Complexity of computing an arbitrary lifted cover inequality

Wei-Kun Chen · Yu-Hong Dai

November 27, 2018

Abstract The well-known lifted cover inequality is widely employed in solving mixed integer programs. However, it is still an open question whether an arbitrary project lifted cover inequality can be computed in polynomial time for a given minimal cover (Gu, Nemhauser, and Savelsbergh, INFORMS J. Comput., 26: 117–123, 1999). We show that this problem is $\mathcal{NP}$-hard, thus giving a negative answer to the question.

Keywords Integer programming · Lifted cover inequality · Complexity · Lifting problem

Mathematics Subject Classification (2000) 90C11 · 90C27

1 Introduction

The lifted cover inequality (LCI) is a well-known cutting plane for mixed integer programs. The LCI was first studied in [2, 5, 17]. Its effectiveness on using as a cutting plane was demonstrated in [4], see also [7, 10, 11, 16, 18]. Given a cover inequality, in order to obtain an LCI, we need to lift variables one at a time sequentially. To lift each variable, a knapsack problem is required to be solved to compute the lifting coefficient. Under certain conditions, the LCI can be computed in polynomial time, see [5, 14, 19]. In general, however, the complexity of computing an arbitrary LCI is still unknown. This was explicitly mentioned in [6] as an open question.

"We show that this dynamic programming algorithm may take exponential time to compute a sequential LCI that is not simple. It is still an open question

This work was supported by the Chinese Natural Science Foundation (Nos. 11631013, 11331012) and the National 973 Program of China (No. 2015CB856002).

LSEC, ICMSEC, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing, China · School of Mathematical Sciences, University of Chinese Academy of Sciences, Beijing, China. Emails: cwk@lsec.cc.ac.cn, dyh@lsec.cc.ac.cn
whether an arbitrary LCI can be computed in polynomial time for a given minimal cover $C$.

The above question was also cited as an open question in [1, 5, 8, 15]. We will give a negative answer to the question by showing that the problem of computing an arbitrary LCI is \textsc{NP}-hard. Thus, unless $P = \textsc{NP}$, there exists no polynomial time algorithm to computing an arbitrary LCI.

This paper is organized as follows. In Section 2, we review how to compute an LCI. In Section 3, we describe the elegant example by Gu [5], which provides exponential lifting coefficients. The main result is given in the last section, which shows the \textsc{NP}-hardness of the problem of computing an arbitrary LCI.

2 Computing a lifted cover inequality

Consider the knapsack set $X := \{x \in \mathbb{B}^n : a^T x \leq b\}$, where $b \in \mathbb{Z}^+$ and $a = (a_1, \cdots, a_n)^T \in \mathbb{Z}^n_+$ are given. A subset $C \subseteq N := \{1, \ldots, n\}$ is called a cover of $X$ if the sum of the items $a_i$’s over $C$ exceeds the knapsack capacity $b$; i.e., $\sum_{i \in C} a_i > b$. A cover $C$ is a minimal cover if and only if $\sum_{i \in C \setminus \{j\}} a_i \leq b$ for all $j \in C$.

For any subsets $N_0$ and $N_1$ of $N$ with $N_0 \cap N_1 = \emptyset$, denote $X(N_0, N_1)$ be the following restriction set of $X$,

$$X(N_0, N_1) = X \cap \{x \in \mathbb{B}^n : x_i = 0 \text{ for } i \in N_0; \; x_i = 1 \text{ for } i \in N_1\}.$$ 

It is easy to see that, for each cover $C$, the cover inequality

$$\sum_{i \in C} x_i \leq c - 1$$

is valid for $X(N \setminus C, \emptyset)$, where $c := |C|$ is the cardinality of $C$. The cover inequality (1) is facet defining for $\text{conv}(X(N \setminus C, \emptyset))$, which is the convex hull of $X(N \setminus C, \emptyset)$, if and only if $C$ is a minimal cover (see for example [3]).

We can consider to fix some variables to be ones as well. Assume that $(C, N_0, N_1)$ is a partition of $N$ and denote $\bar{b} = b - \sum_{i \in N_1} a_i$. In this case, the inequality (1) is facet defining for $\text{conv}(X(N_0, N_1))$ if and only if $C$ is a minimal cover of $X(N_0, N_1)$; i.e.,

$$\sum_{i \in C} a_i > \bar{b}; \quad \sum_{i \in C \setminus \{j\}} a_i \leq \bar{b} \quad \text{for all } j \in C.$$ 

Throughout this paper, we shall assume that $(C, N_0, N_1)$ be a partition of $N$ and $C$ is a minimal cover of $X(N_0, N_1)$.

