論文 / 著書情報
Article / Book Information

Title	Computational Complexity Analysis of Set-Bin-Packing Problem
Authors	Tomonori Izumi, Toshihiko Yokomaru, Atsushi Takahashi, Yoji Kajitani
Citation	IEICE Trans. Fundamentals, Vol. E81-A, No. 5, pp. 842-849
Pub. date	1998, 5
URL	http://search.ieice.org/
Copyright	(c) 1998 Institute of Electronics, Information and Communication Engineers

PAPER Special Section on Discrete Mathematics and Its Applications

Computational Complexity Analysis of Set-Bin-Packing Problem*

Tomonori IZUMI†, Toshihiko YOKOMARU†, Atsushi TAKAHASHI†, *and* Yoji KAJITANI†, *Members*

SUMMARY The *packing problem* is to pack given items into given containers as efficiently as possible under various constraints. It is fundamental and significant with variations and applications. The *Set-Bin-Packing (SBP)* is a class of packing problems: Pack given items into as few bins which have the same capacity where every item is a set and a bin can contain items as long as the number of distinct elements in the union of the items equals to or less than the capacity. One of applications is in FPGA technology mapping, which is our initial motivation. In this paper, the computational complexity of SBP is studied with respect to three parameters α, γ, and δ which are the capacity, the upper bound of the number of elements in an item, and the upper bound of the number of items having an element, respectively. In contrast that the well known *Integer-Bin-Packing (IBP)* is NP-hard but is proved that even a simplest heuristics *First-Fit-Decreasing (FFD)* outputs exact solutions as long as $\alpha \leq 6$, our result reveals that SBP remains NP-hard for a small values of these parameters. The results are summarized on a 3D map of computational complexities with respect to these three parameters.

key words: *bin-packing, complexity, technology mapping, FPGA*

1. Introduction

The *packing problem* is to pack given items into given containers as efficiently as possible under various constraints. Since it is fundamental and significant with variations and applications, there have been many researches including computational complexity analysis and development of exact/approximate/heuristic algorithms [1]. The *Bin-Packing* is a class of packing problems: Pack a given set of items each of which has its own size into as few bins which have the same capacity.

One of applications of Bin-Packing is found in VLSI circuit clustering (or partitioning, technology mapping).Traditionally, the main concern has been in the area where the area of a gate (or a cell, a module) corresponds to the size of an item and the area of clusters (or blocks) corresponds to the capacity of bins. Recent increase of the density of circuit elements causes "pin-crisis" because the number of terminals placed at periphery of a layout area is approximately proportional to only the square-root of the density. In a specific programmable devices such as *Field Programmable Gate Arrays (FPGAs)*, the number of terminals is one of the most critical constraints for realizability. The number of terminals needed for gates in a cluster may be less than the sum of the numbers of the terminals of the gates because one common signal connected to multiple gates occupies only one terminal in going outside. From this set-theoretic property of the terminals contrast to the algebraic one of the areas, the *Set-Bin-Packing (SBP)* is abstracted where every item is a set and a bin can contain items as long as the number of distinct elements in the union of the items equals to or less than the capacity.

A realistic application of SBP is in the technology mapping of gates into *Look-Up Tables (LUTs)* of an FPGA [2], [3]. An LUT has fixed number α of input terminals and an output terminal, called the α-input LUT, and any logic circuit with α or less input signals and with an output signal is able to be implemented in an α-input LUT. For example, four AND-gates in Fig. 1 (above) are packed into two 5-input LUTs as shown in Fig. 1 (below). In the following, we discuss on SBP using the terms in technology mapping, such as 'gate,' 'signal,' and 'LUT,' for the sake of practical image.

If there are no common signals, or if the advantage of common signals are ignored, SBP is reduced to *Integer-Bin-Packing (IBP)* [4]–[7] where the size of ev-

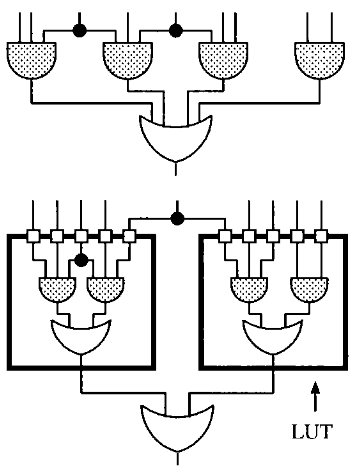


Fig. 1 An example of mapping: Four gates are mapped into two 5-input LUTs.

