{2} \left(1 + \frac{1}{B+1} \right) \text{OPT} \\ &\geq \frac{1-\varepsilon}{2} \left\{ 1 + \frac{1}{d(d-1)+1} \right\} \text{OPT}. \end{aligned}$$

Notice that the first equality follows from the fact that $\text{PROFIT}_L(p^*) + \text{PROFIT}_R(p^*) = \text{OPT}$, the third inequality follows from the assumption that $\text{PROFIT}_L(p^*) \geq \text{OPT}/2$, and the last inequality follows from the result of Lemma 4.1, i.e., $B = \Delta(H) \leq d(d-1)$. ■

4.2 General Graphs with Bounded Pseudodegree

For graph vertex pricing problems, Balcan and Blum [1, Theorem 1] presented a $1/4$ -approximation algorithm $\text{ALG}_{\text{BB}}^{\text{Graph}}$ as in Fig. 7. The algorithm $\text{ALG}_{\text{BB}}^{\text{Graph}}$ is a randomized algorithm, but it can be derandomized by using pairwise independent 0/1-random variables with a small sample space [6].

Input: graph $G = (V, E)$ with $\{w(e)\}_{e \in E}$.

1. Partition V into L and R uniformly at random and remove all the edges $e = (v, u) \in E$ such that $v, u \in L$ or $v, u \in R$. Let $\tilde{E} \subseteq E$ be the remaining edges.
2. For the bipartite graph $\tilde{G} = (L \cup R, \tilde{E})$, set the prices of all vertices of L to 0 and compute p_R that maximizes the total profit. Let $R_{\max} = \text{PROFIT}(\mathbf{0}, p_R)$.
3. For the bipartite graph $\tilde{G} = (L \cup R, \tilde{E})$, set the prices of all vertices of R to 0 and compute p_L that maximizes the total profit. Let $L_{\max} = \text{PROFIT}(p_L, \mathbf{0})$.
4. Output a vertex pricing vector p such that

$$p = \begin{cases} (p_L, \mathbf{0}) & L_{\max} \geq R_{\max}; \\ (\mathbf{0}, p_R) & L_{\max} < R_{\max}. \end{cases}$$

Fig. 7 Algorithm: $\text{ALG}_{\text{BB}}^{\text{Graph}}$.

Theorem 4.2: For the graph vertex pricing problem with the maximum pseudodegree d , there exists a deterministic polynomial time algorithm GRAPH^d that outputs a vertex pricing vector p such that

$$\text{PROFIT}(p) \geq \begin{cases} \left(1 + \frac{1}{d}\right) \frac{\text{OPT}}{4} & d \text{ is odd;} \\ \left(1 + \frac{1}{d+1}\right) \frac{\text{OPT}}{4} & d \text{ is even.} \end{cases}$$

Proof: For the graph $G = (V, E)$ with the maximum pseudodegree d , it follows from Proposition 2.1 that the vertices in V can be colored by the set C of $d+1$ colors, where $C = \{c_1, c_2, \dots, c_{d+1}\}$. Then we partition V into $V_1, V_2, \dots, V_{d+1}$ according to $c_i \in C$ and define bipartite graphs $G_i = (L_i \cup R_i, E_i)$ as follows:

(d: odd) Let $f = (d+1)/2$. Let L be a union of f sets among $\{V_i\}_{i=1}^{d+1}$ and $R = V - L$. It is easy to see that there exist N_o such partitions $(L_1 : R_1), (L_2 : R_2), \dots, (L_{N_o} : R_{N_o})$, where

$$N_o = \binom{d+1}{f} = \binom{d+1}{(d+1)/2}. \quad (5)$$

For each $1 \leq i \leq N_o$, remove all the edges $e = (v, u) \in$

E such that $v, u \in L_i$ or $v, u \in R_i$ and let $E_i \subseteq E$ be the remaining edges, which defines a bipartite graph $G_i = (L_i \cup R_i, E_i)$.

(**d: even**) Let $f_L = d/2$ and $f_R = d/2 + 1$. Let L be a union of f_L sets among $\{V_i\}_{i=1}^{d+1}$ and $R = V - L$. There exist N_e such partitions $(L_1 : R_1), (L_2 : R_2), \dots, (L_{N_e} : R_{N_e})$, where

$$N_e = \binom{d+1}{f_L} = \binom{d+1}{d/2}. \quad (6)$$

For each $1 \leq i \leq N_e$, remove all the edges $e = (v, u) \in E$ such that $v, u \in L_i$ or $v, u \in R_i$ and let $E_i \subseteq E$ be the remaining edges, which defines a bipartite graph $G_i = (L_i \cup R_i, E_i)$.

Input: graph $G = (V, E)$ with the maximum pseudodegree d .

1. For each bipartite graph $G_i = (L_i \cup R_i, E_i)$, compute a vertex pricing vector $\mathbf{p}_i$ by running the algorithm $\text{ALG}_{\text{BB}}^{\text{Graph}}$ given in Fig. 7.
2. Output a vertex pricing vector $\mathbf{p}$ such that

$$\text{PROFIT}(\mathbf{p}) = \begin{cases} \max_{1 \leq i \leq N_o} \text{PROFIT}(\mathbf{p}_i) & d \text{ is odd;} \\ \max_{1 \leq i \leq N_e} \text{PROFIT}(\mathbf{p}_i) & d \text{ is even.} \end{cases}$$

Fig. 8 Algorithm: GRAPH^d .

We show the algorithm GRAPH^d in Fig. 8. For a graph $G = (V, E)$, let $E_1 \subseteq E$ be a set of selfloops and $E_2 = E - E_1$. For each $1 \leq i < j \leq d+1$, let E_{ij} be the set of edges between V_i and V_j . For a bipartite graph $G_i = (L_i \cup R_i, E_i)$, let $E_i^2 = E_i \cap E_2$ be the set of edges that are not selfloops. For the vertex pricing vector $\mathbf{p}_i$ in step 1 of the algorithm GRAPH^d , let $\text{PROFIT}_{E_i^2}(\mathbf{p}_i)$ and $\text{PROFIT}_{E_1}(\mathbf{p}_i)$ be the total profit returned from the edges in E_i^2 and the edges in E_1 , respectively.

We show the case where d is odd (for the case where d is even, the proof is similar). For the graph G with the maximum pseudodegree d , we use $\mathbf{p}^*$ to denote the optimal vertex pricing vector, i.e., $\text{PROFIT}(\mathbf{p}^*) = \text{OPT}$. For each $1 \leq i \leq N_o$, let $\mathbf{p}_i^*$ be the optimal vertex pricing vector for a bipartite graph $G_i = (L_i \cup R_i, E_i)$. Then we have that for each $1 \leq i \leq N_o$,

$$\begin{aligned} & \text{PROFIT}_{E_i^2}(\mathbf{p}_i^*) + \text{PROFIT}_{E_1}(\mathbf{p}_i^*) \\ & \geq \text{PROFIT}_{E_i^2}(\mathbf{p}^*) + \text{PROFIT}_{E_1}(\mathbf{p}^*). \end{aligned} \quad (7)$$

Since $\text{ALG}_{\text{BB}}^{\text{Bpt}}$ is a $1/2$ -approximation algorithm, it follows from Inequality (7) that

$$\begin{aligned} \text{PROFIT}(\mathbf{p}_i) & \geq \text{PROFIT}_{E_i^2}(\mathbf{p}_i) + \text{PROFIT}_{E_1}(\mathbf{p}_i) \\ & \geq \frac{1}{2} \{ \text{PROFIT}_{E_i^2}(\mathbf{p}_i^*) + \text{PROFIT}_{E_1}(\mathbf{p}_i^*) \} \\ & \geq \frac{1}{2} \{ \text{PROFIT}_{E_i^2}(\mathbf{p}^*) + \text{PROFIT}_{E_1}(\mathbf{p}^*) \}. \end{aligned} \quad (8)$$

For each $1 \leq s < t \leq d+1$, let I_{st} be the number of bipartite graphs $G_i = (L_i \cup R_i, E_i)$ such that $E_{st} \subseteq E_i^2$, i.e., I_{st} is the number of occurrences of E_{st} in the bipartite graphs G_i 's.

