Supplementary Material:
Threshold Influence Model for Allocating Advertising Budgets

Atsushi Miyauchi MIYAUCHI.A.AA@M.TITECH.AC.JP
Graduate School of Decision Science and Technology, Tokyo Institute of Technology, Japan

Yuni Iwamasa YUNI_IWAMASA@MIST.I.U-TOKYO.AC.JP
Graduate School of Information Science and Technology, University of Tokyo, Japan

Takuro Fukunaga TAKURO@NII.AC.JP
National Institute of Informatics and JST, ERATO, Kawarabayashi Large Graph Project, Japan

Naonori Kakimura KAKIMURA@GLOBAL.C.U-TOKYO.AC.JP
Graduate School of Arts and Sciences, University of Tokyo, Japan

A. Proof of Theorem 1 (in Section 2)

Proof. We reduce the maximum thresholds coverage problem to the cost-effective version of the problem. Suppose that we are given a bipartite graph $G = (S, T; E)$ with weight w_t and threshold θ_t for each $t \in T$ and budget k being a positive integer. We consider the unweighted case, i.e., $w_t = 1$ for each $t \in T$. We add $p > k \cdot |T|$ new nodes $u_1, \dots, u_p$ to T. Each vertex u_i is connected to all the nodes in S, and has threshold $\theta_{u_i} = k$. The resulting graph is denoted by G'. Note that $f_{G'}(X) = f_G(X)$ if $|X| < k$ and $f_{G'}(X) = f_G(X) + p$ otherwise.

We claim that any most cost-effective solution for G' has to have size k. Indeed, for any $X, Y \subseteq S$ with $|X| < k$ and $|Y| = k$,

$$\frac{f_{G'}(X)}{|X|} \leq |T| < \frac{p}{k} \leq \frac{f_G(Y) + p}{|Y|} = \frac{f_{G'}(Y)}{|Y|}.$$

Moreover, for any $X, Y \subseteq S$ with $|X| > k$ and $|Y| = k$,

$$\frac{f_{G'}(X)}{|X|} \leq \frac{|T| + p}{|X|} < \frac{p}{k} \leq \frac{f_G(Y) + p}{|Y|} = \frac{f_{G'}(Y)}{|Y|}.$$

Thus the claim holds.

Therefore, a vertex subset X is the most cost-effective solution for G' if and only if X is an optimal solution for G. Thus the theorem holds. □

B. Proof of Lemma 1 (in Section 3)

Proof. For each $s \in S$, we make u_s copies $(s, 1), \dots, (s, u_s)$ of s, and connect each (s, i) to the neighbors of s in G. The resulting graph is denoted by $\tilde{G}$, which has node sets $\tilde{S} = \{(s, i) \mid s \in S, i = 1, \dots, u_s\}$ and T. From the given function $f_t \colon \mathbb{Z}_+^{\Gamma(t)} \to \mathbb{R}$, construct a set function $\tilde{f}_t \colon 2^{\tilde{S}_t} \to \mathbb{R}$ as mentioned in the main text, where $\tilde{S}_t = \{(s, i) \mid s \in \Gamma(t), i \in \{0, 1, \dots, u_s\}\}$. Moreover, we define the cost $c(s, i) = c_s i$ as the weight of $(s, i) \in \tilde{S}$. Then consider solving this problem to find a solution $X \subseteq \tilde{S}$ that satisfies $\sum_{(s,i) \in X} c(s, i) \leq B$ and maximizes $\tilde{f}(X)$. Suppose that $X \subseteq \tilde{S}$ is an α-approximate solution for the problem. The corresponding vector is denoted by x. Then we have $f(x) = \tilde{f}(X)$ and $\sum_{s \in S} c_s x_s \leq \sum_{(s,i) \in X} c_s i \leq B$. Hence x is a feasible solution to the original instance, whose objective value is $\tilde{f}(X)$. Since the optimal value to the original instance is equal to that of the reduced instance, x is an α-approximate solution to the original instance.