Manuscript received September 19, 1997.
Manuscript revised November 25, 1997.
† The authors are with the Department of Electrical and Electronic Engineering, Tokyo Institute of Technology, Tokyo, 152–8552 Japan.
* This work has been supported in part by the Research Body of CAD21 at Tokyo Institute of Technology.

ery item is an integer that follows conventional algebra. Although IBP has been known NP-hard, it is not only known to be polynomial time solvable when the capacity of bins is fixed [8] but also was proved recently [9] that a very simple algorithm *First Fit Decreasing (FFD)* outputs an exact solution if the capacity α is 6 or less, which is large enough in technology mapping.

Motivated by these circumstances, this paper is to analyze SBP from computational complexity. Let γ and δ be the upper bound of the number of input signals of a gate and the upper bound of the number δ of fanout gates of a signal (gates which has the same input signal). Although SBP is NP-hard in general, it is expected that SBP might have polynomial time algorithms when the parameters α, γ and δ are small values.

In this paper, the computational complexity of SBP with respect to these parameters is discussed and we determined for almost all the cases if SBP is NP-hard or polynomial time solvable. As opposed to our expectation, SBP remains mostly hard even within the small range.

The rest of this paper is organized as follows. The definition of SBP and preliminaries are presented in Sect. 2. In Sect. 3, our results are summarized in the 3-dimensional map of the computational complexities of SBP(α, γ, δ). Our main theorems on the NP-completeness and polynomial time solvability are in Sects. 4 and 5, respectively. Readers who only need to know the result can skip Sects. 4 and 5. Section 6 concludes the work.

2. Preliminaries

Let $S = \{s_1, s_2, \ldots, s_{N\mathrm{s}}\}$ be a set of signals and $G = \{g_1, g_2, \ldots, g_{N\mathrm{g}}\}$ be a set of logic gates. A set of *input signals* of a gate g is denoted by $\mathsf{input}(g)$. The *size* of a gate g is defined as $|\mathsf{input}(g)|$ and denoted simply by $|g|$. The set of gates which has an input signal s is referred to as the *fanout gates* of s and denoted by $\mathsf{fanout}(s)$. The *fanout* of a signal s is defined as $|\mathsf{fanout}(s)|$ and denoted simply by $|s|$. Let $\Pi = \{\pi_1, \pi_2, \ldots, \pi_\beta\}$ be a partition of G into clusters π_i's, that is, $\pi_i \subseteq G$ for $1 \leqq i \leqq \beta$, $\pi_i \cap \pi_j = \emptyset$ for $i \neq j$, and $\bigcup_{1 \leqq i \leqq \beta} \pi_i = G$. The set of *input signals* of a cluster π is defined as $\mathsf{input}(\pi) = \cup_{g \in \pi} \mathsf{input}(g)$. The *size* of a cluster π is defined as $|\mathsf{input}(\pi)|$. An *i-cluster* is a cluster whose size is i or less. The number α of input terminals of LUTs is referred to as the *capacity* of LUTs. A cluster π must be an α-cluster to be mapped into an α-input LUT. Set-Bin-Packing is defined as follows.

Set-Bin-Packing (SBP)

Instance: A set S of signals, a set G of gates, the capacity α of LUTs, and the number β of LUTs.

Question: Is there any partition Π of G into β or less α-clusters?

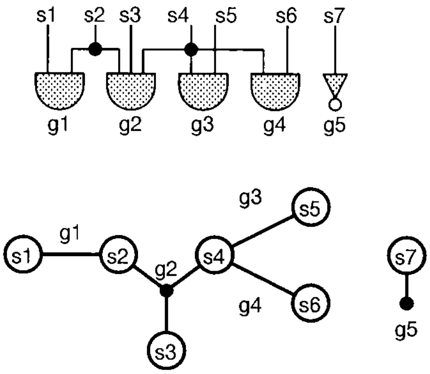


Fig. 2 The diagram consisting of signals and gates (above) and the hyper-graph representing it (below).