From the definition of E_{st} , we have that $I_{st} = 2 \binom{d-1}{(d-1)/2}$. Thus from Inequality (8),

$$\begin{aligned} \text{PROFIT}(\mathbf{p}) & \geq \frac{1}{N_o} \sum_{i=1}^{N_o} \text{PROFIT}(\mathbf{p}_i) \\ & \geq \frac{1}{2N_o} \sum_{i=1}^{N_o} \{ \text{PROFIT}_{E_i^2}(\mathbf{p}^*) + \text{PROFIT}_{E_1}(\mathbf{p}^*) \} \\ & = \frac{1}{2N_o} \sum_{i=1}^{N_o} \text{PROFIT}_{E_i^2}(\mathbf{p}^*) + \frac{\text{PROFIT}_{E_1}(\mathbf{p}^*)}{2} \\ & = \frac{1}{2N_o} 2 \binom{d-1}{\frac{d-1}{2}} \sum_{1 \leq s < t \leq d+1} \text{PROFIT}_{E_{st}}(\mathbf{p}^*) + \frac{\text{PROFIT}_{E_1}(\mathbf{p}^*)}{2} \\ & = \binom{d-1}{\frac{d-1}{2}} \frac{\text{PROFIT}_{E_2}(\mathbf{p}^*)}{N_o} + \frac{\text{PROFIT}_{E_1}(\mathbf{p}^*)}{2} \\ & = \left(1 + \frac{1}{d}\right) \frac{\text{PROFIT}_{E_2}(\mathbf{p}^*)}{4} + \frac{\text{PROFIT}_{E_1}(\mathbf{p}^*)}{2} \\ & \geq \left(1 + \frac{1}{d}\right) \frac{\text{PROFIT}_{E_2}(\mathbf{p}^*) + \text{PROFIT}_{E_1}(\mathbf{p}^*)}{4} \\ & = \left(1 + \frac{1}{d}\right) \frac{\text{OPT}}{4}. \end{aligned}$$

For the case where d is even, we can show the theorem in a way similar to the proof for the case where d is odd, except that $I_{st} = \binom{d-1}{d/2}$ for each $1 \leq s < t \leq d+1$. ■

4.3 Hypergraphs with Bounded Pseudodegree

For k -hypergraph vertex pricing problems, Balcan and Blum [1, Theorem 2] showed an α_k -approximation algorithm $\text{ALG}_{\text{BB}}^{\text{Hyper}}$ as shown in Fig. 9, where

$$\alpha_k = \frac{1}{k} \left(1 - \frac{1}{k}\right)^{k-1} > \frac{1}{ek} \approx \frac{0.3678}{k}.$$

Input: k -hypergraph $G = (V, E)$ with $\{w(e)\}_{e \in E}$.

1. For each vertex $v \in V$, choose v independently with probability $1/k$. Let L be a set of chosen vertices and $R = V - L$ be a set of unchosen vertices.
2. For each k -hyperedge $e \in E$, $e \in \hat{E} \subseteq E$ iff only one vertex of e is in L , i.e., $|e \cap L| = 1$, and let $\hat{G} = (V, \hat{E})$ be an k -hypergraph as a subgraph of G .
3. Set the prices of all vertices in R to 0 and compute a vertex pricing vector $\mathbf{q}$ for the k -hypergraph $\hat{G} = (V, \hat{E})$ that maximizes the total profit.

Fig. 9 Algorithm: $\text{ALG}_{\text{BB}}^{\text{Hyper}}$.

The algorithm $\text{ALG}_{\text{BB}}^{\text{Hyper}}$ is a randomized algorithm, but it can be derandomized by applying the sophisticated tools for the distributions due to Even, et al. [3].

For the k -hypergraph $G = (V, E)$, let $\mathbf{p}^*$ be an optimal vertex pricing vector that maximizes the total profit, i.e.,

Input: k -hypergraph $G = (V, E)$ with the maximum pseudodegree d .

1. By running the algorithm $\text{ALG}_{\text{BB}}^{\text{Hyper}}$ on $G = (V, E)$, compute the partition $(L : R)$ of V satisfying Inequality (9) and the graph $\hat{G} = (V, \hat{E})$.
1. For each $u \in R$, let $N_L(u) = \{v \in L : \{u, v\} \subseteq e \text{ for some } k\text{-hyperedge } e \in \hat{E}\}$.
2. For each pair of distinct vertices $s, t \in R$, $(s, t) \in F$ iff $N_L(s) \cap N_L(t) \neq \emptyset$.

Fig. 10 Definition of a Graph $H = (R, F)$.

$\text{PROFIT}(\mathbf{p}^*) = \text{OPT}$. For any vertex pricing vector $\mathbf{p}$, we use $\text{PROFIT}_L(\mathbf{p})$ to denote the profit returned from the vertices in L . Note that the algorithm $\text{ALG}_{\text{BB}}^{\text{Hyper}}$ computes the partition $(L : R)$ of V and the vertex pricing vector $\mathbf{q}$ such that

$$\text{ALG}_{\text{BB}}^{\text{Hyper}}(\mathbf{q}) \geq \text{PROFIT}_L(\mathbf{p}^*) \geq \alpha_k \cdot \text{OPT}. \quad (9)$$

For the k -hypergraph $G = (V, E)$ with the maximum pseudodegree d , we define a simple graph $H = (R, F)$ according to the procedure in Fig. 10.

Lemma 4.2: For a k -hypergraph $G = (V, E)$ with the maximum pseudodegree d , let $\Delta(H)$ be the maximum degree of a graph $H = (R, F)$ as defined in Fig. 10. Then $\Delta(H) \leq d\{d(k-1)-1\}$.

Proof: For the k -hypergraph $G = (V, E)$ with the maximum pseudodegree d , let $(L : R)$ be the partition of V satisfying Inequality (9) and $\hat{G} = (V, \hat{E})$ be the k -hypergraph for the partition $(L : R)$ of V . For each $u \in R$, let $N_L(u) = \{v \in L : \{u, v\} \subseteq e \text{ for some } k\text{-hyperedge } e \in \hat{E}\}$ and for each $v \in L$, $N_R(v) = \{u \in R : \{v, u\} \subseteq e \text{ for some } k\text{-hyperedge } e \in \hat{E}\}$. Since the maximum pseudodegree of the k -hypergraph G is d , it follows that the maximum pseudodegree of the k -hypergraph $\hat{G}$ is at most d . Then for each $u \in R$, we have that $|N_L(u)| \leq d$. From the definition of $\hat{E}$ and the fact that the maximum pseudodegree of the k -hypergraph $\hat{G} = (V, \hat{E})$ is at most d , we also have that $|N_R(v)| \leq d(k-1)$ for each $v \in L$. For the graph $H = (R, F)$, let $N(u) = \{v \in R : (u, v) \in F\}$ be neighbors of $u \in R$ in H . From the definition of H , it is immediate that for each $u \in R$,

$$N(u) = \bigcup_{v \in N_L(u)} \{N_R(v) - \{u\}\}.$$

For the graph $H = (R, F)$, we use $d_H(u)$ to denote the degree of $u \in R$. Note that $d_H(u) = |N(u)|$ for each $u \in R$ and that $u \in N_R(v)$ for each $v \in N_L(u)$. Thus we have that for each $u \in R$,

$$\begin{aligned} d_H(u) &= |N(u)| = \left| \bigcup_{v \in N_L(u)} \{N_R(v) - \{u\}\} \right| \\ &\leq \sum_{v \in N_L(u)} |N_R(v) - \{u\}| \\ &\leq \sum_{v \in N_L(u)} \{d(k-1)-1\} \end{aligned}$$

$$= |N_L(u)| \cdot \{d(k-1)-1\} \leq d\{d(k-1)-1\},$$

and for the maximum degree $\Delta(H)$ of the graph H , this implies that $\Delta(H) \leq d\{d(k-1)-1\}$. ■