In general, however, the inequality (1) may not be valid for $X$ if $N_1 \neq \emptyset$. Furthermore, even if $N_1 = \emptyset$, such an inequality may not represent a facet of $\text{conv}(X)$. To obtain a strong valid inequality, we can lift the variables in $N_0 \cup N_1$.
up-lifting. If \( l_{j+1} \in N_0 \), compute the lifting coefficient \( \alpha_{l_{j+1}} \) by solving the lifting problem

\[
\alpha_{l_{j+1}} = \min \ c - 1 + \sum_{i \in N_1} \beta_{i} - \sum_{i \in C} \beta_{i} x_{i} - \sum_{i \in N_0} \alpha_{i} x_{i} - \sum_{i \in N_0} \beta_{i} x_{i} \\
\text{s.t.} \quad \sum_{i \in C \cup N_0} a_{i} x_{i} \leq b + \sum_{i \in N_0} a_{i} - a_{l_{j+1}}, \ x \in \mathbb{B}^{c+j}.
\]

(ii) Down-lifting. If \( l_{j+1} \in N_1 \), compute the lifting coefficient \( \beta_{l_{j+1}} \) by solving the lifting problem

\[
\beta_{l_{j+1}} = \max \sum_{i \in C} x_{i} + \sum_{i \in N_0} \alpha_{i} x_{i} + \sum_{i \in N_1} \beta_{i} x_{i} - c + 1 - \sum_{i \in N_1} \beta_{i} \\
\text{s.t.} \quad \sum_{i \in C \cup N_1} a_{i} x_{i} \leq b + \sum_{i \in N_1} a_{i} + a_{l_{j+1}}, \ x \in \mathbb{B}^{c+j}.
\]

After having lifted all the variables, we obtain the lifted cover inequality (LCI)

\[
\sum_{i \in C} x_{i} + \sum_{i \in N_0} \alpha_{i} x_{i} + \sum_{i \in N_1} \beta_{i} x_{i} \leq c - 1 + \sum_{i \in N_1} \beta_{i}.
\]

See for example \[5, 12, 14\] for more details about LCI. Here we just notice that different lifting sequences may lead to different LCIs.

The inequality (4) is called a non-project LCI if \( N_1 = \emptyset \) and a project LCI if \( N_1 \neq \emptyset \) (see \[5\]). Given a lifting sequence, the non-project LCI can be computed (19) in the complexity of \( \mathcal{O}(cn) \), where \( c = |C| \) again. For the project LCI, if \( C \cup N_1 \) is a minimal cover of \( X \) and the lifting sequence is enforced to \( \left\{ l_1, \ldots, l_j, \ldots, l_r, l_{r+1}, \ldots, l_N \right\} \), where \( l_i \in N \) for \( i = 1, \ldots, |N| \) and \( j = 0, 1 \) and \( r \) is a given integer between 1 and \( |N_0| \), Gu \[3\] proved that the LCI can be obtained in the complexity of \( \mathcal{O}(cn^3) \). As mentioned above, however, the complexity of computing an LCI with an arbitrary lifting sequence is still unknown.

We close this section by noting that the LCI is invariant under scaling.

**Observation 1** Given the same partition and the lifting sequence, multiplying a positive integer to the knapsack constraint gives the same LCI.
In this section, we describe the elegant example constructed by Gu [5], which leads the lifting coefficients to be exponential. It is related to the following vector \( f \in \mathbb{Z}^{2r+1} \), where \( r \) is a given positive integer.

\[
f_1 = 1, \ f_2 = 1, \ f_3 = 1, \text{ and } f_i = f_{i-2} + f_{i-1}, \text{ for } i = 4, \ldots, 2r+1. \tag{5}
\]

We give two facts on the vector \( f \), which can easily be verified by induction.

**Observation 2** For \( j = 3, \ldots, 2r+1 \), it holds that 
\[
f_j = \sum_{i=1}^{j-2} f_i.
\]

**Observation 3** For \( j = 3, \ldots, 2r+1 \), it holds that 
\[
\frac{1}{4}(\sqrt{2} - 1)(\sqrt{2})^j \leq f_j \leq 2j.
\]

Consider the knapsack set \( X \) with \( 2r+1 \) variables, where the coefficients of the knapsack constraint are \( f_1, \ldots, f_{2r+1} \) in (5) and the associated knapsack capacity is 
\[
b = \sum_{i=1}^{2r} f_i.
\]

Since \( f_1 = 1, f_2 = 1, \) and \( \bar{b} = b - \sum_{i \in N_1} f_i = b - f_3 - \sum_{i=3}^{2r} f_i = 1 \),
we know that \( C \) is a minimal cover of \( X(N_0, N_1) = \{ x \in \mathbb{B}^2 : x_1 + x_2 \leq 1 \} \).