SBP is known to be NP-complete in general [2].

We consider SBP with limitations as follows.

1. The capacity of LUTs is α.
2. The size of every gates is at most γ.
3. The fanout of every signal is at most δ.

These parameters are constants (not input values). SBP with respect to parameters α, γ, and δ is denoted by SBP(α, γ, δ).

If there exists a pair of gates g and g' such that $\mathsf{input}(g') \subseteq \mathsf{input}(g)$, they can be mapped in the same LUT without increasing the number of LUTs. Therefore, we assume that there is no such pair of gates. If there exists a gate g such that $|g| > \alpha$, there is no partition into α-clusters. Therefore, we assume that there is no such gate, that is, $\gamma \leqq \alpha$.

For convenience, we introduce a hyper-graph to express the relation between signals and gates. A gate is represented by hyper-edge connecting the vertices which correspond to the input signals of the gate. For example, the circuit shown in Fig. 2 (above) is represented by the hyper-graph in Fig. 2 (below). Hereinafter, we omit the word 'hyper' for simplicity. A set of gates is said to be *connected* if there is a sequence for any pair of gates which connects them such that two consecutive gates in the sequence have a common signal. A *path* is a sequence of distinct gates such that two consecutive gates have exactly one common signal and such common signals are distinct each other. A *cycle* is a closed path. The length of a path (cycle) is the number of gates contained in the sequence.

3. 3D Map of Computational Complexities

Before the detailed discussion on the computational complexity of SBP, we present the 3D map with respect to α, γ, and δ as shown in Fig. 3. It summarizes our results. The map consists of three planes corresponding to $\delta = 1$ (above), $\delta = 2$ (bottom-left), and $\delta \geqq 3$

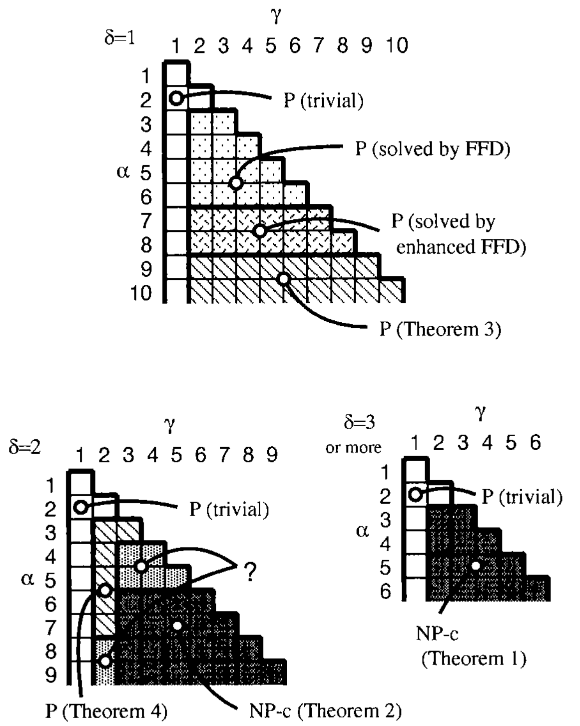


Fig. 3 3D map of the computational complexities of SBP (α, γ, δ).

(bottom-right). Each area specified by (α, γ, δ) is labeled 'P,' 'NP-c,' or '?' indicating that the computational complexity of $\mathrm{SBP}(\alpha, \gamma, \delta)$ is polynomial time solvable, NP-complete, or unknown, respectively.

$\mathrm{SBP}(\alpha, 1, \delta)$ is trivially solved for any α and δ. Thus, the areas $(\alpha, 1, \delta)$ is labeled 'P.'

The label of an area $\mathrm{SBP}(\alpha, \gamma, \delta)$ such that $\gamma = \alpha$ is same as the label of the area $\mathrm{SBP}(\alpha, \gamma - 1, \delta)$. This is by the following property: Let $I_{\mathrm{SBP}} = (S, G, \beta)$ be an instance of $\mathrm{SBP}(\alpha, \gamma, \delta)$ where $\gamma = \alpha$ and G' be a set of gates of size α in G; The answers for I_{SBP} and for $(S, G \backslash G', \beta - |G'|)$ are the same.

