Yokohama Publishers
ISSN 1349-8169 ONLINE JOURNAL

Pacific Journal of Optimization
Vol. 15, No. 2, pp. 195 – 206, 2019

AN IMPROVED APPROXIMATION ALGORITHM FOR THE COVERING 0–1 INTEGER PROGRAM*

YOTARO TAKAZAWA, SHINJI MIZUNO AND TOMONARI KITAHARA

Abstract: We present an improved approximation algorithm for covering 0–1 integer programs (CIP), a well-known problem as a natural generalization of the set cover problem. Our algorithm uses a primal–dual algorithm for CIP by Fujito (2004) as a subroutine and achieves an approximation ratio of $\left(f - \frac{f-1}{m}\right)$ when $m \geq 2$, where m is the number of the constraints and f is the maximum number of non-zero entries in the constraints. In addition, when $m = 1$ our algorithm can be regarded as a PTAS. These results improve the previously known approximation ratio f.

Key words: *approximation algorithms, covering integer program, primal-dual method*

Mathematics Subject Classification: *90C27, 68W25*

1 Introduction

The covering 0–1 integer program (CIP) is a well-known problem which generalizes fundamental combinatorial optimization problems. CIP is formulated as follows.

$$\text{CIP}\left|\begin{array}{lll} \min & \sum_{j\in N} c_j x_j & \\ \text{s.t.} & \sum_{j\in N} u_{ij} x_j \geq d_i, & \forall i \in M, \\ & x_j \in \{0,1\}, & \forall j \in N, \end{array}\right. \tag{1.1}$$

where $M = \{1,\ldots,m\}$, $N = \{1,\ldots,n\}$, $c_j \geq 0$ $(j \in N)$, $u_{ij} \geq 0$ $(i \in M, j \in N)$ and $d_i \geq 0$ $(i \in M)$. For a given minimization problem having an optimal solution, an algorithm is called an α-approximation algorithm if it runs in polynomial time and produces a feasible solution whose objective value is less than or equal to α times the optimal value.

There is no $o(\log m)$-approximation algorithms for CIP unless P = NP since the set cover problem is a special case of CIP (Raz and Safra 1997). Kolliopoulos and Young (2013) present an $O(\log m)$-approximation algorithm for CIP. Let f be the maximum number of non-zero entries in the constraints. For any $f \geq 2$ and $\epsilon > 0$, CIP is hard to approximate better than a factor of $f - 1 - \epsilon$ unless P=NP (Dinur and Safra 2005) and $f - \epsilon$ under the unique games conjecture (Khot and Regev 2008). In this paper, we focus on algorithms whose approximation ratios depend on f.

*This research is supported in part by Grant-in-Aid for Science Research (A) 26242027 and 19H00808 of Japan Society for the Promotion of Science and Grant-in-Aid for Young Scientist (B) 15K15941.

For CIP, f-approximation algorithms are proposed in (Carr et al. 2000; Fujito 2004; Fujito and Yabuta 2004). Takazawa and Mizuno (2017) present an f_2-appoximation algorithm for CIP, where f_2 is the second largest number of non-zero entries in the constraints. Approximation algorithms for generalizations of CIP are also well studied. Koufogiannakis and Young (2005) and Pritchard and Chakrabarty (2011) give f-approximation algorithms for CIP with general upper bounds on variables. McCormick et al. (2016) develop an approximation algorithm for precedence constrained CIP. Takazawa et al. (2017) present an approximation algorithm for the partial CIP, where some constraints may not be satisfied. To the best of our knowledge, there are no approximation algorithms for CIP whose approximation ratio is strictly less than f when $f \geq 2$.

Table 1: Approximation ratios for problems related to CIP

Problem Names	Restrictions on CIP	Approximation Ratios
Covering 0–1 integer program (CIP)	-	f (Carr et al. 2000; Fujito 2004; Fujito and Yabuta 2004; Koufogiannakis and Young 2005; Pritchard and Chakrabarty 2011) f_2 (Takazawa and Mizuno 2017) $f - \frac{f-1}{m}$ (**this paper**)
Minimum Knapsack Problem (MKP)	$m = 1$	FPTAS (Babat 1975)
(Weighted) Set Cover Problem (SCP)	$u_{ij} \in \{0,1\},\ d_i = 1$	$f\left(1 - (f-1)\frac{\ln\ln\Delta}{\ln\Delta}\right)$ (Halperin 2002)
Vertex Cover Problem (VCP)	$u_{ij} \in \{0,1\},\ d_i = 1$ $c_j = 1,\ f = 2$	$2 - \frac{\ln\ln\Delta}{\ln\Delta}$ (Halperin 2002) $2 - \Theta(1/\sqrt{\log m})$ (Karakostas 2009)

where $i^* \in M$ is an index such that $|\{j \in N \mid u_{i^*j} > 0\}| = f$ and

$$\begin{aligned} f &= \max_{i\in M} |\{j \in N \mid u_{ij} > 0\}|, \\ f_2 &= \max_{i\in M\setminus i^*} |\{j \in N \mid u_{ij} > 0\}|, \\ \Delta &= \max_{i\in N} \textstyle\sum_{i\in M} u_{ij}. \end{aligned} \tag{1.2}$$