Now consider the partition \( (C, N_0, N_1) \) of \( \{1, \ldots, 2r+1\} \) with \( C = \{1, 2\}, N_0 = \{4, 6, 8, \ldots, 2r\}, \) and \( N_1 = \{3, 5, 7, \ldots, 2r+1\} \). Since \( f_1 = 1, f_2 = 1, \) and
\[
\bar{b} = b - \sum_{i \in N_1} f_i = b - f_3 - \sum_{i=3}^{2r} f_i = 1,
\]
we know that \( C \) is a minimal cover of \( X(N_0, N_1) = \{ x \in \mathbb{B}^2 : x_1 + x_2 \leq 1 \} \).

Now consider the lifting sequence \( \{3, 4, \ldots, 2r+1\} \); i.e., \( r_j = j + 2 \) for \( 1 \leq j \leq 2r-1 \). The following lemma is due to [5]. For completeness, we provide a proof.

**Lemma 1 (Gu 1995)** (i) \( \alpha_i = f_i \) for \( i \in N_0 \); (ii) \( \beta_i = f_i \) for \( i \in N_1 \).

**Proof.** We proceed by induction. At first, the lifting problem of the variable \( x_3 \) reads
\[
\beta_3 = \max \ x_1 + x_2 - 1 \\
\text{s.t. } x_1 + x_2 \leq 2, \ x \in \mathbb{B}^2.
\]

Hence \( \beta_3 = 1 = f_3 \). Assume that \( \alpha_i = f_i \) for \( i \leq j, i \in N_0 \) and \( \beta_i = f_i \) for \( i \leq j, i \in N_1 \), respectively. Now we consider lifting the variable \( x_{j+1} \). If \( j + 1 \in N_0 \), the problem (2) reduces to
\[
\alpha_{j+1} = \min \ 1 + \sum_{i=1}^{(j-1)/2} f_{2i+1} - \sum_{i=1}^{j} f_i x_i \\
\text{s.t. } \sum_{i=1}^{j} f_i x_i \leq 1 + \sum_{i=1}^{(j-1)/2} f_{2i+1} - f_{j+1}, \ x \in \mathbb{B}^j. \tag{6}
\]
Since
\[
\sum_{i=1}^{(j-1)/2} f_{2i+1} = f_3 + \sum_{i=2}^{(j-1)/2} (f_{2i} + f_{2i-1}) = f_3 + \sum_{i=3}^{j-1} f_i = \sum_{i=1}^{j-1} f_i = f_j + 1 = f_{j+1} - 1,
\]
where the last equality follows from Observation 2, the feasibility of the problem requires \(x_i = 0, 1 \leq i \leq j\) and hence \(\alpha_{j+1} = f_{j+1}\). If \(j+1 \in N_1\), the problem reduces to
\[
\beta_{j+1} = \max \sum_{i=1}^{j} f_i x_i - 1 - \sum_{i=1}^{j/2-1} f_{2i+1}
\]
s.t. \(\sum_{i=1}^{j} f_i x_i \leq 1 + \sum_{i=1}^{j/2-1} f_{2i+1} + f_{j+1}, x \in \mathbb{B}^j\). (7)

It is easy to verify that \(\sum_{i=1}^{j} f_i = 1 + \sum_{i=1}^{j/2-1} f_{2i+1} + f_{j+1}\). Hence the all-one vector is feasible and solves the problem (7), yielding \(\beta_{j+1} = f_{j+1}\). So the statement is true for \(j+1\) as well. By induction, this lemma holds. \(\square\)

Lemma 1 indicates that the LCI for this specific example is
\[
\sum_{i=1}^{2r+1} f_i x_i \leq \sum_{i=1}^{2r} f_i. \tag{8}
\]

By Observation 3 the input size of this example is polynomial, but the lifting coefficients \(\{f_i\}\) are exponential with \(r\). This example by Gu will play an important role in the coming complexity analysis.

4 \(\mathcal{NP}\)-hardness of computing an arbitrary lifted cover inequality

In this section, we show the \(\mathcal{NP}\)-hardness of the problem of computing an arbitrary LCI. To begin with, we give a basic property of the vector \(f\) in (6).

**Lemma 2** Let \(f\) be defined as in (5), where \(r \geq 1\) is given. For any \(\tau \in \mathbb{Z}_+\) satisfying \(0 \leq \tau \leq \sum_{i=1}^{2r+1} f_i\), there exists a subset \(S \subseteq \{1, \ldots, 2r+1\}\) such that \(\tau = \sum_{i \in S} f_i\).

