Algorithm 3 Pricing algorithm for collaborating buyers

For $s = 1, 2, \ldots, |V|$
  Let X^s be s largest elements of $f(x)$ $(= \max_i f_i(x))$
  Price $p^s(x) = \min_{i \in N} \frac{f_i(X^s) - f_i(X^s \setminus x)}{f_i(x)} f(x)$ $(x \in X^s)$
and $p(y) = +\infty$ $(y \in V \setminus X^s)$
Return p and X that attains maximum of $p^s(X^s)$

$\sum_{x \in X} \min_{Y \subseteq X : x \in Y} (f(Y) - f(Y \setminus x))$. Thus, we can apply the same principles as the ones of Algorithm 1. In fact, setting prices $p^s(x) = \min_{Y \subseteq X^s : x \in Y} (f(Y) - f(Y \setminus x))$ implies a similar result. However, computing $f(X)$ is intractable (this problem is called submodular welfare problem) and hence it is hard to evaluate the value $\min_{Y \subseteq X : x \in Y} (f(Y) - f(Y \setminus x))$. Thus, we need a further modification.

Our algorithm, summarized in Algorithm 3, finds an approximate solution to (8) in $O(An|V|^2 + |V| \log |V|)$ time, where A is the computational cost of evaluating $f_i(X)$ $(i \in N)$. We analyze the approximation ratio of our algorithm. Let $\kappa_1, \ldots, \kappa_n$ be curvatures of $f_1, \ldots, f_n$ and $\kappa(s) = \max_j \kappa_j(s)$ for $s = 1, \ldots, |V|$.

Theorem 12. Let (p^*, X^*) be an optimal solution to (8), and let (p, X) be the output of Algorithm 3. It then holds that $X \in D(p)$ and $(1 - \kappa(|X^*|)) p^*(X^*) \leq p(X)$.

To prove this theorem, we show the following two lemmas.

Lemma 13. For a set X and $x \in X$, it holds that $f(X) - f(X \setminus x) \leq f(x)$.

Lemma 14. For a set X and $x \in X$, it holds that $f(X) - f(X \setminus x) \geq \min_{i \in N} \frac{f_i(X) - f_i(X \setminus x)}{f_i(x)} f(x) \geq (1 - \kappa(|X|)) f(x)$.

6 Experiments

In this section, we present experimental results on our pricing algorithms for a variant of budget allocation problem, which are described in Section 2.3. All experiments were conducted on an Intel Xeon E5-2690 2.90GHz CPU (32 cores) with 256GB memory running Ubuntu 12.04. All codes were implemented in Python 2.7.3.

We performed the following five experiments: For the single advertiser case, (1) we computed prices of each channel for a realistic dataset; (2) we compared the proposed algorithm with other baseline algorithms; (3) we evaluated the scalability of the proposed algorithm; and (4) we observed the relationship between the activation probabilities and the number of allocated channels. For the multiple advertisers case and the multiple collaborating advertisers case, (5) we observed the relationship between the obtained profit and the number of advertisers.

For these experiments, we used two random synthetic networks (`Uniform`, `PowerLaw`) and three networks constructed from real-world datasets (`Last.fm`, `MovieLens`, `BookCrossing`). Throughout the experiments, we assume that the expected revenue from one loyal customer is 1, i.e., $\gamma_i = 1$. The description of the datasets is given in Appendix.