Theorem 4.3: For any $k \geq 3$, let $G = (V, E)$ be a k -hypergraph vertex pricing problem with the maximum pseudodegree d . For any constant $\varepsilon > 0$, there exists a deterministic polynomial time algorithm $k\text{-HYPER}_\varepsilon^d$ that outputs a vertex pricing vector $\mathbf{p}$ such that

$$\text{PROFIT}(\mathbf{p}) \geq (1 - \varepsilon)\alpha_k \left\{ 1 + \frac{1/\alpha_k - 1}{d\{d(k-1)-1\} + 1} \right\} \text{OPT},$$

where $\alpha_k = \frac{1}{k}(1 - 1/k)^{k-1}$ for any integer $k \geq 3$.

Proof: The algorithm $\text{HYPER}_\varepsilon^d$ is given in Fig. 11.

Input: k -hypergraph $G = (V, E)$ with the maximum pseudodegree d .

1. For the k -hypergraph $G = (V, E)$, let $(L : R)$ be the partition of V satisfying Inequality (9) and $\hat{G} = (V, \hat{E})$ be the k -hypergraph in step 2 of the algorithm $\text{ALG}_{\text{BB}}^{\text{Hyper}}$.
2. For the k -hypergraph $\hat{G} = (V, \hat{E})$, define the graph $H = (R, F)$ as in Fig. 10 and let $B = \Delta(H)$ be the maximum pseudodegree of the graph H .
3. Color the vertices of $H = (R, F)$ with the set $C = \{c_1, c_2, \dots, c_{B+1}\}$ of $B+1$ colors and partition R into $R_1, R_2, \dots, R_{B+1}$ according to $c_i \in C$.
4. For each $1 \leq i \leq B+1$, let $\tilde{R}_i = R - R_i$ and $\tilde{L}_i = L - N_L(R_i)$, where $N_L(R_i) = \bigcup_{u \in R_i} N_L(u)$, and let $\tilde{V}_i = \tilde{L}_i \cup \tilde{R}_i$. For the k -hypergraph $G = (V, E)$, define a k -hypergraph $\tilde{G}_i = (\tilde{V}_i, \tilde{E}_i)$ such that for each $e \in E$, $e \in \tilde{E}_i$ iff $e \subseteq \tilde{V}_i$.
5. For each $1 \leq i \leq B+1$, set the prices of all vertices of $\tilde{R}_i$ to 0 and compute a vertex pricing vector $\tilde{\mathbf{p}}_i$ that maximizes the total profit of the k -hypergraph $\tilde{G}_i = (\tilde{V}_i, \tilde{E}_i)$.
6. For each $1 \leq i \leq B+1$ and each $u \in R_i$, define a star $S_i^u = (\{u\} \cup N_L(u), E_i^u)$, where $E_i^u = \{(u, v) : v \in N_L(u)\}$, and compute a vertex pricing vector $\mathbf{p}_i^u$ by running $\text{STAR}_\varepsilon^d$ on S_i^u .
7. For each $1 \leq i \leq B+1$, let $\mathbf{p}_i = \{\tilde{\mathbf{p}}_i\} \cup \{\mathbf{p}_i^u\}_{u \in R_i}$ and output a vertex pricing vector $\mathbf{p}$ such that

$$\text{PROFIT}(\mathbf{p}) = \max_{1 \leq i \leq B+1} \text{PROFIT}(\mathbf{p}_i).$$

Fig. 11 Algorithm: $\text{HYPER}_\varepsilon^d$.

From the definition of the graph $H = (R, F)$, it follows that for any pair of distinct $s, t \in R$, $(s, t) \in F$ iff $N_L(s) \cap N_L(t) \neq \emptyset$. Notice that for each $1 \leq i \leq B+1$, every $u \in R_i$ is colored with the same $c_i \in C$ in step 3 of the algorithm $\text{HYPER}_\varepsilon^d$, which implies that $(s, t) \notin F$ for any pair of distinct $s, t \in R_i$. Then for each $1 \leq i \leq B+1$ and any pair of distinct $s, t \in R_i$, we have that $N_L(s) \cap N_L(t) = \emptyset$. Thus for each $1 \leq i \leq B+1$, it follows that the set $\{S_i^u\}_{u \in R_i}$ of stars is disjoint.

For the k -hypergraph $G = (L \cup R, E)$, let $\mathbf{p}^*$ be the optimal vertex pricing vector, i.e., $\text{PROFIT}(\mathbf{p}^*) = \text{OPT}$. For the vertex pricing vector $\mathbf{p}^*$, we use $\text{PROFIT}_L(\mathbf{p}^*)$ and $\text{PROFIT}_R(\mathbf{p}^*)$ to denote the total profit returned from the vertices in L and R , respectively, and for each $1 \leq i \leq B+1$, we use

$\text{PROFIT}_{R_i}(\mathbf{p}^*)$ to denote the total profit returned from the vertices in R_i . For any constant $\varepsilon > 0$, it follows from Theorem 3.1 that $\text{PROFIT}(\mathbf{p}_i) \geq (1-\varepsilon) \cdot \{\text{PROFIT}_L(\mathbf{p}^*) + \text{PROFIT}_{R_i}(\mathbf{p}^*)\}$ for each $1 \leq i \leq B+1$. Then for the vertex pricing vector $\mathbf{p}$ in step 7 of the algorithm $\text{HYPER}_\varepsilon^d$, we have that

$$\begin{aligned}
\text{PROFIT}(\mathbf{p}) &\geq \frac{1}{B+1} \sum_{i=1}^{B+1} \text{PROFIT}(\mathbf{p}_i) \\
&\geq \frac{1-\varepsilon}{B+1} \sum_{i=1}^{B+1} \{\text{PROFIT}_L(\mathbf{p}^*) + \text{PROFIT}_{R_i}(\mathbf{p}^*)\} \\
&= (1-\varepsilon) \left\{ \text{PROFIT}_L(\mathbf{p}^*) + \frac{\text{PROFIT}_R(\mathbf{p}^*)}{B+1} \right\} \\
&= (1-\varepsilon) \left\{ \text{PROFIT}_L(\mathbf{p}^*) + \frac{\text{OPT} - \text{PROFIT}_L(\mathbf{p}^*)}{B+1} \right\} \\
&= (1-\varepsilon) \left\{ \frac{\text{OPT}}{B+1} + \frac{B}{B+1} \text{PROFIT}_L(\mathbf{p}^*) \right\} \\
&\geq (1-\varepsilon) \left\{ \frac{1}{B+1} \text{OPT} + \alpha_k \frac{B}{B+1} \text{OPT} \right\} \\
&= (1-\varepsilon) \alpha_k \left(1 + \frac{1/\alpha_k - 1}{B+1} \right) \text{OPT} \\
&\geq (1-\varepsilon) \alpha_k \left\{ 1 + \frac{1/\alpha_k - 1}{d\{d(k-1)-1\} + 1} \right\} \text{OPT}.
\end{aligned}$$