**Proof.** We proceed by induction on \(r\). The result apparently holds for \(r = 1\). Assume that the result is true for some \(r \geq 1\). To verify the result for \(r+1\), it suffices to consider the case that \(\sum_{i=1}^{2r+1} f_i < \tau \leq \sum_{i=1}^{2(r+1)+1} f_i\). In fact, from Observation 3 we have that
\[
\sum_{i=1}^{2r+1} f_i = f_{2r+3}, \quad \sum_{i=1}^{2r+3} f_i = f_{2r+3} + f_{2r+2} + f_{2r+3}.
\]
So \( f_{2r+3} \leq \tau \leq f_{2r+3} + f_{2r+2} + f_{2r+3} \). Let us define \( \tau_1 \) as
\[
\tau_1 = \begin{cases} 
\tau - f_{2r+2} - f_{2r+3}, & \text{if } \tau > f_{2r+2} + f_{2r+3}; \\
\tau - f_{2r+3}, & \text{if } \tau \leq f_{2r+2} + f_{2r+3}.
\end{cases}
\]
Then it is easy to see that \( \tau_1 \leq f_{2r+3} + \sum_{i=1}^{2r+1} f_i \). By the induction assumption, there exists \( S \subseteq \{1, 2, 3, \ldots, 2r+1\} \) such that \( \tau_1 = \sum_{i \in S} f_i \). By picking a more element \( 2r+3 \) and a possible element \( 2r+2 \), we know that there exists some subset of \( \{1, 2, 3, \ldots, 2r+3\} \) such that the sum of \( f_i \)'s over this subset is exactly \( \tau \). Thus the result holds for \( r+1 \). By induction, this lemma is true. \( \square \)

Next, we introduce the RPP problem [13], which is \( \mathcal{NP} \)-hard.

**Problem RPP.** Given a nonnegative integer \( m \), a finite set \( K \) with \( |K| = k \), and a size \( \omega_i \in \mathbb{Z}^+ \) for the \( i \)-th element with \( \sum_{i \in K} \omega_i = 2(2^{m+1} - 1) \), does there exist a subset \( T \subseteq K \) such that \( \sum_{i \in T} \omega_i = 2^{m+1} - 2 \)?

We are now ready to present the main result of this paper; i.e., provide an \( \mathcal{NP} \)-hardness proof for computing an arbitrary LCI. The basic idea is as follows. Firstly, we adopt Gu’s example (see Section 3) to make the lifting coefficients exponential. Secondly, some variables fixed at zero will be lifted, where the lifting coefficients can easily be obtained. Finally, we lift a variable fixed at one, whose corresponding lifting problem is equivalent to the RPP problem and hence is \( \mathcal{NP} \)-hard.

**Theorem 1** The problem of computing an arbitrary LCI is \( \mathcal{NP} \)-hard.

**Proof.** For an RPP instance with \( m \geq 0 \) and \( K = \{2r+4, \ldots, n-1\} \), we shall construct a problem of computing an LCI in polynomial time. For convenience, define \( \lambda = 2^{m+1} - 1 \) and then \( \sum_{i \in K} \omega_i = 2\lambda \). Assume without loss of generality that \( 1 \leq \omega_i \leq \lambda - 1 \) for all \( i \in K \). We construct a problem of computing an LCI as follows. Set \( r = m + 6, n = 2r + 4 + k, \) and \( b = \sum_{i=1}^{2r+1} f_i + (3\lambda + 6) \), where \( f \) is defined as in [13]. Set the coefficients of the knapsack constraint as
\[
a_i = \begin{cases} 
\lambda f_i, & \text{for } i = 1, \ldots, 2r + 1; \\
\lambda(\lambda + 3) + 1, & \text{for } i = 2r + 2; \\
\lambda(\lambda + 3) - 1, & \text{for } i = 2r + 3; \\
\omega_i(\lambda + 1), & \text{for } i = 2r + 4, \ldots, n - 1; \\
(3\lambda + 6 + f_{2r+1}), & \text{for } i = n.
\end{cases}
\]

Define the partition \((C, N_0, N_1)\) with \( C = \{1, 2\}, N_0 = \{4, 6, 8, \ldots, 2r, 2r + 2, 2r + 3, \ldots, n-1\} \), and \( N_1 = \{3, 5, 7, \ldots, 2r + 1, n\} \). Finally, let the lifting sequence be \( \{3, \ldots, n\} \). We shall prove that the lifting coefficient \( \beta_n = f_{2r+1} + 3\lambda + 5 \) if and only if the answer to the RPP instance is yes.