While there are no approximation algorithms for CIP whose approximation ratio is better than f, there are such approximation algorithms for special cases of CIP such as the minimum knapsack problem (MKP), the set cover problem (SCP) and the vertex cover problem (VCP), see Table 1. MKP is a special case of CIP when the number of the constraints is only one and can be regarded as the minimization version of the knapsack problem (KP). It is well-known that KP and MKP admit a fully polynomial time approximation scheme (FPTAS) (Babat 1975; Kellerer et al. 2004). SCP is a special case of CIP when $u_{ij} \in \{0,1\}$ and $d_i = 1$ and VCP is a special case of SCP when $f = 2$ and $c_j = 1$. Halperin (2002) gives a $(2 - \ln\ln\Delta/\ln\Delta)$-approximation algorithm for VCP, where Δ is the maximal degree of the graph, and extends this result to SCP. Karakostas (2009) obtains a $(2 - \Theta(1/\sqrt{\log m}))$-approximation algorithm for VCP. Both the approaches use SDP-relaxation. Fujito (2004) gives an approximation algorithm whose approximation ratio is better than f for a more general problem where $u_{ij} \in \{0,1\}$ and d_i is a postive integer. For more information on approximation algorithms for CIP and its special cases, we refer the reader to Fujito (2004).

Contribution

In this paper we propose an $\left(f - \frac{f-1}{m}\right)$-approximation algorithm for CIP when $m \geq 2$. When $m = 1$ (MKP), for any fixed $\epsilon > 0$, our algorithm finds a solution whose objective value is less than or equal to $(1 + \epsilon)$ times the optimal value within polynomial in m, n and $n^{\lceil 1/\epsilon \rceil}$, that is, it can be regarded as a polynomial-time approximation scheme (PTAS). Our algorithm is the first algorithm for CIP whose approximation ratio is strictly less than f when $f \geq 2$.

Our algorithm, which we call the main algorithm, uses a primal-dual f-approximation algorithm (PD) for CIP by Fujito (2004). The main algorithm solves $O(n^2)$ subproblems of CIP by using PD as a subroutine. Similar approaches can be seen in literature of the partial covering problems (Gandhi et al. 2004; Könemann et al. 2011; Takazawa et al. 2017). Note that analysis of the main algorithm is a non-trivial work even though the algorithm uses PD presented by Fujito.

Notation and Assumption

Let $I = (\boldsymbol{U}, \boldsymbol{d}, \boldsymbol{c})$ be a data of (1.1), where $\boldsymbol{U}$ is the matrix of u_{ij}. We call I an instance of CIP. Let CIP(I) be the problem for instance I. Let I_0 be an instance to be solved. Without loss of generality, for the problem CIP(I_0), we assume that

- $f \geq 2$,
- CIP(I_0) is feasible.

Let $OPT(I)$ be the optimal value of CIP(I) when CIP(I) is feasible. For any subset $S \subseteq N$, we define a vector $\boldsymbol{x}(S) \in \mathbb{R}^n$ as follows:

$$x_j(S) = \begin{cases} 1 & \text{if } j \in S \\ 0 & \text{if } j \notin S \end{cases} \quad \text{for any } j \in N. \tag{1.3}$$

2 Main Algorithm

In this section, we introduce the main algorithm. The main algorithm solves many subproblems of CIP with the algorithm PD in Fujito (2004) as a subroutine. The algorithm PD and its analysis are presented in Section 3. This section is organized as follows:

1. We show a property (Lemma 2.1) of the solution generated by PD.
2. We explain that we can get an $\left(f - \frac{f-1}{m}\right)$-approximation solution by using PD if we know partial information about an optimal solution.
3. We introduce the main algorithm which gives an $\left(f - \frac{f-1}{m}\right)$-approximation without information about an optimal solution.

First, we state a property of the solution generated by PD. Details of PD and the proof of Lemma 2.1 are shown in Section 3.

Lemma 2.1. *For any instance I, the algorithm PD presented in Section 3 runs in $O(mn^2)$ and determines feasibility of CIP(I). If CIP(I) is feasible, the algorithm PD produces a* *feasible solution* $\boldsymbol{x}$ *satisfying*

$$\sum_{j\in N} c_j x_j \le \left(f - \frac{f-1}{m}\right) OPT(I) + c_{\max},$$

where $c_{\max} = \max_{j\in N} c_j$.