Note that the second equality follows from the fact that $\text{PROFIT}_L(\mathbf{p}^*) + \text{PROFIT}_R(\mathbf{p}^*) = \text{OPT}$, the third inequality follows from Inequality (9), and the last Inequality follows from Lemma 4.2. ■

5. Bounded Valuation Graphs

5.1 Overview

For a graph $G = (V, E)$ with a set $\{w(e)\}_{e \in E}$ of weights, let $r = \ell/h$ be the *valuation ratio*, where h is the largest valuation in $\{w(e)\}_{e \in E}$ and ℓ is the smallest valuation in $\{w(e)\}_{e \in E}$. Let $E_1 \subseteq E$ be a set of selfloops and $E_2 = E - E_1$. Let $\mathbf{p}^*$ be the optimal vertex pricing vector for G , i.e., $\text{PROFIT}(\mathbf{p}^*) = \text{OPT}$. For the vertex pricing vector $\mathbf{p}^*$, let $\text{PROFIT}_{E_1}(\mathbf{p}^*)$ and $\text{PROFIT}_{E_2}(\mathbf{p}^*)$ be the total profit returned from the edges in E_1 and E_2 , respectively. Let $x = \text{PROFIT}_{E_2}(\mathbf{p}^*)/\text{PROFIT}(\mathbf{p}^*)$.

Balcan and Blum [1, Theorem 1] showed that for a graph vertex pricing problem $G = (V, E)$ with a set $\{w(e)\}_{e \in E}$ of weights, the algorithm $\text{ALG}_{\text{BB}}^{\text{Graph}}$ outputs a vertex pricing vector α such that

$$\begin{aligned}
\text{PROFIT}(\alpha) &\geq \frac{\text{PROFIT}_{E_1}(\mathbf{p}^*)}{2} + \frac{\text{PROFIT}_{E_2}(\mathbf{p}^*)}{4} \\
&= \frac{\text{PROFIT}(\mathbf{p}^*) - \text{PROFIT}_{E_2}(\mathbf{p}^*)}{2} + \frac{\text{PROFIT}_{E_2}(\mathbf{p}^*)}{4} \\
&= \left(\frac{1}{2} - \frac{x}{4} \right) \text{OPT},
\end{aligned} \tag{10}$$

which is a decreasing function of $0 \leq x \leq 1$. Let $g : x \mapsto \mathbf{R}$

be an increasing function of $0 \leq x \leq 1$. For the graph vertex pricing problem $G = (V, E)$, an algorithm $\text{ALG}_{\text{BB}}^{\text{Graph}}$ is assumed to output a vertex pricing vector β such that $\text{PROFIT}(\beta) \geq g(x) \cdot \text{PROFIT}(\mathbf{p}^*)$. Then construct an algorithm $\text{ALG}_{\text{BB}}^{\text{Graph}}$ computing vertex pricing vectors α and β by running the algorithms $\text{ALG}_{\text{BB}}^{\text{Graph}}$ and $\text{ALG}_{\text{BB}}^{\text{Graph}}$ on G , respectively, and outputs a vertex pricing vector $\mathbf{p}$ such that

$$\mathbf{p} = \begin{cases} \alpha & \text{if } \text{PROFIT}(\alpha) \geq \text{PROFIT}(\beta); \\ \beta & \text{if } \text{PROFIT}(\alpha) < \text{PROFIT}(\beta), \end{cases}$$

which obviously improves the approximation ratio of the algorithm $\text{ALG}_{\text{BB}}^{\text{Graph}}$.

5.2 Algorithm for Bounded Valuation Graphs

In this subsection, we show an algorithm $\text{VALUATION}_r^{\text{Inc}}$ to improve the approximation ratio for the graph vertex pricing problem. As noted in Sect. 4.2, the algorithm $\text{ALG}_{\text{BB}}^{\text{Graph}}$ deterministically computes the partition $(L : R)$ of V that guarantees the approximation ratio of $1/4$ due to Balcan and Blum [1, Theorem 1]. In Fig. 12, we describe the algorithm $\text{VALUATION}_r^{\text{Inc}}$.

Input: graph $G = (V, E)$ with $\{w(e)\}_{e \in E}$.

1. For the graph $G = (V, E)$, run the algorithm $\text{ALG}_{\text{BB}}^{\text{Graph}}$ on $G = (V, E)$ to deterministically compute the partition $(L : R)$ of V . Let $|L| = n$ and $|R| = m$.
2. Let $E_2^L = \{e = (v, u) \in E_2 : v, u \in L\}$ and $E_2^R = \{e = (v, u) \in E_2 : v, u \in R\}$.
3. Let $\bar{E}_2^L = E - E_2^L$ and $\bar{E}_2^R = E - E_2^R$ and define graphs $G_L = (V, \bar{E}_2^L)$ and $G_R = (V, \bar{E}_2^R)$.
4. For the graph $G_R = (V, \bar{E}_2^R)$, set the prices of all vertices in L (denoted by $s_L = \mathbf{0}$) and compute a vertex pricing vector $s_R = (s_1^R, s_2^R, \dots, s_m^R)$ that maximizes the total profit returned by the edges in $\bar{E}_2^R$. For the vertex pricing vector $s = (s_L, s_R)$, define $\sigma = (\sigma_L, \sigma_R)$ by

$$\begin{aligned}
\sigma_L &= (\sigma_1^L, \sigma_2^L, \dots, \sigma_n^L) = (\ell/2, \ell/2, \dots, \ell/2); \\
\sigma_R &= (\sigma_1^R, \sigma_2^R, \dots, \sigma_m^R) \\
&= (s_1^R - \ell/2, s_2^R - \ell/2, \dots, s_m^R - \ell/2).
\end{aligned}$$
5. For the graph $G_L = (V, \bar{E}_2^L)$, set the prices of all vertices in R (denoted by $t_R = \mathbf{0}$) and compute a vertex pricing vector $t_L = (t_1^L, t_2^L, \dots, t_n^L)$ that maximizes the total profit returned by the edges in $\bar{E}_2^L$. For the vertex pricing vector $t = (t_L, t_R)$, define $\tau = (\tau_L, \tau_R)$ by

$$\begin{aligned}
\tau_L &= (\tau_1^L, \tau_2^L, \dots, \tau_n^L) \\
&= (t_1^L - \ell/2, t_2^L - \ell/2, \dots, t_n^L - \ell/2); \\
\tau_R &= (\tau_1^R, \tau_2^R, \dots, \tau_m^R) = (\ell/2, \ell/2, \dots, \ell/2).
\end{aligned}$$
6. Output a vertex pricing vector β such that

$$\beta = \begin{cases} \sigma & \text{if } \text{PROFIT}(\sigma) \geq \text{PROFIT}(\tau); \\ \tau & \text{if } \text{PROFIT}(\sigma) < \text{PROFIT}(\tau). \end{cases}$$

Fig. 12 Algorithm: $\text{VALUATION}_r^{\text{Inc}}$.