Before doing this, we note that the input size of this LCI problem is polynomial of that of the RPP instance. To see this, let \( L \) be the input size of the RPP instance. Then it suffices to show that \( r \leq O(L^q) \) for some fixed integer \( q \). Since the input size of a positive integer \( t \) is \( \log_2(t + 1) \), it follows that
\[
\log_2(2^{m+2} - 2 + 1) = \log_2\left(\sum_{i \in K} \omega_i + 1\right) \leq \sum_{i \in K} \log_2(\omega_i + 1) \leq L.
\]
where the first inequality follows from $\omega_i + 1 \geq 2$ for all $i \in K$. Thus, we have that $m \leq L - 2$, which further implies that

$$r = m + 6 = O(m) = O(L).$$

For preparation, we also verify

$$f_{2r+1} \geq \frac{1}{4}(\sqrt{2} - 1)(\sqrt{2})^{2r+1} = \frac{1}{4}(2 - \sqrt{2})2^r = \frac{1}{4}(2 - \sqrt{2})2^{m+6} \quad (10)$$

$$> 2^{m+2} + 4 = 2\lambda + 6.$$

In the following, we consider the lifting procedure. By construction, the knapsack inequality of this instance reads

$$\sum_{i=1}^{2r+1} \lambda f_i x_i + [\lambda(\lambda + 3) + 1] x_{2r+2} + [\lambda(\lambda + 3) - 1] x_{2r+3} + \sum_{i=2r+4}^{n-1} \omega_i (\lambda + 1) x_i$$

$$+ \lambda(3\lambda + 6 + f_{2r+1})x_n \leq \sum_{i=1}^{2r+1} \lambda f_i + \lambda(3\lambda + 6).$$

Since $a_1 = \lambda$, $a_2 = \lambda$, and

$$b - \sum_{i \in N_1} a_i = \sum_{i=1}^{2r+1} \lambda f_i + \lambda(3\lambda + 6) - \sum_{i=1}^{r} \lambda f_{2i+1} - \lambda(3\lambda + 6 + f_{2r+1})$$

$$= \sum_{i=1}^{2r} \lambda f_i - \sum_{i=1}^{r} \lambda f_{2i+1} = \sum_{i=1}^{2r} \lambda f_i - \lambda f_3 - \sum_{i=1}^{r} \lambda(f_{2i-1} + f_{2i}) = \lambda,$$

we know that $C$ is a minimal cover of $X(N_0, N_1)$ with the cover inequality

$$x_1 + x_2 \leq 1. \quad (11)$$

**Step 1. Lifting the variables $x_3, \ldots, x_{2r+1}$**

Starting with the cover inequality (11), we know from Observation 1 and the inequality (8) that, after lifting the variables $x_3, \ldots, x_{2r+1}$, the inequality is

$$\sum_{i=1}^{2r+1} f_i x_i \leq \sum_{i=1}^{2r} f_i.$$

**Step 2. Lifting the variables $x_{2r+2}$ and $x_{2r+3}$**

We first consider the variable $x_{2r+2}$. The associated lifting problem is

$$a_{2r+2} = \min \sum_{i=1}^{2r} f_i - \sum_{i=1}^{2r+1} f_i x_i$$

$$\text{s.t.} \sum_{i=1}^{2r+1} \lambda f_i x_i \leq \lambda \sum_{i=1}^{2r} f_i - \lceil \lambda(\lambda + 3) + 1 \rceil,$$

$$x \in \mathbb{B}^{2r+1}. \quad (12)$$
The problem (12) is feasible at the zero vector since
\[ \lambda \sum_{i=1}^{2r} f_i - [\lambda(\lambda + 3) + 1] \geq \lambda(\sum_{i=1}^{2r} f_i - \lambda - 4) > \lambda(f_{2r+1} - \lambda - 4) > 0, \]
where the last inequality follows from (10). Let \( \bar{x} \) be the optimal solution of (12). Since \( f_i \in \mathbb{Z} \) for \( i = 1, \ldots, 2r + 1 \), its feasibility requires
\[ 2r + 1 \sum_{i=1}^{2r} f_i \bar{x}_i \leq \lfloor \lambda \sum_{i=1}^{2r} f_i - \lambda(\lambda + 3) - 1 \rfloor = 2r \sum_{i=1}^{2r} f_i - \lambda - 4. \]
On the other hand, from Lemma 2, we can always find an \( \bar{x} \) such that
\[ 2r + 1 \sum_{i=1}^{2r} f_i \bar{x}_i = 2r \sum_{i=1}^{2r} f_i - \lambda - 4. \]
The optimality of \( \bar{x} \) gives that
\[ \alpha_{2r+2} = \sum_{i=1}^{2r} f_i \bar{x}_i = \sum_{i=1}^{2r} f_i - (\lambda + 4). \]
Similarly, the lifting problem of \( \alpha_{2r+3} \) reads
\[
\begin{align*}
\alpha_{2r+3} = & \min_{\bar{x}} \sum_{i=1}^{2r+1} f_i - \sum_{i=1}^{2r+1} f_i \bar{x}_i - (\lambda + 4) \bar{x}_{2r+2} \\
\text{s.t.} & \sum_{i=1}^{2r+1} \lambda f_i \bar{x}_i + [\lambda(\lambda + 3) + 1] \bar{x}_{2r+2} \leq \lambda \sum_{i=1}^{2r} f_i - \lambda(\lambda + 3) - 1, \\
& \bar{x} \in \mathbb{R}^{2r+2}. \quad (13)
\end{align*}
\]
Then \( \alpha_{2r+3} = \lambda + 2 \), which is achieved at an optimal solution \( \bar{x} \) satisfying \( \bar{x}_{2r+2} = 1 \) and \( \sum_{i=1}^{2r+1} f_i \bar{x}_i = \sum_{i=1}^{2r+1} f_i - 2\lambda - 6. \)