For any $A \subseteq N$, define

$$\begin{array}{rcl} N(A) & = & \{j \in N\backslash A \mid c_j \le \min_{j'\in A} c_{j'}\}, \\ \bar{N}(A) & = & \{j \in N\backslash A \mid c_j > \min_{j'\in A} c_{j'}\}. \end{array} \tag{2.1}$$

We can see that $\{A,\ N(A),\ \bar{N}(A)\}$ is a partition of N into elements A, elements $N(A)$ that are cheaper than all items in A, and elements $\bar{N}(A)$. For an instance I_0 and $A \subseteq N$, we consider a subproblem of CIP(I_0) by fixing some variables as follows:

$$\begin{array}{ll} x_j = 1 & \text{if } j \in A, \\ x_j = 0 & \text{if } j \in \bar{N}(A). \end{array}$$

This subproblem can be expressed as:

$$\begin{array}{lll} \min & \displaystyle\sum_{j\in N(A)} c_j x_j & \\ \text{s.t.} & \displaystyle\sum_{j\in N(A)} u_{ij} x_j \ge \max\left\{ d_i - \sum_{j\in A} u_{ij},\ 0\right\}, & \forall i \in M, \\ & x_j \in \{0,1\}, & \forall j \in N(A). \end{array} \tag{2.2}$$

Note that this problem is also CIP and the number of decision variables is $|N(A)|$. Let $I_0(A)$ be the instance of this subproblem. Let $\bar{\boldsymbol{x}}^A \subseteq \{0,1\}^{|N(A)|}$ be an output by the algorithm PD for the subproblem CIP($I_0(A)$) when CIP($I_0(A)$) is feasible. Define a solution $\boldsymbol{x}^A \subseteq \{0,1\}^n$ for CIP(I_0) as

$$\boldsymbol{x}^A = \begin{cases} \bar{x}_j^A & \text{if } j \in N(A), \\ 1 & \text{if } j \in A, \\ 0 & \text{if } j \in \bar{N}(A). \end{cases} \tag{2.3}$$

Let S^* be a subset of N such that $\boldsymbol{x}(S^*)$ is an optimal solution of CIP(I_0). For any positive integer k such that $k \le |S^*|$, we define $A_k^* \subseteq S^*$ as $\{j_1, \ldots, j_k\}$ such that $\{c_{j_1}, \ldots, c_{j_k}\}$ is a set of the k largest numbers in $\{c_j \mid j \in S^*\}$. The following lemma shows that we can get an $\left(f - \frac{f-1}{m}\right)$-approximation solution for CIP(I_0) by using PD if we know A_2^*.

Lemma 2.2. *If* $m \ge 2$, $k = 2$ *and* $k < |S^*|$, *then* $\boldsymbol{x}^{A_k^*}$ *defined by (2.3) is feasible to CIP(I_0) and the following inequality holds:*

$$\sum_{j\in N} c_j x_j^{A_k^*} \le \left(f - \frac{f-1}{m}\right) OPT(I_0).$$

Proof. First, we will prove that $\boldsymbol{x}^{A_k^*}$ is feasible to CIP(I_0). The problem CIP($I_0\,(A_k^*)$) is feasible since a subvector of $\boldsymbol{x}\,(S^*\backslash A_k^*)$ is a feasible solution for CIP($I_0\,(A_k^*)$). Thus, PD outputs a feasible solution $\bar{\boldsymbol{x}}^{A_k^*}$ for CIP($I_0\,(A_k^*)$). Then, a solution $\boldsymbol{x}^{A_k^*}$ defined by (2.3) is clearly feasible to CIP(I_0).

Let $\alpha = f - \frac{f-1}{m}$. Let f' be the maximum number of non-zero entries in the constraints of the subproblem CIP$(I_0\left(A_k^*\right))$. we can see that $f' - \frac{f'-1}{m} \leq \alpha$ from $f' \leq f$. From Lemma 1, the algorithm PD outputs a solution $\bar{\boldsymbol{x}}^{A_k^*}$ for the problem CIP$(I_0\left(A_k^*\right))$ and the next inequality holds:

$$\sum_{j \in N\left(A_k^*\right)} c_j \bar{x}_j^{A_k^*} \leq \left(f' - \frac{f'-1}{m}\right) OPT\left(I_0\left(A_k^*\right)\right) + \max_{j \in N\left(A_k^*\right)} c_j \leq \alpha OPT\left(I_0\left(A_k^*\right)\right) + \max_{j \in N\left(A_k^*\right)} c_j. \tag{2.4}$$