Table 1: Ranking by #plays.

rank	artist – music	#play	UU
1	The Postal Service – Such Great Heights	3992	321
2	Boy Division – Love Will Tear Us Apart	3663	318
3	Radiohead – Karma Police	3534	346
4	Muse – Supermassive Black Hole	3483	263
5	Death Cab For Cutie – Soul Meets Body	3479	233
6	The Knife – Heartbeats	3156	177
7	Muse – Starlight	3060	260
8	Arcade Fire – Rebellion (Lies)	3048	292
9	Britney Spears – Gimme More	3004	59
10	The Killers – When You Were Young	2998	235

Table 2: Ranking by prices.

rank	original	artist – music	price
1	1	The Postal Service – Such Great Heights	3.330
2	8	Arcade Fire – Rebellion (Lies)	2.101
3	4	Muse – Supermassive Black Hole	2.029
4	11	Interpol – Evil	2.026
5	3	Radiohead – Karma Police	2.003
6	6	The Knife – Heartbeats	1.992
7	12	Kanye West – Love Lockdown	1.893
8	17	Arcade Fire – Neighborhood #1 (Tunnels)	1.868
9	23	Kanye West – Heartless	1.788
10	24	Radiohead – Nude	1.770

(1) Typical result First, we ran Algorithm 1 to `Last.fm` dataset to compute prices for the musics played in Last.fm. Top 10 frequently played musics and top 10 high price musics are displayed in Table 1 and Table 2, respectively. We can observe that some musics with a large number of plays (or unique users) are not assigned high prices. This occurs because of the stability condition.

(2) Comparison with other pricing algorithms Next, we compared Algorithm 1 with the following four baseline algorithms:

Selling all items. Assign $X = V$ and price $p(v) = f(V) - f(V \setminus v)$ for each $v \in V$. This algorithm gives a stable assignment.

Random pricing. Price $p(v) \in [0, f(v)]$ uniformly at random for each $v \in V$ and find an assignment X by the greedy algorithm.

Scaled pricing. Price $p(v) = \alpha f(v)$ for each v and find an assignment X by the greedy algorithm. α is chosen optimally from $\{0.1, \ldots, 1.0\}$. 1.[Ascending pricing.] Start

Table 3: Comparison of pricing algorithms on several datasets. Each value is the ratio of the profit obtained by the algorithm and the proposed algorithm.

	Proposed	SellAll	Random	Scaled	Ascend
`Uniform`	**1.00**	0.89	0.55	0.98	0.96
`PowerLaw`	**1.00**	0.89	0.65	0.98	0.51
`Last.fm`	**1.00**	0.71	0.46	0.99	0.78
`MovieLens`	**1.00**	0.58	0.48	0.96	0.67
`BookCrossing`	**1.00**	**1.00**	0.39	0.78	0.43

from $X = V$ and $p(v) = 0$ $(v \in V)$, repeat the following process: Price $p(v) = \min_{X: x \in X}(f(X) - f(X \setminus x))$ for each $v \in X$, remove $\tilde{x}$ that attains the minimum from X, and then price $p(\tilde{x}) = +\infty$. This algorithm is motivated by the ascending auction (**?**).

We remark that there are no existing algorithms that are directly applicable to the optimal pricing problem with submodular valuations (see also Section 1.3).

We used all the networks described above; we set $|V| = 100$, $|W| = 10000$, $d = 10$ and $q_{\max} = 0.3$ for `Uniform` and `PowerLaw`. The result is summarized in Table 3. The proposed algorithm outperforms all compared algorithms for all datasets

(3) Scalability We evaluated the scalability of the proposed algorithm. We used `Uniform` with $|V| \in \{16, 32, \ldots, 1024\}$, $|W| \in \{100, 1000, 10000, 100000\}$, $d = 10$, and $q_{\max} = 0.3$. We also conducted the same experiment on `PowerLaw` but we omit it since it yields very similar results.

The result is shown in Figure 1. The elapsed times were (roughly) proportional to both $|V|$ and $|W|$. This is consistent with our analysis that the proposed algorithm runs in $O(|V||E|)$ time, and the number of edges is proportional to $|W|$ for these networks. Therefore, the proposed algorithm scales to moderately large networks.

(4) Number of allocated channels and activation probabilities We observe the relationship between activation probability and the obtained allocation. We used `Uniform` and `PowerLaw` with the parameters $|V| = 100$, $|W| = 10000$, and $d = 10$. We controlled the maximum activation probability $q_{\max} \in \{0.05, 0.10, \ldots, 0.95\}$ and observe the number of assigned marketing channels.