**Step 3. Lifting the variables** \( x_{2r+4}, \ldots, x_{n-1} \)

We shall show that \( \alpha_i = \omega_i \) for all \( i \in K \) by induction. At first, consider the lifting of the variable \( x_{2r+4} \). This requires to solve the problem
\[
\begin{align*}
\alpha_{2r+4} = & \min_{\bar{x}} \sum_{i=1}^{2r+1} f_i - \sum_{i=1}^{2r+1} f_i \bar{x}_i - (\lambda + 4) \bar{x}_{2r+2} - (\lambda + 2) \bar{x}_{2r+3} \\
\text{s.t.} & \sum_{i=1}^{2r+1} \lambda f_i \bar{x}_i + [\lambda(\lambda + 3) + 1] \bar{x}_{2r+2} + [\lambda(\lambda + 3) - 1] \bar{x}_{2r+3} \leq \lambda \sum_{i=1}^{2r} f_i - \omega_{2r+4}(\lambda + 1), \quad x \in \mathbb{R}^{2r+3}. \quad (14)
\end{align*}
\]
If \( \hat{x} \) is an optimal solution of the problem (14) with \( \hat{x}_{2r+3} = 1 \), by the feasibility and (10), we have
\[ \sum_{i=1}^{2r+1} f_i \hat{x}_i + \lambda + 2 < \sum_{i=1}^{2r} f_i + \lambda + 2 < \sum_{i=1}^{2r+1} f_i. \]
This, together with Lemma 2 indicates that we can define a new feasible point $\hat{x}$ such that $\hat{x}_{2r+2} = \hat{x}_{2r+3} = 0$, and $\sum_{i=1}^{2r+1} f_i \hat{x}_i = \sum_{i=1}^{2r+1} f_i \hat{x}_i + \lambda + 2$. It is easy to see that $\hat{x}$ and $\hat{x}$ give the same objective values. Hence, we can assume that $x_{2r+3} = 0$ in the problem (14). Furthermore, since $\omega_{2r+4} \leq \lambda - 1$, similar to the problem (13), we can show that the optimal value of (14) is $\omega_{2r+4} = \omega_{2r+4}$, which is achieved at an optimal solution $\hat{x}$ satisfying $\hat{x}_{2r+3} = 0$, $\hat{x}_{2r+2} = 1$, and $\sum_{i=1}^{2r+1} f_i \hat{x}_i = \sum_{i=1}^{2r+1} f_i - (\lambda + 4) - \omega_{2r+4}$.  

Now assume that $\alpha_i = \omega_i$ for $2r + 2 \leq i \leq j$ and $i < n - 1$ and consider the lifting problem of $x_{j+1}$:

$$
\begin{align*}
\alpha_{j+1} &= \min_{x \in B^j} \sum_{i=1}^{2r+1} f_i - \sum_{i=1}^{2r+1} f_i x_i - (\lambda + 4)x_{2r+2} - (\lambda + 2)x_{2r+3} - \sum_{i=1}^j \omega_i x_i \\
\text{s.t.} \quad & \sum_{i=1}^j \lambda f_i x_i + [\lambda(\lambda + 3) + 1]x_{2r+2} + [\lambda(\lambda + 3) - 1]x_{2r+3} + \\
& \sum_{i=1}^j \omega_i (\lambda + 1)x_i \leq \lambda \sum_{i=1}^{2r} f_i - \omega_{j+1}(\lambda + 1), \ x \in B^j. (15)
\end{align*}
$$