From the definition of the subproblem CIP$(I_0\left(A_k^*\right))$, we have that

$$OPT(I_0) = \sum_{j \in S^* \setminus A_k^*} c_j + \sum_{j \in A_k^*} c_j = OPT\left(I_0\left(A_k^*\right)\right) + \sum_{j \in A_k^*} c_j. \tag{2.5}$$

From the definition of $N\left(A_k^*\right)$ in (2.1), we obtain that

$$\max_{j \in N\left(A_k^*\right)} c_j \leq \min_{j \in A_k^*} c_j \leq \frac{1}{k} \sum_{j \in A_k^*} c_j. \tag{2.6}$$

From (2.4), (2.5) and (2.6), we have that

$$\begin{aligned}
\sum_{j \in N} c_j x_j^{A_k^*} &= \sum_{j \in N\left(A_k^*\right)} c_j \bar{x}_j^{A_k^*} + \sum_{j \in A_k^*} c_j \\
&\leq \alpha OPT(I_0(A_k^*)) + \max_{j \in N\left(A_k^*\right)} c_j + \sum_{j \in A_k^*} c_j \\
&= \alpha \left(OPT(I_0(A_k^*)) + \sum_{j \in A_k^*} c_j \right) + (1-\alpha) \sum_{j \in A_k^*} c_j + \max_{j \in N\left(A_k^*\right)} c_j \\
&\leq \alpha OPT(I_0) + \left(1 - \alpha + \frac{1}{k}\right) \sum_{j \in A_k^*} c_j
\end{aligned} \tag{2.7}$$

Since $m \geq 2$ and $f \geq 2$ from the assumptions, $\alpha = f - \frac{f-1}{m} \geq 3/2$ holds. From this observation and $k = 2$, we obtain that

$$\left(1 - \alpha + \frac{1}{k}\right) \sum_{j \in A_k^*} c_j \leq \left(1 - \frac{3}{2} + \frac{1}{2}\right) \sum_{j \in A_k^*} c_j = 0.$$

Therefore, we get

$$\sum_{j \in N} c_j x_j^{A_k^*} \leq \alpha OPT(I_0).$$

□

The following lemma shows that when $m = 1$, we can get a $\left(1 + \frac{1}{k}\right)$-approximation solution for any fixed positive integer k such that $k < |S^*|$.

Lemma 2.3. *If $m = 1$, for any positive integer k such that $k < |S^*|$, $\boldsymbol{x}^{A_k^*}$ is feasible to CIP(I) and the following inequality holds:*

$$\sum_{j \in N} c_j x_j^{A_k^*} \leq \left(1 + \frac{1}{k}\right) OPT(I_0).$$

Proof. The proof of Lemma 2.2 until (2.7) also holds in this case. By substituting $\alpha = f - \frac{f-1}{m} = 1$ in (2.7), we have that

$$\sum_{j \in N} c_j x_j^{A_k^*} \leq OPT(I_0) + \frac{1}{k} \sum_{j \in A_k^*} c_j \leq \left(1 + \frac{1}{k}\right) OPT(I_0),$$

where the last inequality holds since $\sum_{j \in A_k^*} c_j \leq OPT(I_0)$ from $A_k^* \subseteq S^*$. □

Even though Lemma 2.2 and Lemma 2.3 require the information about A_k^*, we don't need it in advance if we execute PD for all supproblems CIP($I_0(A)$) such that $A \subseteq N$ and $|A| \leq k$. The main algorithm is presented as follows:

The main algorithm

Input: $I_0 = (\boldsymbol{U}, \boldsymbol{d}, \boldsymbol{c})$ and a positive integer k.

Step 0: Calculate $\mathcal{D}(k)$ defined by

$$\mathcal{D}(k) = \{A \subseteq N \mid |A| \leq k \mid\}.$$

Step 1: For each $A \in \mathcal{D}(k)$, do the following process:

Let $I_0(A)$ be the data derived from the subproblem (2.2). If $\boldsymbol{x}(A)$ is feasible to CIP(I_0), that is $\sum_{j \in A} u_{ij} \geq d_i$ for any $i \in M$, go to Step 1-A. Otherwise go to Step 1-B.

(**Step 1-A**): Make a solution for CIP(I_0) as follows:

$$\boldsymbol{x}^A = \boldsymbol{x}(A).$$

(**Step 1-B**): Execute the algorithm PD for the subproblem CIP($I_0(A)$) defined by (2.2). If the problem CIP($I_0(A)$) is feasible, the algorithm outputs a feasible solution for CIP($I_0(A)$). Denote this solution by $\bar{\boldsymbol{x}}^A \subseteq \{0,1\}^{|N(A)|}$. By using this solution, make a solution $\boldsymbol{x}^A$ for CIP(I_0) by (2.3):

$$\boldsymbol{x}^A = \begin{cases} \bar{x}_j^A & \text{if } j \in N(A), \\ 1 & \text{if } j \in A, \\ 0 & \text{if } j \in \bar{N}(A). \end{cases}$$

Step 2: Set $\hat{A} = \underset{A \subseteq \mathcal{D}(k)}{\arg\min} \sum_{j \in N} c_j x_j^A$ and output $\boldsymbol{x}^{\hat{A}}$.