Lemma 5.1: Let $G = (V, E)$ be a graph vertex pricing problem with the valuation ratio r . There exists a deterministic polynomial time algorithm $\text{VALUATION}_r^{\text{Inc}}$ that outputs a vertex pricing vector $\mathbf{p}$ such that

$$\text{PROFIT}(\mathbf{p}) \geq \frac{x+r}{4} \text{OPT}.$$

The formal proof of Lemma 5.1 is given in Sect. 5.3. Then we can show the following theorem.

Theorem 5.1: Let $G = (V, E)$ be a graph vertex pricing problem with the valuation ratio r . There exists a deterministic polynomial time algorithm VALUATION_r that outputs a vertex pricing vector $\mathbf{p}$ such that

$$\text{PROFIT}(\mathbf{p}) \geq \frac{2+r}{8} \text{OPT}.$$

Proof: For the graph $G = (V, E)$ with a set $\{w(e)\}_{e \in E}$ of weights, the algorithm VALUATION_r computes vertex pricing vectors α and β by running the algorithms $\text{ALG}_{\text{BB}}^{\text{Graph}}$ and $\text{VALUATION}_r^{\text{Inc}}$ on $G = (V, E)$, respectively, and outputs a vertex pricing vector $\mathbf{p}$ such that

$$\mathbf{p} = \begin{cases} \alpha & \text{PROFIT}(\alpha) \geq \text{PROFIT}(\beta); \\ \beta & \text{PROFIT}(\alpha) < \text{PROFIT}(\beta), \end{cases}$$

From Inequality (10) and Lemma 5.1, we have that for the vertex pricing vectors α and β ,

$$\text{PROFIT}(\alpha) \geq \left(\frac{1}{2} - \frac{x}{4}\right) \text{OPT};$$

$$\text{PROFIT}(\beta) \geq \frac{x+r}{4} \text{OPT},$$

respectively. Thus it follows that

$$\begin{aligned} \text{PROFIT}(\mathbf{p}) &= \max \{\text{PROFIT}(\alpha), \text{PROFIT}(\beta)\} \\ &\geq \frac{\text{PROFIT}(\alpha) + \text{PROFIT}(\beta)}{2} \\ &\geq \max \left\{ \frac{1}{2} - \frac{x}{4}, \frac{x+r}{4} \right\} \text{OPT} \geq \frac{2+r}{8} \text{OPT}, \end{aligned}$$

where the last inequality follows from $r \in [0, 1]$ and the fact that the decreasing function $\frac{1}{2} - \frac{x}{4}$ and the increasing function $\frac{x+r}{4}$ intersect at $x = 1 - \frac{r}{2} \in [0, 1]$. ■

5.3 Proof of Lemma 5.1

For the graph $G = (V, E)$, let E_1^L and E_1^R be the set of self-loops in L and R , respectively, and for the sets E_2^L and E_2^R in step 2 of $\text{VALUATION}_r^{\text{Inc}}$, let $E_2^* = E_2 - (E_2^L \cup E_2^R)$. For a vertex pricing vector $\mathbf{q}$, let $\text{PROFIT}_{E'}(\mathbf{q})$ be the total profit returned from the edges in $E' \subseteq E$. For $\text{PROFIT}_{E_2^*}(\mathbf{q})$, let $\text{PROFIT}_{E_2^L}(\mathbf{q})$ and $\text{PROFIT}_{E_2^R}(\mathbf{q})$ be the total profit returned from the edges in E_2^* at the L -side and the R -side, respectively. From Balcan and Blum [1, Theorem 1], it follows that for the optimal vertex pricing vector $\mathbf{p}^*$, i.e., $\text{PROFIT}(\mathbf{p}^*) = \text{OPT}$,

$$\text{PROFIT}_{E_2^L}(\mathbf{p}^*) + \text{PROFIT}_{E_2^R}(\mathbf{p}^*)$$

$$\geq \frac{1}{2} \text{PROFIT}_{E_2}(\mathbf{p}^*). \quad (11)$$

From the definitions of E_2^L , E_2^R , and E_2^* , it is obvious that $E_2 = E_2^* \cup E_2^L \cup E_2^R$. Thus we have that

$$\begin{aligned} \text{PROFIT}_{E_2}(\mathbf{p}^*) &= \text{PROFIT}_{E_2^L}(\mathbf{p}^*) + \text{PROFIT}_{E_2^R}(\mathbf{p}^*) \\ &\quad + \text{PROFIT}_{E_2^*}(\mathbf{p}^*) + \text{PROFIT}_{E_2^*}(\mathbf{p}^*). \end{aligned} \quad (12)$$

The following claims hold for the vertex pricing vectors $\mathbf{s} = (s_L, s_R)$ and $\sigma = (\sigma_L, \sigma_R)$ in step 4 of the algorithm $\text{VALUATION}_r^{\text{Inc}}$ (the proofs of the claims are given in Appendices A.1 and A.2).

Claim 5.1: $\ell \leq s_1^R, s_2^R, \dots, s_m^R \leq h$ for the vertex pricing vector $\mathbf{s}_R = (s_1^R, s_2^R, \dots, s_m^R)$.

Claim 5.2: For the vertex pricing vectors $\mathbf{s} = (s_L, s_R)$ and $\sigma = (\sigma_L, \sigma_R)$ in step 4 of $\text{VALUATION}_r^{\text{Inc}}$, the following holds: (a) $\text{PROFIT}_{E_2^L}(\sigma) + \text{PROFIT}_{E_2^R}(\sigma) \geq \text{PROFIT}_{E_2^*}(\mathbf{p}^*)$; (b) $\text{PROFIT}_{E_2^L}(\sigma) \geq r \text{PROFIT}_{E_2^*}(\mathbf{p}^*)$; (c) $\text{PROFIT}_{E_2^R}(\sigma) \geq \frac{r}{2} \text{PROFIT}_{E_2^*}(\mathbf{p}^*)$.

Note that $\sigma_1^R, \sigma_2^R, \dots, \sigma_m^R \geq \ell/2$ for the vertex pricing vector $\sigma_R = (\sigma_1^R, \sigma_2^R, \dots, \sigma_m^R)$.

For the vertex pricing vectors $\mathbf{t} = (t_L, t_R)$ and $\tau = (\tau_L, \tau_R)$ in step 5 of the algorithm $\text{VALUATION}_r^{\text{Inc}}$, the following claims can also be shown in a way similar to the proofs of Claims 5.1 and 5.2.

Claim 5.3: $\ell \leq t_1^L, t_2^L, \dots, t_n^L \leq h$ for the vertex pricing vector $\mathbf{t}_L = (t_1^L, t_2^L, \dots, t_n^L)$.

Claim 5.4: For the vertex pricing vectors $\mathbf{t} = (t_L, t_R)$ and $\tau = (\tau_L, \tau_R)$ in step 5 of $\text{VALUATION}_r^{\text{Inc}}$, the following holds: (a) $\text{PROFIT}_{E_2^L}(\tau) + \text{PROFIT}_{E_2^R}(\tau) \geq \text{PROFIT}_{E_2^*}(\mathbf{p}^*)$; (b) $\text{PROFIT}_{E_2^L}(\tau) \geq r \text{PROFIT}_{E_2^*}(\mathbf{p}^*)$; (c) $\text{PROFIT}_{E_2^R}(\tau) \geq \frac{r}{2} \text{PROFIT}_{E_2^*}(\mathbf{p}^*)$.