Since $|\tilde{S}| = \sum_{s \in S} u_s$, the size of the reduced problem is pseudo-polynomial in the input size. Thus the obtained instance has size pseudo-polynomial in the input size. □

C. Proof of Theorem 2 (in Section 3)

Proof. First suppose that $c_s = 1$ for each $s \in S$ and $w_t = 1$ for each $t \in T$. Then the contribution $\Delta(X, s)$ is always an integer from 0 to $|T|$. We maintain $|T| + 1$ doubly-linked lists, each of which contains source nodes with the same contribution. Each iteration of Algorithm 2 involves identifying and removing the node with the smallest contribution, and updating the lists.

We can find s^* with the smallest contribution in $O(|\Gamma(s^*)|)$ time by checking the lists from below, and remove it from the list in constant time. For each $t \in \Gamma(s^*)$, we compute the value of f_t in β time, and check whether it still exceeds the threshold θ_t. If it is lower than θ_t, we move the source nodes in $\Gamma(t)$ to the appropriate lists, which can be done in $O(|\Gamma(t)|)$ time. Note that for each node in T, this happens at most once.

Thus, the algorithm runs in $O(|S| + |T| + \beta|E|)$. Using a similar argument, when the weights are integers, the running time becomes $O(|S| + \beta|E| + W)$.

When c_s's are integers, we maintain a priority queue of the source nodes. Throughout the algorithm, we need $|S|$ insert operations, $|S|$ delete-min operations, and $|E|$ decrease-key operations. If we use a Fibonacci heap (Fredman & Tarjan, 1987), the algorithm runs in $O(|S| \log |S| + |T| + \beta|E|)$. □

D. Proof of Theorem 3 (in Section 3)

Proof. Suppose that f is monotone and submodular. It is easy to see that $\tilde{f}$ is monotone. Let us prove the submodularity of $\tilde{f}$. Let $X, Y \subseteq \tilde{S}$, and x, y be the corresponding vectors defined in the main text, respectively. Then $\tilde{f}(X) = f(x)$ and $\tilde{f}(Y) = f(y)$ hold by the definition of $\tilde{f}$. For each $s \in S$, $\max\{i \in \mathbb{Z}_+ \mid (s, i) \in X \cup Y\}$ is equal to the maximum of $x(s)$ and $y(s)$, and $\max\{i \in \mathbb{Z}_+ \mid (s, i) \in X \cap Y\}$ is not larger than the minimum of $x(s)$ and $y(s)$. The former fact implies $\tilde{f}(X \cup Y) = f(x \vee y)$, and the latter one implies $\tilde{f}(X \cap Y) \le f(x \wedge y)$ together with the monotonicity of f. Therefore, we have

$$\tilde{f}(X) + \tilde{f}(Y) = f(x) + f(y) \ge f(x \vee y) + f(x \wedge y) \ge \tilde{f}(X \cup Y) + \tilde{f}(X \cap Y).$$

Next, suppose that $\tilde{f}$ is monotone and submodular. For vectors $x, y \in \mathbb{Z}_+^S$, define $X = \{(s, i) \mid s \in S, i \le x(s)\}$ and $Y = \{(s, i) \mid s \in S, i \le y(s)\}$. Then $\tilde{f}(X) = f(x)$ and $\tilde{f}(Y) = f(y)$. Moreover, we have $\tilde{f}(X \cap Y) = f(x \wedge y)$ and $\tilde{f}(X \cup Y) = f(x \vee y)$. These relationships show

$$f(x) + f(y) = \tilde{f}(X) + \tilde{f}(Y) \ge \tilde{f}(X \cup Y) + \tilde{f}(X \cap Y) = f(x \vee y) + f(x \wedge y).$$

If $x \le y$, then $X \subseteq Y$ holds, and hence $f(x) \le f(y)$ follows from $\tilde{f}(X) \le \tilde{f}(Y)$. Therefore, f is monotone and submodular. □

E. Pseudo-Polynomial-Time $(e/(e-1))$-Approximation Algorithm for the Submodular Maximization over Integer Lattice (in Section 3)