We claim that there exists an optimal solution $\tilde{x}$ such that $\tilde{x}_{2r+3} = \tilde{x}_{2r+4} = \cdots = \tilde{x}_j = 0$. To see this, suppose that an optimal solution $\hat{x}$ is such that $\hat{x}_t = 1$ for some $t \subseteq [2r + 3, j]$. Analogously, define a new point $\hat{x}$ with $\hat{x}_t = \hat{x}_t$ for $2r + 2 \leq i \leq j$ and $i \neq t$, $\hat{x}_t = 0$, and $\sum_{i=1}^{2r+1} f_i \hat{x}_i = \sum_{i=1}^{2r+1} f_i \hat{x}_i + \theta_t$, where

$$
\theta_t = \begin{cases} \lambda + 2, & \text{if } t = 2r + 3; \\
\omega_t, & \text{otherwise.}
\end{cases}
$$

By simple calculations, $\tilde{x}$ is feasible to the problem (15) and gives the same objective value as $\hat{x}$. Similar to the problem (13), we can verify that $\alpha_{j+1} = \omega_{j+1}$. Thus by induction, we have that $\alpha_i = \omega_i$ for all $i \in K$.

**Step 4. Lifting the variable $x_n$**

Finally, we concentrate on lifting the variable $x_n$. The lifting problem is

$$
\begin{align*}
\beta_n &= \max_{x \in B^{n-1}} \sum_{i=1}^{2r+1} f_i x_i + (\lambda + 4)x_{2r+2} + (\lambda + 2)x_{2r+3} + \sum_{i=2r+4}^{n-1} \omega_i x_i - \sum_{i=1}^{2r} f_i \\
\text{s.t.} \quad & \sum_{i=1}^{2r+1} \lambda f_i x_i + [\lambda(\lambda + 3) + 1]x_{2r+2} + [\lambda(\lambda + 3) - 1]x_{2r+3} + \\
& \sum_{i=2r+4}^{n-1} \omega_i (\lambda + 1)x_i \leq \lambda \sum_{i=1}^{2r+1} f_i + \lambda(3\lambda + 6), \ x \in B^{n-1}. (16)
\end{align*}
$$

For convenience, denote $g(x)$ to be the objective function in the above problem. Consider the point $\hat{x}$ with $\hat{x}_i = 1$ for $2r + 2 \leq i \leq n - 1$ and $\sum_{i=1}^{2r+1} f_i \hat{x}_i = \sum_{i=1}^{2r+1} f_i - \lambda - 2$. By Lemma 2 such a point must exist. We can check that $\hat{x}$ is feasible to the problem (16) and $g(\hat{x}) = f_{2r+1} + 3\lambda + 4$. Furthermore, for
a binary vector $x$, if at least one of the two components $x_{2r+2}$ and $x_{2r+3}$ is equal to zero, we have that

$$g(x) \leq \sum_{i=1}^{2r+1} f_i + \lambda + 4 + \sum_{i=2r+4}^{n-1} \omega_i - \sum_{i=1}^{2r} f_i = f_{2r+1} + 3\lambda + 4.$$  

Thus to seek better values for $\beta_n$, we may set $x_{2r+2} = x_{2r+3} = 1$. In this case, the problem (16) reduces to

$$\beta_n = \max \sum_{i=1}^{2r+1} f_i x_i + \sum_{i=2r+4}^{n-1} \omega_i x_i + 2\lambda + 6 - \sum_{i=1}^{2r} f_i$$

s.t.  

$$\sum_{i=1}^{2r+1} \lambda f_i x_i + \sum_{i=2r+4}^{n-1} \omega_i(x + 1)x_i \leq \lambda \sum_{i=1}^{2r} f_i + \lambda^2, \ x \in \mathbb{B}^{n-3}.$$

Now assume that $\bar{x}$ is an optimal solution of (17). Denote $p = \sum_{i=2r+4}^{n-1} \omega_i \bar{x}_i$. It is easy to see that $p \leq 2\lambda$. Consider the following four cases.

(a) $p \leq \lambda - 2$. In this case, the knapsack constraint in the problem (17) is trivially satisfied and the optimality of $\bar{x}$ implies that

$$g(\bar{x}) = \sum_{i=1}^{2r+1} f_i + p + 2\lambda + 6 - \sum_{i=1}^{2r} f_i = f_{2r+1} + p + 2\lambda + 6 \leq f_{2r+1} + 3\lambda + 4.$$  

(b) $p = \lambda - 1$. Similar to the case (a), we have that $g(\bar{x}) = f_{2r+1} + 3\lambda + 5$.

(c) $p = \lambda$. In this case, the feasibility of $\bar{x}$ indicates that

$$\sum_{i=1}^{2r+1} f_i \bar{x}_i \leq \lambda \sum_{i=1}^{2r+1} f_i + \lambda^2 - \lambda(\lambda + 1)/\lambda = \sum_{i=1}^{2r+1} f_i - 1.$$  

Furthermore, the optimality of $\bar{x}$ implies that $\sum_{i=1}^{2r+1} f_i \bar{x}_i = \sum_{i=1}^{2r+1} f_i - 1$. Thus we can also check that $g(\bar{x}) = f_{2r+1} + 3\lambda + 5$.