Theorem 2.4. *Suppose $m \geq 2$ and set $k = 2$ in the main algorithm. Then the main algorithm is an $\left(f - \frac{f-1}{m}\right)$-approximation algorithm for CIP.*

Proof. The running time of one iteration of Step 1 is $O(mn^2)$ from Lemma 2.1. The number of iterations of the main algorithm is $O(n^k)$. Thus, the main algorithm runs in polynomial time.

If $k \geq |S^*|$, that implies $S^* \in \mathcal{D}(k)$. We consider when Step 1 is executed for $A = S^*$. In this case, the algorithm goes to Step 1-A since $\boldsymbol{x}(S^*) = \boldsymbol{x}^*$ is feasible to CIP(I_0) and sets $\boldsymbol{x}^{S^*} = \boldsymbol{x}^*$ at this iteration. Thus, the algorithm outputs an optimal solution.

Next we consider the case when $k < |S^*|$. In this case, $A_k^* \in \mathcal{D}(k)$ holds and that implies the main algorithm executes Step 1-B for the set A_k^* since $\boldsymbol{x}(A_k^*)$ is infeasible to CIP(I_0). From Lemma 2.2, we have that

$$\sum_{j \in N} c_j x_j^{A_k^*} \leq \left(f - \frac{f-1}{m}\right) OPT(I_0).$$

Therefore, we get

$$\sum_{j\in N} c_j x_j^{\hat{A}} = \min_{A\subseteq \mathcal{D}(k)} \sum_{j\in N} c_j x_j^{A} \le \sum_{j\in N} c_j x_j^{A_k^*} \le \left(f - \frac{f-1}{m}\right) OPT(I_0).$$

□

When $m = 1$, we have the following result from Lemma 2.3 in the same way as the proof of Theorem 2.4.

Theorem 2.5. *For any fixed $\epsilon > 0$, set $k = \lceil 1/\epsilon \rceil$ in the main algorithm. If $m = 1$, then the main algorithm finds a feasible solution whose objective value is less than or equal to $(1+\epsilon)$ times the optimal value within polynomial in m, n and $n^{\lceil 1/\epsilon \rceil}$.*

3 Algorithm PD and Proof of Lemma 2.1

In this section, we introduce the algorithm PD proposed by Fujito (2004) and prove Lemma 2.1. First we show a relaxation problem of CIP, which is utilized by PD. Let

$$\begin{aligned} d_i(A) &= \max\{0, d_i - \textstyle\sum_{j\in A} u_{ij}\}, \ \forall i \in M, \forall A \subseteq N, \\ u_{ij}(A) &= \min\{u_{ij}, d_i(A)\}, \ \forall i \in M, \forall A \subseteq N, \forall j \in N\backslash A, \\ M(A) &= \{i \in M \mid d_i(A) > 0\}, \ \forall A \subseteq N, \\ U_j(A) &= \sum_{i\in M(A)} \frac{u_{ij}(A)}{d_i(A)}, \ \forall A \subseteq N, \forall j \in N\backslash A \end{aligned} \tag{3.1}$$

Using these symbols, we have the following problem.

$$\begin{aligned} \min \quad & \sum_{j\in N} c_j x_j \\ \text{s.t.} \quad & \sum_{j\in N\backslash A} U_j(A) x_j \ge |M(A)|, \quad \forall A \subseteq N \\ & x_j \ge 0, \qquad\qquad\qquad\quad \forall j \in N. \end{aligned} \tag{3.2}$$

This problem is a relaxation problem of CIP from Proposition 1 in Fujito (2004).

Lemma 3.1 (Fujito, 2004)**.** *(3.2) is a relaxation problem of CIP, that is, any feasible solution $\boldsymbol{x}$ for CIP is feasible to (3.2).*

The dual problem of (3.2) is expressed as

$$\begin{aligned} \max \quad & \sum_{A\subseteq N} |M(A)| y(A) \\ \text{s.t.} \quad & \sum_{A\subseteq N : j\notin A} U_j(A) y(A) \le c_j, \quad \forall j \in N, \\ & y(A) \ge 0, \qquad\qquad\qquad \forall A \subseteq N. \end{aligned} \tag{3.3}$$

Now, we introduce a useful and well-known result in analysis of the primal-dual method. We provide a proof for the sake of completeness.