Consider maximizing a monotone submodular function $f(x)$ over integer lattice subject to a budget constraint $\sum_{i \in S} c_i x_i \le B$ and an upper bound $x \le u$. From the given monotone submodular function $f \colon \mathbb{Z}_+^S \to \mathbb{R}$, construct a set function $\tilde{f} \colon 2^{\tilde{S}} \to \mathbb{R}$ as mentioned in the main text, where $\tilde{S} = \{(s, i) \mid s \in S, i \in \{0, 1, \ldots, u(s)\}\}$. Moreover, we regard a cost $c(s, i) = c_s i$ as the weight of $(s, i) \in \tilde{S}$. Then solve the submodular maximization problem with a budget constraint to find a solution $X \subseteq \tilde{S}$ that satisfies $\sum_{(s,i) \in X} c(s, i) \le B$ and maximizes $\tilde{f}(X)$. Suppose that $X \subseteq \tilde{S}$ is an α-approximate solution for the problem. Then, letting x be the corresponding vector, we have $f(x) = \tilde{f}(X)$ and $\sum_{s \in S} c(s, x_s) \le \sum_{(s,i) \in X} c(s, i) \le B$. Therefore, x is an α-approximate solution to the original instance. Since the reduced problem has an $(e/(e-1))$-approximation algorithm running in polynomial time, we obtain an $(e/(e-1))$-approximation algorithm that runs in pseudo-polynomial time.

F. Proof of Theorem 4 (in Section 4)

Proof. Let $X^* \subseteq S$ be an optimal solution for the problem. Choose an arbitrary node $s \in X^*$. Then, by the optimality of X^*, it holds that

$$d(X^*) = \frac{f(X^*)}{c(X^*)} \geq \frac{f(X^* \setminus \{s\})}{c(X^*) - c_s}.$$

By using the fact $f(X^* \setminus \{s\}) \geq f(X^*) - \sum_{t \in T(X^*) \cap \Gamma(s)} w_t$, this can be transformed to $(c(X^*) - c_s)d(X^*) \geq f(X^*) - \sum_{t \in T(X^*) \cap \Gamma(s)} w_t$. Since $f(X^*) = c(X^*)d(X^*)$, we have

$$d(X^*) \leq \Delta(X^*, s). \tag{F.1}$$

Consider the first iteration when some node $s^* \in X^*$ is removed by the algorithm. Let X denote the subset of nodes at this moment just before the removal. Clearly, we have $X^* \subseteq X$. Let us choose an arbitrary node $s \in X$. Then $\Delta(X, s) \geq \Delta(X, s^*)$ holds by the choice of s^*. Moreover, $\Delta(X, s^*) \geq \Delta(X^*, s^*)$ because $T(X) \supseteq T(X^*)$. Combining with (F.1), we have $\Delta(X, s) \geq \Delta(X, s^*) \geq \Delta(X^*, s^*) \geq d(X^*)$.

Therefore, we obtain

$$\begin{aligned} d(X) = \frac{1}{c(X)} \sum_{t \in T(X)} w_t &\geq \frac{1}{\gamma c(X)} \sum_{s \in X} \sum_{t \in T(X) \cap \Gamma(s)} w_t \\ &= \frac{1}{\gamma c(X)} \sum_{s \in X} c_s \Delta(X, s) \geq \frac{1}{\gamma c(X)} \sum_{s \in X} c_s d(X^*) = \frac{1}{\gamma} d(X^*). \end{aligned}$$

Thus the output is γ-approximation. □

G. Proof of Lemma 2 (in Section 4)

Proof. For $X \subseteq S$, we construct a solution $(\overline{x}, \overline{y})$ of (LP) as follows:

$$\overline{x}_s = \begin{cases} \frac{1}{|X|} & s \in X, \\ 0 & \text{otherwise}, \end{cases} \qquad \overline{y}_t = \begin{cases} \frac{1}{|X|} & t \in T(X), \\ 0 & \text{otherwise}. \end{cases}$$