From Claims 5.2 and 5.4, we have that for the vertex pricing vectors $\sigma = (\sigma_L, \sigma_R)$ and $\tau = (\tau_L, \tau_R)$ computed in steps 4 and 5 of the algorithm $\text{VALUATION}_r^{\text{Inc}}$,

$$\begin{aligned} \text{PROFIT}(\sigma) &\geq \text{PROFIT}_{E_2^L}(\sigma) + \text{PROFIT}_{E_2^R}(\sigma) \\ &\quad + \text{PROFIT}_{E_2^*}(\sigma) + \text{PROFIT}_{E_2^*}(\sigma) \\ &\geq \text{PROFIT}_{E_2^L}(\mathbf{p}^*) + \frac{r}{2} \text{PROFIT}_{E_2^*}(\mathbf{p}^*) \\ &\quad + r \text{PROFIT}_{E_2^*}(\mathbf{p}^*); \end{aligned} \quad (13)$$

$$\begin{aligned} \text{PROFIT}(\tau) &\geq \text{PROFIT}_{E_2^L}(\tau) + \text{PROFIT}_{E_2^R}(\tau) \\ &\quad + \text{PROFIT}_{E_2^*}(\tau) + \text{PROFIT}_{E_2^*}(\tau) \\ &\geq \text{PROFIT}_{E_2^L}(\mathbf{p}^*) + \frac{r}{2} \text{PROFIT}_{E_2^*}(\mathbf{p}^*) \\ &\quad + r \text{PROFIT}_{E_2^*}(\mathbf{p}^*). \end{aligned} \quad (14)$$

Thus for the vertex pricing vector β in step 6 of the algorithm $\text{VALUATION}_r^{\text{Inc}}$, it follows that

$$\text{PROFIT}(\beta) \geq \frac{1}{2} \{\text{PROFIT}(\sigma) + \text{PROFIT}(\tau)\}$$

$$\begin{aligned}
&\geq \frac{1}{2} \left\{ \text{PROFIT}_{E_2^*|R}(\mathbf{p}^*) + \frac{r}{2} \text{PROFIT}_{E_1^L}(\mathbf{p}^*) \right. \\
&\quad + r \text{PROFIT}_{E_2^L}(\mathbf{p}^*) + \text{PROFIT}_{E_2^*|L}(\mathbf{p}^*) \\
&\quad \left. + \frac{r}{2} \text{PROFIT}_{E_1^R}(\mathbf{p}^*) + r \text{PROFIT}_{E_2^R}(\mathbf{p}^*) \right\} \\
&= \frac{1}{2} \left[\text{PROFIT}_{E_2^*|R}(\mathbf{p}^*) + \text{PROFIT}_{E_2^*|L}(\mathbf{p}^*) \right. \\
&\quad + r \left\{ \text{PROFIT}_{E_2^L}(\mathbf{p}^*) + \text{PROFIT}_{E_2^R}(\mathbf{p}^*) \right\} \\
&\quad \left. + \frac{r}{2} \left\{ \text{PROFIT}_{E_1^L}(\mathbf{p}^*) + \text{PROFIT}_{E_1^R}(\mathbf{p}^*) \right\} \right] \\
&= \frac{1}{2} \left[\text{PROFIT}_{E_2^*|R}(\mathbf{p}^*) + \text{PROFIT}_{E_2^*|L}(\mathbf{p}^*) \right. \\
&\quad + r \left\{ \text{PROFIT}_{E_2^L}(\mathbf{p}^*) + \text{PROFIT}_{E_2^R}(\mathbf{p}^*) \right\} \\
&\quad \left. + \frac{r}{2} \text{PROFIT}_{E_1}(\mathbf{p}^*) \right] \tag{15}
\end{aligned}$$

$$\begin{aligned}
&= \frac{1}{2} \left[\text{PROFIT}_{E_2^*|R}(\mathbf{p}^*) + \text{PROFIT}_{E_2^*|L}(\mathbf{p}^*) \right. \\
&\quad - r \left\{ \text{PROFIT}_{E_2^*|L}(\mathbf{p}^*) + \text{PROFIT}_{E_2^*|R}(\mathbf{p}^*) \right\} \\
&\quad \left. + \frac{r}{2} \text{PROFIT}_{E_1}(\mathbf{p}^*) + r \text{PROFIT}_{E_2}(\mathbf{p}^*) \right] \tag{16} \\
&= \frac{1}{2} \left[r \text{PROFIT}_{E_2}(\mathbf{p}^*) + \frac{r}{2} \text{PROFIT}_{E_1}(\mathbf{p}^*) \right. \\
&\quad + (1-r) \left\{ \text{PROFIT}_{E_2^*|L}(\mathbf{p}^*) \right. \\
&\quad \quad \left. + \text{PROFIT}_{E_2^*|R}(\mathbf{p}^*) \right\} \left. \right] \\
&\geq \frac{1}{2} \left[r \text{PROFIT}_{E_2}(\mathbf{p}^*) + \frac{1-r}{2} \text{PROFIT}_{E_2}(\mathbf{p}^*) \right. \\
&\quad \left. + \frac{r}{2} \text{PROFIT}_{E_1}(\mathbf{p}^*) \right] \tag{17}
\end{aligned}$$

Notice that Equality (15) can be derived from the fact that $\text{PROFIT}_{E_1}(\mathbf{p}^*) = \text{PROFIT}_{E_1^L}(\mathbf{p}^*) + \text{PROFIT}_{E_1^R}(\mathbf{p}^*)$, Equality (16) follows from Equality (12), and Inequality (17) follows from Inequality (11). ■

5.4 Extension of Lemma 5.1

To improve Theorem 5.1, we extend Lemma 5.1.

Lemma 5.2: For a graph vertex pricing problem $G = (V, E)$

with the valuation ratio $r \geq 2/3$, $\text{VALUATION}_r^{\text{Inc}}$ outputs a vertex pricing vector $\mathbf{p}$ such that

$$\text{PROFIT}(\mathbf{p}) \geq \frac{(1-r)x + 2r}{4} \text{OPT}.$$

Proof: We note that for $0 \leq r \leq 1$, the vertex pricing vectors $\mathbf{s} = (s_L, s_R)$ and $\boldsymbol{\sigma} = (\sigma_L, \sigma_R)$ satisfy Claims 5.1 and 5.2, and the vertex pricing vectors $\mathbf{t} = (t_L, t_R)$ and $\boldsymbol{\tau} = (\tau_L, \tau_R)$ satisfy Claims 5.3 and 5.4. For $r \geq 2/3$, the following claim holds for the vertex pricing vector $\boldsymbol{\sigma} = (\sigma_L, \sigma_R)$.

Claim 5.5: If $r \geq 2/3$, then for the vertex pricing vector $\boldsymbol{\sigma} = (\sigma_L, \sigma_R)$ computed in step 4 of the algorithm $\text{VALUATION}_r^{\text{Inc}}$, $\text{PROFIT}_{E_1}(\boldsymbol{\sigma}) \geq \frac{r}{2} \text{PROFIT}_{E_1}(\mathbf{p}^*)$.

The proof of Claim 5.5 is given in Appendix A.3. In a way similar to Claim 5.5, the following claim holds for the vertex pricing vector $\boldsymbol{\tau} = (\tau_L, \tau_R)$.

Claim 5.6: If $r \geq 2/3$, then for the vertex pricing vector $\boldsymbol{\tau} = (\tau_L, \tau_R)$ computed in step 5 of the algorithm $\text{VALUATION}_r^{\text{Inc}}$, $\text{PROFIT}_{E_1}(\boldsymbol{\tau}) \geq \frac{r}{2} \text{PROFIT}_{E_1}(\mathbf{p}^*)$.