Lemma 3.2. *Let $\boldsymbol{x} \in \{0,1\}^n$ and let $\boldsymbol{y}$ be a feasible solution to (3.3). For $\alpha \ge 0$, if $\boldsymbol{x}$ and $\boldsymbol{y}$ satisfy*

(a) $\forall j \in N,\ x_j = 1 \Rightarrow \sum_{A \subseteq N : j \notin A} U_j(A) y(A) = c_j$,

(b) $\forall A \subseteq N,\ y(A) > 0 \Rightarrow \sum_{j \in N \setminus A} U_j(A) x_j \leq \alpha |M(A)|$,

then the following inequality holds:

$$\sum_{j \in N} c_j x_j \leq \alpha OPT(I).$$

Proof. Suppose $\boldsymbol{x}$ and $\boldsymbol{y}$ satisfy the conditions of Lemma 3.2. Let $S = \{j \in N \mid x_j = 1\}$. From the condition (a), we have that

$$\sum_{j \in N} c_j x_j = \sum_{j \in S} c_j = \sum_{j \in S} \sum_{A \subseteq N : j \notin A} U_j(A) y(A) = \sum_{A \subseteq N} \sum_{j \in S \setminus A} U_j(A) y(A).$$

From the condition (b), we obtain that

$$\sum_{A \subseteq N} \sum_{j \in S \setminus A} U_j(A) y(A) = \sum_{A \subseteq N} y(A) \sum_{j \in S \setminus A} U_j(A) \leq \alpha \sum_{A \subseteq N} |M(A)| y(A).$$

Since $\boldsymbol{y}$ is feasible to (3.3), the objective value of $\boldsymbol{y}$ is less than or equal to the optimal value of (3.2), which is less than or equal to OPT(I). Therefore, we get

$$\sum_{j \in N} c_j x_j \leq \alpha \sum_{A \subseteq N} |M(A)| y(A) \leq \alpha OPT(I).$$

□

The algorithm PD is presented below. Solutions generated by the algorithm, except for the final solution, satisfy all the conditions in Lemma 3.2 for $\alpha = f - \frac{f-1}{m}$.

Algorithm PD

Input: An instance I.

Step 0: Set $\boldsymbol{x} = \boldsymbol{0}$, $\boldsymbol{y} = \boldsymbol{0}$ and $\bar{\boldsymbol{c}} = \boldsymbol{c}$. Check whether the solution $(1, \ldots, 1)$ is feasible to CIP(I) or not. If it is not feasible, declare INFEASIBLE and stop.

Step 1: Let

$$\begin{aligned} S &= \{j \in N \mid x_j = 1\}, \\ d_i(S) &= \max\{0, d_i - \textstyle\sum_{j \in S} u_{ij}\},\ \forall i \in M, \\ u_{ij}(S) &= \min\{u_{ij}, d_i(S)\},\ \forall i \in M, \forall j \in N \setminus S, \\ M(S) &= \{i \in M \mid d_i(S) > 0\}, \\ U_j(S) &= \textstyle\sum_{i \in M(S)} \frac{u_{ij}(S)}{d_i(S)},\ \forall j \in N \setminus S, \\ N_{>0}(S) &= \{j \in N \setminus S \mid U_j(S) > 0\}. \end{aligned}$$

If $M(S) = 0$, output $\boldsymbol{x}$, $\boldsymbol{y}$ and stop. Otherwise, go to Step 2.

Step 2: Increase $y(S)$ as much as possible while maintaining dual feasibility for (3.3). That is, set

$$y(S) = \frac{\bar{c}_t}{U_t(S)},$$

where

$$t = \arg \min_{j \in N_{>0}(S)} \frac{\bar{c}_j}{U_j(S)}.$$

Set $\bar{c}_j := \bar{c}_j - U_j(S) y(S)$ for all $j \in N_{>0}(S)$. Update $x_t = 1$. Go back to Step 1.

Fujito (2004) shows that the algorithm PD is an f-approximation algorithm for CIP since we can easily show that the output of PD satisfies the conditions in Lemma 3.2 for $\alpha = f$. In this study, we show that solutions produced by PD satisfies the stronger conditions in Lemma 3.2.