For the remained areas labeled 'NP-c' or 'P,' theorems are presented in Sect. 4 or Sect. 5, respectively.

4. NP-Completeness

We present two theorems on NP-completeness of SBP in this section. For polynomial reductions in the proofs, we introduce the problem called *Exact Cover by k-Sets.* Let X be a set. A *k-set* on X is a subset of X with exactly k elements. An *exact cover* of X by k-sets is a set Λ of k-sets such that every element of X occurs in exactly one member of Λ. Exact Cover by k-Sets is defined as follows.

Exact Cover by k-Sets (XkC)

Instance: A set X and a set C of k-sets on X where $|X| = kq$ and q is a positive integer.

Question: Does C contain an exact cover of X?

Note that the number k is fixed (not an input value).

Example 1: Let $X = \{x_1, x_2, x_3, x_4, x_5, x_6\}$ and $C = \{c_1, c_2, c_3, c_4\}$ where $c_1 = \{x_1, x_2, x_3\}$, $c_2 = \{x_1, x_4, x_6\}$, $c_3 = \{x_1, x_5, x_6\}$, and $c_4 = \{x_4, x_5, x_6\}$. $I_{\mathrm{XC}} = (X, C)$ is an instance of X3C. The answer for the question is 'yes' since C contains an exact cover $\Lambda = \{c_1, c_4\}$.

XkC is known to be NP-complete for $k \geqq 3$ [10].

4.1 The Case of $\alpha \geqq 3$, $\gamma \geqq 2$, and $\delta \geqq 3$

Lemma 1: $\mathrm{SBP}(\alpha, 2, 3)$ is NP-complete for $\alpha \geqq 3$.

Proof: It is easy to see that SBP is in NP. We reduce XkC of $k = \alpha$ to $\mathrm{SBP}(\alpha, 2, 3)$. Let $I_{\mathrm{XC}} = (X, C)$ be an instance of XkC where $X = \{x_1, x_2, \ldots, x_{kq}\}$, $C = \{c_1, c_2, \ldots, c_p\}$, $|X| = kq$, and $|C| = p$. The set of k-sets containing x is denoted by $\mu(x)$. The cardinality of $\mu(x)$ is denoted simply by $|x|$. The ℓ-th k-set of x is $c_i \in \mu(x)$ such that $|\{c_j \colon c_j \in \mu(x), j \leqq i\}| = \ell$. Similarly, the ℓ-th element of c, is $x_i \in c$ such that $|\{x_j \colon x_j \in c, j \leqq i\}| = \ell$. If $|x| < 1$ for some $x \in X$, I_{XC} has no exact cover and we immediately generate some SBP instance with no partition into β or less α-clusters. We assume $|x| \geqq 1$ for all $x \in X$ in the following. The assumption implies $p \geqq q$.

We construct an instance I_{SBP} of $\mathrm{SBP}(\alpha, 2, 3)$ such that G has a partition into β or less α-clusters if and only if I_{XC} has an exact cover. The construction will be made up of several components. An *element-component* of $x \in X$ consists of a set $S_1(x) = \{s_1(x, i) \colon 1 \leqq i \leqq |x|\}$ of signals and a set $G_1(x) = \{g_1(x, i) \colon 1 \leqq i \leqq |x| - 1\}$ of gates such that $\mathsf{input}(g_1(x, i))$ is $\{s_1(x, i), s_1(x, i+1)\}$. A *set-component* of $c \in C$ consists of a set $S_2(c) = \{s_2(c, i) \colon 1 \leqq i \leqq k\}$ of signals and a set $G_2(c) = \{g_2(c, i) \colon 1 \leqq i \leqq k\}$ of gates such that $\mathsf{input}(g_2(c, i))$ is $\{s_2(c, i), s_2(c, (i \bmod k) + 1)\}$. For each pair of an element $x \in X$ and a k-set $c \in C$ such that $x \in c$, a *connecting-component* between $c \in C$ and $x \in c$ consists of a set $S_3(x, c) = \{s_3(x, c, \ell) \colon 1 \leqq \ell \leqq k-3\}$ of signals and a set $G_3(x, c) = \{g_3(x, c, \ell) \colon 1 \leqq \ell \leqq k-2\}$ of gates. Let c be the i-th k-set of x and x be the j-th element of c. If $k > 3$, $\mathsf{input}(g_3(x, c, \ell))$ is $\{s_1(x, i), s_3(x, c, \ell)\}$ if $\ell = 1$, or $\{s_3(x, c, \ell{-}1), s_3(x, c, \ell)\}$ if $1 < \ell < k{-}2$, or $\{s_3(x, c, \ell{-}1), s_2(c, j)\}$ if $\ell = k{-}2$. Otherwise ($k = 3$), $\mathsf{input}(g_3(x, c, \ell))$ is $\{s_1(x, i), s_2(c, j)\}$. Note that $S_3(x, c)$ is empty if $k = 3$.