From Claims 5.5 and 5.6, we can show in a way similar to Inequalities (13) and (14) that for the vertex pricing vectors $\boldsymbol{\sigma} = (\sigma_L, \sigma_R)$ and $\boldsymbol{\tau} = (\tau_L, \tau_R)$ in steps 4 and 5 of the algorithm $\text{VALUATION}_r^{\text{Inc}}$,

$$\begin{aligned}
\text{PROFIT}(\boldsymbol{\sigma}) &\geq \text{PROFIT}_{E_2^*|R}(\mathbf{p}^*) \\
&\quad + \frac{r}{2} \text{PROFIT}_{E_1}(\mathbf{p}^*) + r \text{PROFIT}_{E_2^L}(\mathbf{p}^*); \\
\text{PROFIT}(\boldsymbol{\tau}) &\geq \text{PROFIT}_{E_2^*|L}(\mathbf{p}^*) \\
&\quad + \frac{r}{2} \text{PROFIT}_{E_1}(\mathbf{p}^*) + r \text{PROFIT}_{E_2^R}(\mathbf{p}^*).
\end{aligned}$$

Then for the vertex pricing vector $\boldsymbol{\beta}$ in step 6 of the algorithm $\text{VALUATION}_r^{\text{Inc}}$, it follows that

$$\begin{aligned}
\text{PROFIT}(\boldsymbol{\beta}) &\geq \frac{1}{2} \{ \text{PROFIT}(\boldsymbol{\sigma}) + \text{PROFIT}(\boldsymbol{\tau}) \} \\
&\geq \frac{1}{2} \left\{ \text{PROFIT}_{E_2^*|R}(\mathbf{p}^*) + \frac{r}{2} \text{PROFIT}_{E_1}(\mathbf{p}^*) \right. \\
&\quad + r \text{PROFIT}_{E_2^L}(\mathbf{p}^*) + \text{PROFIT}_{E_2^*|L}(\mathbf{p}^*) \\
&\quad \left. + \frac{r}{2} \text{PROFIT}_{E_1}(\mathbf{p}^*) + r \text{PROFIT}_{E_2^R}(\mathbf{p}^*) \right\} \\
&= \frac{1}{2} \left[\text{PROFIT}_{E_2^*|R}(\mathbf{p}^*) + \text{PROFIT}_{E_2^*|L}(\mathbf{p}^*) \right. \\
&\quad + r \text{PROFIT}_{E_1}(\mathbf{p}^*) \\
&\quad \left. + r \left\{ \text{PROFIT}_{E_2^L}(\mathbf{p}^*) + \text{PROFIT}_{E_2^R}(\mathbf{p}^*) \right\} \right] \\
&= \frac{1}{2} \left[\text{PROFIT}_{E_2^*|R}(\mathbf{p}^*) + \text{PROFIT}_{E_2^*|L}(\mathbf{p}^*) \right. \\
&\quad + r \text{PROFIT}_{E_1}(\mathbf{p}^*) + r \text{PROFIT}_{E_2}(\mathbf{p}^*) \\
&\quad \left. - r \left(\text{PROFIT}_{E_2^*|L}(\mathbf{p}^*) + \text{PROFIT}_{E_2^*|R}(\mathbf{p}^*) \right) \right]
\end{aligned}$$

$$\begin{aligned}
&= \frac{1}{2} \left[r \text{PROFIT}_{E_2}(\mathbf{p}^*) + r \text{PROFIT}_{E_1}(\mathbf{p}^*) \right. \\
&\quad \left. + (1-r) \text{PROFIT}_{E_2|L}(\mathbf{p}^*) \right. \\
&\quad \left. + (1-r) \text{PROFIT}_{E_2|R}(\mathbf{p}^*) \right] \\
&\geq \frac{1}{2} \left[r \text{PROFIT}_{E_2}(\mathbf{p}^*) + \frac{1-r}{2} \text{PROFIT}_{E_2}(\mathbf{p}^*) \right. \\
&\quad \left. + r \text{PROFIT}_{E_1}(\mathbf{p}^*) \right] \\
&= \frac{1+r}{4} \text{PROFIT}_{E_2}(\mathbf{p}^*) + \frac{r}{2} \text{PROFIT}_{E_1}(\mathbf{p}^*) \\
&= \frac{1+r}{4} x \text{PROFIT}(\mathbf{p}^*) + \frac{r}{2} (1-x) \text{PROFIT}(\mathbf{p}^*) \\
&= \frac{(1-r)x + 2r}{4} \text{PROFIT}(\mathbf{p}^*) \\
&= \frac{(1-r)x + 2r}{4} \text{OPT.}
\end{aligned}$$

Notice that the evaluations above are almost the same as the proof of Lemma 5.1. $\blacksquare$

From Lemma 5.2, it is immediate to show the following theorem that improves Theorem 5.1.

Theorem 5.2: Let $G = (V, E)$ be a graph vertex pricing problem with the valuation ratio r . There exists a deterministic polynomial time algorithm VALUATION_r^* that outputs a vertex pricing vector $\mathbf{p}$ such that

$$\text{PROFIT}(\mathbf{p}) \geq \begin{cases} \frac{2+r}{8} \text{OPT} & \text{if } 0 < r < \frac{2}{3}; \\ \frac{1}{2(2-r)} \text{OPT} & \text{if } \frac{2}{3} \leq r \leq 1. \end{cases}$$

Proof: For the graph $G = (V, E)$ with a set $\{w(e)\}_{e \in E}$ of weights, the algorithm VALUATION_r^* computes vertex pricing vectors α and β by running the algorithms $\text{ALG}_{\text{BB}}^{\text{Graph}}$ and $\text{VALUATION}_r^{\text{Inc}}$ on $G = (V, E)$, respectively, and outputs a vertex pricing vector $\mathbf{p}$ such that

$$\mathbf{p} = \begin{cases} \alpha & \text{PROFIT}(\alpha) \geq \text{PROFIT}(\beta); \\ \beta & \text{PROFIT}(\alpha) < \text{PROFIT}(\beta). \end{cases}$$

Thus for any $0 \leq r < 2/3$, it follows from Theorem 5.1 that $\text{PROFIT}(\mathbf{p}) \geq \frac{2+r}{8} \text{OPT}$. On the other hand, it follows from Lemma 5.2 that for any $r \geq 2/3$,

$$\begin{aligned}
\text{PROFIT}(\mathbf{p}) &= \max \{ \text{PROFIT}(\alpha), \text{PROFIT}(\beta) \} \\
&\geq \frac{\text{PROFIT}(\alpha), \text{PROFIT}(\beta)}{2} \\
&\geq \max \left\{ \frac{1}{2} - \frac{x}{4}, \frac{(1-r)x + 2r}{4} \right\} \text{OPT} \\
&\geq \frac{1}{2(2-r)} \text{OPT},
\end{aligned}$$

where the last inequality follows from the definition of

$r \in [2/3, 1]$ and the fact that the decreasing function $\frac{1}{2} - \frac{x}{4}$ and the increasing function $\frac{(1-r)x + 2r}{4}$ intersect at $x = \frac{2-2r}{2-r} \in [0, \frac{1}{2}] \subseteq [0, 1]$. $\blacksquare$

References

- [1] M. Balcan and A. Blum, "Approximation algorithms and online mechanisms for item pricing," Proc. 7th ACM Conference on Electronic Commerce, pp.29–35, 2006.
- [2] P. Briest and P. Krysta, "Single-minded unlimited supply pricing on sparse instances," Proc. 17th Annual ACM-SIAM Symposium on Discrete Algorithms, pp.1093–1102, 2006.
- [3] G. Even, O. Goldreich, M. Luby, N. Nisan, and B. Veličković, "Efficient approximation of product distributions," Random Structure and Algorithms, vol.13, no.1, pp.1–16, 1998.
- [4] V. Guruswami, J.D. Hartline, A.R. Karlin, D. Karger, C. Kenyon, and F. McSherry, "On profit-maximizing envy-free pricing," Proc. 16th Annual ACM-SIAM Symposium on Discrete Algorithms, pp.1164–1174, 2005.
- [5] R. Motowani and P. Raghavan, Randomized Algorithms, Cambridge University Press, 1995.
- [6] M. Luby and A. Wigderson, "Pairwise independence and derandomization," UC Berkeley Technical Report: CSD-95-880, 1995.