Lemma 3.3. *Let $\boldsymbol{x}$ be the output by PD and x_ℓ be the variable which becomes 1 from 0 at the last iteration of PD. Let $\tilde{\boldsymbol{x}}$ be the solution obtained by setting $x_\ell = 0$ in $\boldsymbol{x}$. Let $\tilde{\boldsymbol{y}}$ be the dual solution at the end of the iteration before x_ℓ becomes 1. Then $\tilde{\boldsymbol{x}}$ and $\tilde{\boldsymbol{y}}$ satisfy the conditions in Lemma 3.2 for $\alpha = f - \frac{f-1}{m}$.*

Proof. Let $\tilde{S} = \{j \in N \mid \tilde{x}_j = 1\}$. $\tilde{\boldsymbol{y}}$ is feasible to the dual (3.3) since PD starts from the dual feasible solution $\boldsymbol{y} = 0$ and maintains dual feasibility at every iteration. Note that $\tilde{\boldsymbol{x}}$ is infeasible to (1.1). $\tilde{\boldsymbol{x}}$ and $\tilde{\boldsymbol{y}}$ satisfies (a) in Lemma 3.2 by the way the algorithm updates $\boldsymbol{x}$ and $\boldsymbol{y}$. Therefore it suffices to show that (b) in Lemma 3.2 holds, that is, for any $A \subseteq N$ such that $\tilde{y}(A) > 0$, the following holds:

$$\sum_{j \in N \setminus A} U_j(A)\tilde{x}_j = \sum_{j \in \tilde{S} \setminus A} U_j(A) = \sum_{i \in M(A)} \sum_{j \in \tilde{S} \setminus A} \frac{u_{ij}(A)}{d_i(A)} \leq \left(f - \frac{f-1}{m}\right) |M(A)|,$$

where we use the definition of $U_j(A)$ by (3.1).

Now, we fix $A \subseteq N$ such that $\tilde{y}(A) > 0$. From Step 2, $\tilde{y}(A) > 0$ implies

$$A \subseteq \tilde{S}. \tag{3.4}$$

From (3.4) and the definition of $M(A)$ by (3.1), we have that

$$M(\tilde{S}) \subseteq M(A), \tag{3.5}$$

and for any $i \in M(\tilde{S})$

$$\sum_{j \in \tilde{S}} u_{ij} < d_i. \tag{3.6}$$

From (3.1), (3.4) and (3.6), for any $i \in M(\tilde{S})$, we obtain that

$$\sum_{j \in \tilde{S} \setminus A} u_{ij}(A) \leq \sum_{j \in \tilde{S} \setminus A} u_{ij} = \sum_{j \in \tilde{S}} u_{ij} - \sum_{j \in A} u_{ij} < d_i - \sum_{j \in A} u_{ij} \leq d_i(A).$$

Note that $d_i(A) > 0$ for any $i \in M(A)$. By dividing both sides by $d_i(A)$ and taking sum of $i \in M(\tilde{S})$, we get

$$\sum_{i \in M(A) \cap M(\tilde{S})} \sum_{j \in \tilde{S} \setminus A} \frac{u_{ij}(A)}{d_i(A)} < |M(\tilde{S})|, \tag{3.7}$$

where we use $M(\tilde{S}) = M(A) \cap M(\tilde{S})$ from (3.5). From the definition of f and $u_{ij}(A) \leq d_i(A)$, we have that for any $i \in M(A)$,

$$\sum_{j \in \tilde{S} \setminus A} u_{ij}(A) \leq \sum_{j \in \tilde{S}} u_{ij}(A) \leq f d_i(A).$$

By dividing both sides by $d_i(A)$ and taking sum of $i \in M(A) \setminus M(\tilde{S})$, we get

$$\sum_{i \in M(A) \setminus M(\tilde{S})} \sum_{j \in \tilde{S} \setminus A} \frac{u_{ij}(A)}{d_i(A)} \leq f(|M(A)| - |M(\tilde{S})|). \tag{3.8}$$

From (3.7) and (3.8), we obtain that

$$\begin{aligned}\sum_{j\in\tilde{S}\setminus A} U_j(A) &= \sum_{i\in M(A)}\sum_{j\in\tilde{S}\setminus A}\frac{u_{ij}(A)}{d_i(A)}\\ &= \sum_{i\in M(A)\cap M(\tilde{S})}\sum_{j\in\tilde{S}\setminus A}\frac{u_{ij}(A)}{d_i(A)} + \sum_{i\in M(A)\setminus M(\tilde{S})}\sum_{j\in\tilde{S}\setminus A}\frac{u_{ij}(A)}{d_i(A)}\\ &< |M(\tilde{S})| + f(|M(A)| - |M(\tilde{S})|)\\ &= (1-f)|M(\tilde{S})| + f|M(A)|.\end{aligned}$$

Since $\tilde{\boldsymbol{x}}$ is infeasible, $1 \leq |M(\tilde{S})|$ holds. Also, $1 \leq M(A) \leq m$ holds. From $f \geq 2$, we finally obtain that

$$\begin{aligned}\sum_{j\in\tilde{S}\setminus A} U_j(A) &\leq (1-f)|M(\tilde{S})| + f|M(A)|\\ &\leq 1-f+f|M(A)|\\ &= \left(f - \frac{f-1}{|M(A)|}\right)|M(A)|\\ &\leq \left(f - \frac{f-1}{m}\right)|M(A)|.\end{aligned}$$

□

Now we can easily prove Lemma 2.1.