Now, we have the whole circuit consisting of $S = \bigcup_{x \in X} S_1(x) \cup \bigcup_{c \in C} S_2(c) \cup \bigcup_{c \in C, x \in c} S_3(x, c)$ and $G = \bigcup_{x \in X} G_1(x) \cup \bigcup_{c \in C} G_2(c) \cup \bigcup_{c \in C, x \in c} G_3(x, c)$. Table 1 summarizes the cardinality in various sets of gates. Note that $\sum_{x \in X} |x| = kp$. We set the number β of LUTs to $(k+1)p - q$. Note that the size of every gate is two and the degree of every signal is at most three. Now, we get an instance of $\mathrm{SBP}(\alpha, 2, 3)$, $I_{\mathrm{SBP}} = (S, G, \beta)$. It is easy to see that this construction is possible in polynomial

Table 1 The number of gates.

set of gates	# gates	set of gates	# gates
$G_1(x)$	$\|x\|-1$	$\bigcup G_1(x)$	$kp-kq$
$G_2(c)$	k	$\bigcup G_2(c)$	kp
$G_3(x,c)$	$k-2$	$\bigcup G_3(x,c)$	$(k-2)kp$
		G	k^2p-kq

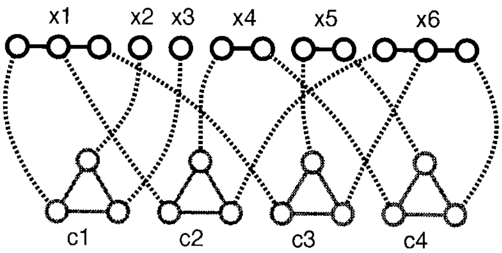


Fig. 4 The signals (vertices) and gates (edges) constructed from the X3C instance. Element-components, set-components, and connecting-components are drawn with solid, shaded, and dotted lines, respectively.

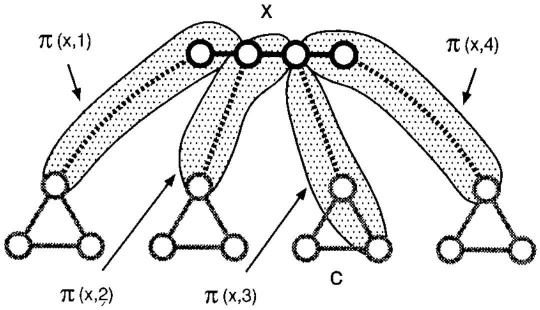


Fig. 5 Examples of $\pi(x,\ell)$s where $|x|=4$ and the third k-set c of x is in Λ.

time. For example, the SBP instance constructed from the XkC instance in Example 1 is shown in Fig. 4.