The first constraints in (LP) are satisfied. Indeed, if $t \in T(X)$ (i.e., $\overline{y}_t = \frac{1}{|X|}$), then we have $|X \cap \Gamma(t)| \geq \theta_t$, which implies that for any $J \subseteq \Gamma(t)$ with $|J| = p_t$, there exists $s \in J$ that satisfies $s \in X$ (i.e., $\overline{x}_s = \frac{1}{|X|}$). The second constraint in (LP) is also satisfied because $\sum_{s \in S} x_s = \sum_{s \in X} x_s = 1$. Therefore, $(\overline{x}, \overline{y})$ is feasible for (LP). The objective value of $(\overline{x}, \overline{y})$ is

$$\sum_{t \in T} w_t \overline{y}_t = \sum_{t \in T(X)} w_t \overline{y}_t = \frac{1}{|X|} \sum_{t \in T(X)} w_t = d(X).$$

This completes the proof. □

H. Proof of Lemma 3 (in Section 4)

Proof. We begin by showing the existence of $r \geq 0$ that satisfies $d(X(r)) \geq \lambda/p$. Without loss of generality, we can assume that $\overline{y}_t = \min\{\sum_{s \in J} \overline{x}_s \mid J \subseteq \Gamma(t), |J| = p_t\}$ for each $t \in T$. We define a sequence of subsets $Y(r) = \{t \in T \mid \overline{y}_t \geq r\}$. If $t \in Y(r)$, then $\min\{\sum_{s \in J} \overline{x}_s \mid J \subseteq \Gamma(t), |J| = p_t\} \geq r$, which means that the p_tth smallest $\overline{x}_s$ is at least $r/p_t \geq r/p$. Hence at least $|\Gamma(t)| - p_t + 1 = \theta_t$ elements in $\Gamma(t)$ are contained in $X(r)$. Thus we see that $Y(r) \subseteq T(X(r))$.

Since for any $r \geq 0$,

$$\frac{1}{|X(r)|} \sum_{t \in T(X(r))} w_t \geq \frac{1}{|X(r)|} \sum_{t \in Y(r)} w_t,$$

it suffices to show that there exists $r \geq 0$ such that

$$\frac{1}{|X(r)|} \sum_{t \in Y(r)} w_t \geq \frac{\lambda}{p}.$$

Suppose not, that is, for any $r \geq 0$,

$$\sum_{t \in Y(r)} w_t < \frac{\lambda}{p} \cdot |X(r)|. \tag{H.2}$$

For each $t \in T$, define an indicator function $Z_t(r) : [0,1] \to \{0,1\}$ to be 1 if $r \leq \overline{y}_t$ and 0 otherwise. Integrating the left-hand side of (H.2) from 0 to 1, we have

$$\int_0^1 \left(\sum_{t \in Y(r)} w_t \right) dr = \int_0^1 \left(\sum_{t \in T} w_t Z_t(r) \right) dr = \sum_{t \in T} w_t \int_0^1 Z_t(r) dr = \sum_{t \in T} w_t \overline{y}_t = \lambda.$$

On the other hand, for each $s \in S$, define an indicator function $Z_s(r) : [0,p] \to \{0,1\}$ to be 1 if $r/p \leq \overline{x}_s$ and 0 otherwise. Integrating the right-hand side of (H.2) from 0 to 1, we have

$$\frac{\lambda}{p} \int_0^1 |X(r)| dr = \frac{\lambda}{p} \int_0^1 \left(\sum_{s \in S} Z_s(r) \right) dr = \frac{\lambda}{p} \sum_{s \in S} \int_0^1 Z_s(r) dr \leq \frac{\lambda}{p} \sum_{s \in S} \int_0^p Z_s(r) dr = \frac{\lambda}{p} \sum_{s \in S} p\overline{x}_s \leq \lambda,$$

which is a contradiction. Thus, we have the existence of $r \geq 0$ that satisfies $d(X(r)) \geq \lambda/p$.

From the definition of $X(r)$, we can enumerate all distinct sets of $X(r)$ by putting $r = p\overline{x}_s$ for all $s \in S$. This completes the proof. □

References

Fredman, M. L. and Tarjan, R. E. Fibonacci heaps and their uses in improved network optimization algorithms. *Journal of the ACM*, 34(3):596–615, 1987.