Proof of Lemma 2.1. From Lemma 3.3, we have that

$$\sum_{j\in N} c_j x_j = \sum_{j\in N} c_j\tilde{x}_j + c_\ell \leq \left(f - \frac{f-1}{m}\right) OPT(I) + c_{\max}.$$

Fujito (2004) shows that the algorithm PD runs in $O(mn^2)$ time. □

Acknowledgment

We thank the anonymous referees for their helpful comments and suggestions.

References

[1] LG. Babat, Linear function on the n-dimentional unit cube, *Dokl Akad Nauk* SSSR221 (1975) 761–762 (in Russian).

[2] R.D. Carr, L. Fleischer, V.J. Leung and C.A. Phillips, Strengthening integrality gaps for capacitated network design and covering problems, in: *Proceedings of the 11th Annual ACM-SIAM Symposium on Discrete Algorithms*, 2000, pp 106–115

[3] I. Dinur and S. Safra, On the hardness of approximating minimum vertex cover, *Ann. Math.* 162 (2005) 439–485.

[4] T. Fujito, *On combinatorial approximation of covering 0–1 integer programs and partial set cover*, *J. Comb. Optim.* 8 (2004) 439–452.

[5] T. Fujito and T. Yabuta, Submodular integer cover and its application to production planning, in: *Proceedings of of the Second international conference on Approximation and Online Algorithms*, 2004, pp 154–166.

[6] R. Gandhi, S. Khuller and A. Srinivasan, Approximation algorithms for partial covering problems, *J. Algorithms* 53 (2004), 55–84.

[7] E. Halperin, Improved approximation algorithms for the vertex cover problem in graphs and hypergraphs, *SIAM J. Comput.* 31 (2002) 1608–162.3

[8] G. Karakostas, A better approximation ratio for the vertex cover problem, in: *Proceedings of the 32nd International Colloquium on Automata, Languages and Programming*, 2009, pp 1043–1050

[9] H. Kellerer, U. Pferschy and D. Pisingerm Knapsack Problems, Springer-Verlag, Berlin, 2004.

[10] S. Khot S and O. Regev, Vertex cover might be hard to approximate to within 2-ϵ, *J. Comput. System Sci.* 74 (2008) 335–349.

[11] SG. Kolliopoulos and N E. Young, Approximation algorithms for covering/packing integer programs, *J. Comput. System Sci.* 71 (2005) 495–505.

[12] J. Könemann, O. Parekh and D. Segev, A unified approach to approximating partial covering problems, *Algorithmica* 59 (2011) 489–509.

[13] D. Koufogiannakis and N.E. Young, Greedy δ-approximation algorithm for covering with arbitrary constraints and submodular cost, *Algorithmica* 66 (2013) 113–152.

[14] S.T. McCormick, B. Peis, J. Verschae and A. Wierz, Primal-dual algorithms for precedence constrained covering problems, *Algorithmica* 78 (2016) 771–787.

[15] D. Pritchard and D. Chakrabarty, Approximability of sparse integer programs, *Algorithmica* 61 (2011) 75–93.

[16] R. Raz and S. Safra, A sub-constant error-probability low-degree test and a sub-constant error-probability PCP characterization of NP, in: *Proceedings of the 29th Annual ACM Symposium on Theory of Computing*, 1997, pp 475–484.

[17] Y. Takazawa and S. Mizuno, A 2-approximation algorithm for the minimum knapsack problem with a forcing graph, *J. Oper. Res. Soc. Jpn.* 60 (2017) 15–23.

[18] Y. Takazawa, S. Mizuno and T. Kitahara, An approximation algorithm for the partial covering 0–1 integer program, *Discrete Appl. Math.* (2018), (in press).

Manuscript received 29 April 2018
revised 13 December 2018
accepted for publication 14 December 2018

Yotaro Takazawa
Department of Industrial Engineering and Economics
Tokyo Institute of Technology
2-12-1 Ohokayama
Meguro-ku Tokyo 152-8552, Japan
E-mail address: takazawa.y.ab@m.titech.ac.jp

Shinji Mizuno
Department of Industrial Engineering and
Economics
Tokyo Institute of Technology
2-12-1 Ohokayama
Meguro-ku Tokyo 152-8552, Japan
E-mail address: mizuno.s.ab@m.titech.ac.jp

Tomonari Kitahara
Department of Industrial Engineering and
Economics
Tokyo Institute of Technology
2-12-1 Ohokayama
Meguro-ku Tokyo 152-8552, Japan
E-mail address: kitahara.t.ab@m.titech.ac.jp