Appendix: Proofs of Claims

A.1 Proof of Claim 5.1

From the definition of h , it is easy to see that $s_i^R \leq h$ for each $1 \leq i \leq m$. From the definition of the vertex pricing vector $\mathbf{s} = (s_L, s_R) = (\mathbf{0}, s_R)$ in step 4 of $\text{VALUATION}_r^{\text{Inc}}$, it follows that

$$\text{PROFIT}_{\bar{E}_2}^R(\mathbf{0}, s_R) = \max_{\mathbf{q}_R} \text{PROFIT}_{\bar{E}_2}^R(\mathbf{0}, \mathbf{q}_R).$$

From the definition of ℓ , we have that $w(e) \geq \ell$ for each $e \in E$. Assume that there exists $1 \leq j \leq m$ such that $s_j^R < \ell$ and define $\tilde{s}_R = (\tilde{s}_1^R, \tilde{s}_2^R, \dots, \tilde{s}_m^R)$ to be $\tilde{s}_j^R = \ell$ and $\tilde{s}_i^R = s_i^R$ for each $i \neq j$. From the facts that $s_L = \mathbf{0}$ and $\bar{E}_2^R = E - E_2^R$, it follows that

$$\text{PROFIT}_{\bar{E}_2}^R(\mathbf{0}, s_R) < \text{PROFIT}_{\bar{E}_2}^R(\mathbf{0}, \tilde{s}_R),$$

which contradicts the definition of s_R . So we have that $s_i^R \geq \ell$ for each $1 \leq i \leq m$.

A.2 Proof of Claim 5.2

We first show the statement (a) of the claim. From the definitions of $\mathbf{s} = (s_L, s_R) = (\mathbf{0}, s_r)$, we have that

$$\text{PROFIT}_{E_2|R}^R(\mathbf{s}) \geq \text{PROFIT}_{E_2|R}^R(\mathbf{p}^*).$$

For vertex pricing vectors s_L, s_R, σ_L , and σ_R in step 4 of the algorithm $\text{VALUATION}_r^{\text{Inc}}$, it is immediate that for each $1 \leq i \leq n$ and each $1 \leq j \leq m$,

$$s_i^L + s_j^R = \left(s_i^L + \frac{\ell}{2} \right) + \left(s_j^R - \frac{\ell}{2} \right) = \sigma_i^L + \sigma_j^R,$$

and this implies that

$$\begin{aligned} \text{PROFIT}_{E_2^L}(\sigma) + \text{PROFIT}_{E_2^R}(\sigma) \\ = \text{PROFIT}_{E_2^L}(\sigma). \end{aligned}$$

We then show the statement (b) of the claim. Notice that $\ell \leq w(e) \leq h$ for each $e \in E$. Then for the vertex pricing vector σ in step 4 of the algorithm $\text{VALUATION}_r^{\text{Inc}}$, it follows that

$$\text{PROFIT}_{E_2^L}(\sigma) = |E_2^L| \cdot \ell; \quad \text{PROFIT}_{E_2^L}(\mathbf{p}^*) \leq |E_2^L| \cdot h.$$

Since $r = \ell/h$, we have that

$$\begin{aligned} \text{PROFIT}_{E_2^L}(\sigma) &= |E_2^L| \cdot \ell = |E_2^L| \cdot r \cdot h \\ &\geq r \cdot \text{PROFIT}_{E_2^L}(\mathbf{p}^*). \end{aligned}$$

We finally show the statement (c) of the claim. Notice again that $\ell \leq w(e) \leq h$ for each $e \in E$. Then for the vertex pricing vector σ in step 4 of $\text{VALUATION}_r^{\text{Inc}}$, it follows that

$$\text{PROFIT}_{E_1^L}(\sigma) = |E_1^L| \cdot \frac{\ell}{2}; \quad \text{PROFIT}_{E_1^L}(\mathbf{p}^*) \leq |E_1^L| \cdot h.$$

Since $r = \ell/h$, we have that

$$\begin{aligned} \text{PROFIT}_{E_1^L}(\sigma) &= |E_1^L| \cdot \frac{\ell}{2} = |E_1^L| \cdot \frac{r}{2} \cdot h \\ &\geq \frac{r}{2} \cdot \text{PROFIT}_{E_1^L}(\mathbf{p}^*). \end{aligned}$$

A.3 Proof of Claim 5.5

Note that $\ell \leq w(e) \leq h$ for each $e \in E$. From the definition of the vertex pricing vector σ in step 4 of $\text{VALUATION}_r^{\text{Inc}}$ and Claim 5.1, it follows that for each $1 \leq i \leq m$,

$$\frac{\ell}{2} \leq \sigma_i^R - \frac{\ell}{2} \leq h - \frac{\ell}{2}.$$

Then from the assumption that $r = \ell/h \geq 2/3$, it is immediate to see that for each $1 \leq i \leq m$,

$$\frac{\ell}{2} \leq \sigma_i^R \leq h - \frac{\ell}{2} \leq \ell.$$

Thus from the definition of the vertex pricing vector σ ,

$$\text{PROFIT}_{E_1^R}(\sigma) \geq |E_1^R| \cdot \frac{\ell}{2}; \quad \text{PROFIT}_{E_1^R}(\mathbf{p}^*) \leq |E_1^R| \cdot h.$$

Since $r = \ell/h$, we have that

$$\begin{aligned} \text{PROFIT}_{E_1^R}(\sigma) &\geq |E_1^R| \cdot \frac{\ell}{2} = |E_1^R| \cdot \frac{r}{2} \cdot h \\ &\geq \frac{r}{2} \cdot \text{PROFIT}_{E_1^R}(\mathbf{p}^*). \end{aligned}$$

Ryoso Hamane was born in Hiroshima, Japan, in 1985. He received the B.E. degree in 2007 from Tokyo Institute of Technology, and since 2007, he has been a master course student in the Department of Information Processing at Tokyo Institute of Technology. His current research interests are discrete algorithms, approximation algorithms, and complexity theory.

Toshiya Itoh was born in Urawa, Japan, in 1959. He received the B.E., M.E., and Dr.Eng. degree in electronic engineering in 1982, 1984, and 1988, respectively from Tokyo Institute of Technology, Tokyo, Japan. From 1985 to 1990, he was an Assistant Professor in the Department of Electrical and Electronic Engineering at Tokyo Institute of Technology, and from 1990 to 1992, he was a Lecturer in the Department of Information Processing at Tokyo Institute of Technology. From 1992 to 2001, he was an Associate

Professor in the Department of Information Processing at Tokyo Institute of Technology, and since 2001, he has been a Professor in the Global Scientific Information and Computing Center at Tokyo Institute of Technology. His current interests are discrete algorithms, combinatorics, and complexity theory. Dr. Itoh is a member of the Institute of Electronics, Information, and Communication Engineers of Japan, the Information Processing Society of Japan, the Association for Computing Machinery, and L.A.