We claim that $I_{\rm SBP}$ has a partition of G into β or less α-clusters if and only if $I_{\rm XC}$ has an exact cover. Suppose that $\Lambda \subseteqq C$ is an exact cover for $I_{\rm XC}$. For each $x \in X$, let c be a k-set which covers x, that is, $c \in \Lambda \cap \mu(x)$ Assume that c is the i-th k-set of x and x is the j-th element of c. We define a cluster $\pi(x,\ell)$ of gates for each $1 \leqq \ell \leqq |x|$ as follows: $G_3(x,c') \cup \{g_1(x,\ell)\}$ if $1 \leqq \ell < i$, or $G_3(x,c') \cup \{g_2(c,j)\}$ if $\ell = i$, or $G_3(x,c') \cup \{g_1(x,\ell-1)\}$ if $i < \ell \leqq |x|$ where c' is the ℓ-th k-set of x. The definition is illustrated in Fig. 5. Let Π be $\{\pi(x,\ell)\colon x \in X, 1 \leqq \ell \leqq |x|\} \cup \{G_2(c)\colon c \notin \Lambda\}$. The size of every cluster $\pi(x,\ell)$ is α and the size of every $G_2(c)$ is α. Sets in Π are distinct each other. The union of sets in Π is G. Thus, Π is a partition of G into α-clusters. The number of clusters in Π is $\sum_{x\in X}|x| + (p-q) = (k+1)p - q = \beta$. Thus, G has a partition into β α-clusters.

Conversely, suppose that Π is a partition of G into β or less α-clusters. Let $\Lambda = \{c\colon G_2(c) \notin \Pi\}$. We claim that Λ is an exact cover of X. Let $\overline{\Lambda}$ be $C \backslash \Lambda$. Let X' be a subset of X covered by Λ, that is, $\bigcup_{c\in\Lambda} c$. Let $\overline{X'}$ be $X \backslash X'$. Since $|X'| = |\bigcup_{c\in\Lambda} c| \leqq \sum_{c\in\Lambda}|c| = k|\Lambda|$ and $|X'| + |\overline{X'}| = |X| = kq$,

$$|\Lambda| \geqq q - \frac{|\overline{X'}|}{k}. \tag{1}$$

For each $x \in \overline{X'}$, let $G'(x)$ be a set of gates in the element-component of x and the connecting-components incident to the component, that is, $G_1(x) \cup \bigcup_{c\in\mu(x)} G_3(x,c)$. Note that $G'(x)$ has $(\alpha-1)|x|-1$ gates. Let $G'_i(x)$ be a set of gates in $G'(x)$ such that a gate in $G'_i(x)$ is contained by an α-cluster in Π with i gates. By the definition of $I_{\rm SBP}$,

- any α-cluster has at most α gates,
- an α-cluster π has α gates if and only if $\pi = G_2(c)$ for some $c \in C$,
- an α-cluster π has $\alpha-1$ gates if and only if π is connected and $\pi \neq G_2(c)$ for any $c \in C$.

Since set-components incident to $G'(x)$ are α-clusters in Π, any α-cluster containing a gate in $G'_{\alpha-1}(x)$ for $x \in \overline{X'}$ consists of only gates of $G'(x)$. The number of such α-clusters is at most $\left\lfloor \frac{|G'(x)|}{\alpha-1} \right\rfloor = |x|-1$. $G'_{\alpha-1}(x)$ has at most $(\alpha-1)(|x|-1)$ gates. The number of gates in $G'(x)$ and in some α-cluster with $\alpha-2$ or less gate is $|G'(x)| - |G'_{\alpha-1}(x)| \geqq \alpha-2$. Totally, the number of gates contained in some α-cluster with $\alpha-2$ or less gates is at least $|\overline{X'}|(\alpha-2)$. The number of α-clusters with $\alpha-2$ or less gates is at least $|\overline{X'}|$. Let Π_i be a set of α-clusters with i gates. We have

$$\begin{aligned} |G| &= \sum_{i=1}^{\alpha} i|\Pi_i| \\ &\leqq \alpha|\Pi_\alpha| + (\alpha-1)|\Pi_{\alpha-1}| + (\alpha-2)\sum_{i=1}^{\alpha-2}|\Pi_i| \\ &= |\Pi_\alpha| + (\alpha-1)|\Pi| - \sum_{i=1}^{\alpha-2}|\Pi_i| \\ &\leqq |\overline{\Lambda}| + (\alpha-1)|\Pi| - |\overline{X'}| \end{aligned} \tag{2}$$