The result is shown in Figure 2. For both networks, when $q_{\max}$ was small the proposed algorithm assigned all channels, and when $q_{\max}$ was large it assigned a few channels. The number of assigned channels decreased much faster in `PowerLaw` than in `Uniform`, since there were highly correlated marketing channels in `PowerLaw`.

(5) Profit and the number of advertisers Next, we conducted experiments on the multiple advertisers case and the multiple collaborating advertisers case. Here we observe the relationship between the profit and the number of advertisers in these settings. For these experiments, we modified `Uniform` and `PowerLaw` to assign multiple probabilities $q_1(e), \ldots, q_n(e)$ for each edge, each of which follows the uniform distribution on $[0, q_{\max}]$.

The result is shown in Figure 3. By comparing two results obtained by the multiple (non-collaborating) advertisers case, the number of advertisers had little influence on the profit. On the other hand, by comparing the results obtained by collaborating advertisers, the profit increased when the number of advertisers increased. Moreover, the profits obtained from the collaborating advertisers consistently outperformed those obtained from non-collaborating advertisers. This means that collaboration of advertisers yields a better profit to the publisher. Note that we could not observe the difference between `Uniform` and `PowerLaw`.

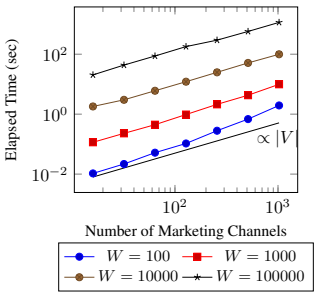

Figure 1: Scalability of the proposed algorithm.

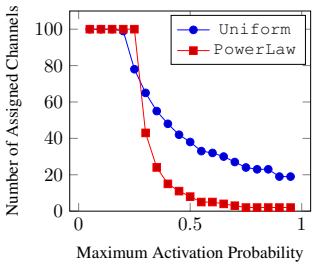

Figure 2: The number of assigned channels versus edge probability.

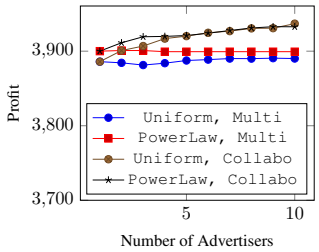

Figure 3: Profit versus the number of non-collaborating or collaborating advertisers.

7 Conclusion

We propose some future works. One is to develop an approximate pricing algorithm for the case that multiple buyers have limited budgets. Another one is to analyze the optimal pricing problem with multiple sellers. In this study, we assumed there is one seller; who can be regarded as a monopolist. The seller can select both the assignment and the price as long as they satisfy the stability condition. This situation is highly advantageous for the seller. Finally, in this study, we do not consider nonlinear or non-anonymous pricing. It would be interesting in future work to analyze the effect of such generalizations of pricing.

Acknowledgments This work was supported by JSPS KAKENHI Grant Number 16K16011, and JST, ERATO, Kawarabayashi Large Graph Project.

Odit eveniet facere excepturi blanditiis aut, fuga sapiente in, placeat alias delectus eveniet quas et corrupti, nesciunt quos voluptatibus necessitatibus quae consequatur accusamus delectus hic, minima quos non quasi cumque quas ex aspernatur.Sint eum fuga explicabo doloribus magni est, dolorum soluta ipsam pariatur ducimus voluptatum facilis eveniet, recusandae necessitatibus in nulla quae deserunt ipsum sed perferendis, vitae doloremque neque, exercitationem nemo fuga quas quae amet consequuntur.Placeat iure eum laudantium beatae unde quasi laborum harum in molestiae, corrupti cum accusamus placeat error dolorem ut repellendus reiciendis illo sed, blanditiis saepe omnis nemo expedita?Dolorem possimus in corporis iusto nesciunt provident dolores, doloremque repudiandae vel rerum