(d) $\lambda + 1 \leq p \leq 2\lambda$. On one hand, the feasibility of $\bar{x}$ requires

$$\sum_{i=1}^{2r+1} f_i x_i \leq \lambda \sum_{i=1}^{2r+1} f_i + \lambda^2 - p(\lambda + 1)/\lambda = \sum_{i=1}^{2r+1} f_i + \lambda - p - 2, \ \ \ (18)$$

where the last equality follows from $p \leq 2\lambda$. On the other hand, the optimality of $\bar{x}$ requires that the inequality in (18) holds with equality, yielding

$$g(\bar{x}) = \sum_{i=1}^{2r+1} f_i + \lambda - p - 2 + p + 2\lambda + 6 - \sum_{i=1}^{2r} f_i = f_{2r+1} + 3\lambda + 4.$$  

To summarize, the lifting coefficient $\beta_n = f_{2r+1} + 3\lambda + 5$ for the problem of computing the LCI constructed in the above, if any only if $p = \lambda$ or $p = \lambda - 1$, or equivalently, the answer to the RPP instance is yes. Since the RPP problem is $\mathcal{NP}$-hard and the construction of the companion problem is in polynomial, we conclude that computing an arbitrary LCI is $\mathcal{NP}$-hard. This completes the proof. \hfill \Box
References

1. Atamtürk, A., Rajan, D.: On splittable and unsplittable flow capacitated network design arc-set polyhedra. Mathematical Programming 92(2), 315–333 (2002)
2. Balas, E.: Facets of the knapsack polytope. Mathematical Programming 8(1), 146–164 (1975)
3. Conforti, M., Cornuéjols, G., Zambelli, G.: Integer programming. Springer (2014)
4. Crowder, H., Johnson, E.L., Padberg, M.: Solving large-scale zero-one linear programming problems. Operations Research 31(5), 803–834 (1983)
5. Gu, Z.: Lifted cover inequalities for 0-1 and mixed 0-1 integer programs. Ph.D. thesis, Georgia Institute of Technology (1995)
6. Gu, Z., Nemhauser, G.L., Savelsbergh, M.W.: Lifted cover inequalities for 0-1 integer programs: Complexity. INFORMS Journal on Computing 11(1), 117–123 (1999)
7. Gu, Z., Nemhauser, G.L., Savelsbergh, M.W.P.: Lifted cover inequalities for 0-1 integer programs: Computation. INFORMS Journal on Computing 10(4), 427–437 (1998)
8. Gupta, S.R., Bulfin, R.L., Smith, J.S.: Lifting cover inequalities for the binary knapsack polytope (2005).
9. Hammer, P.L., Johnson, E.L., Peled, U.N.: Facet of regular 0-1 polytopes. Mathematical Programming 8(1), 179–206 (1975)
10. Hoffman, K.L., Padberg, M.: Improving LP-representations of zero-one linear programs for branch-and-cut. ORSA Journal on Computing 3(2), 121–134 (1991)
11. Kaparis, K., Letchford, A.N.: Separation algorithms for 0-1 knapsack polytopes. Mathematical Programming 124(1-2), 69–91 (2010)
12. Kaparis, K., Letchford, A.N.: Cover inequalities. In: Wiley Encyclopedia of Operations Research and Management Science, pp. 1074–1080 (2011)
13. Klabjan, D., Nemhauser, G.L., Tovey, C.: The complexity of cover inequality separation. Operations Research Letters 23(1-2), 35–40 (1998)
14. Nemhauser, G.L., Wolsey, L.A.: Integer and combinatorial optimization. Wiley-Interscience (1988)
15. Van Hoesel, S.P.M., Koster, A.M.C.A., Van De Leensel, R.L.M.J., Savelsbergh, M.W.P.: Polyhedral results for the edge capacity polytope. Mathematical Programming 92(2), 335–358 (2002)
16. Van Roy, T.J., Wolsey, L.A.: Solving mixed integer programming problems using automatic reformulation. Operations Research 35(1), 45–57 (1987)
17. Wolsey, L.A.: Faces for a linear inequality in 0-1 variables. Mathematical Programming 8(1), 165–178 (1975)
18. Wolter, K.: Implementation of cutting plane separators for mixed integer programs. Diploma thesis, Technische Universität Berlin (2006)
19. Zemel, E.: Easily computable facets of the knapsack polytope. Mathematics of Operations Research 14(4), 760–764 (1989)