Upon substituting $\alpha = k$, $|\Pi| \leqq \beta = (k+1)p - q$, $|G| = k^2p - kq$, and $|\overline{\Lambda}| = p - |\Lambda|$ for (2),

$$|\Lambda| \leqq q - |\overline{X'}|. \tag{3}$$

Finally, from inequalities (1), (3), $|\overline{X'}| \geq 0$, $|\Lambda| \geq 0$, and $k \geq 3$, we have $|\overline{X'}| = 0$ and $|\Lambda| = q$. These equations imply that Λ is an exact cover of $I_{\rm XC}$. □

Lemma 1 leads the theorem.

Theorem 1: SBP(α,γ,δ) is NP-complete for $\alpha \geq 3$, $\gamma \geq 2$, and $\delta \geq 3$.

4.2 The Case of $\alpha \geqq 6$, $\gamma \geqq 3$, and $\delta \geqq 2$

Lemma 2: SBP$(\alpha, 3, 2)$ is NP-complete where $\alpha = 2k$ for $k \geqq 3$.

Proof: Since this lemma is proved by the similar way to Lemma 1, we only illustrate the construction of an SBP$(\alpha, 3, 2)$ instance from an XkC.

We construct SBP$(\alpha, 3, 2)$ instance which consists of element-components, set-components, and connecting-components. For example, the SBP instance constructed from the XkC instance in Example 1 is shown in Fig. 6. For each $x \in X$, the element-component of x is a path consisting of $2|x| - 1$ gates of size 3. For each $c \in C$, the set-component of c is a cycle consisting k gates of size 3. For each pair of an element $x \in X$ and a k-set $c \in C$ such that $x \in c$, the connecting-component between x and c is a path consisting of $k - 2$ gates. It connects between the signal of $(2i - 1)$-th gate in the element-component of x and the signal of the j-th gate in the element-component of c such that c is i-th k-set of x, x is j-th element of c, and the signals are not common in the path or cycle. The size of a gate in a connecting-component is 3 except that the $(\lfloor \frac{k-3}{2} \rfloor + 1)$-th gate from the side of an element-component is 2. The number β of LUTs is $(k + 1)p - q$.

We claim that there is a partition of G into β or less α-clusters if and only if I_{XC} has an exact cover. A key issue for the proof is that k-sets corresponding to set-components each of which is a cluster in the partition are the complements of the exact cover. □

Lemma 3: SBP$(\alpha, 3, 2)$ is NP-complete where $\alpha = 2k + 1$ for $k \geqq 3$

Proof: This lemma is also proved by the similar way to Lemmas 2 and 1. We only describe the differences from the construction in the proof of Lemma 2.

A connecting-component is a path consisting $k - 2$ gates of size 3 (not containing a gate of size 2). There

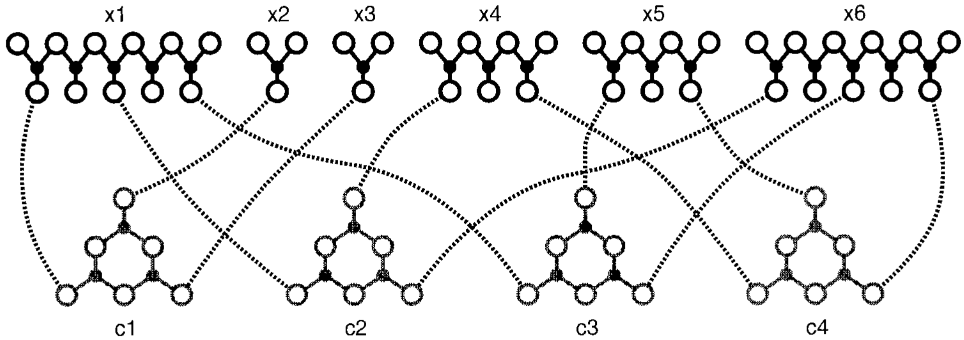


Fig. 6 The signals and gates constructed from the X3C instance. Element-components, set-components, and connecting-components are drawn with solid, shaded, and dotted lines, respectively.

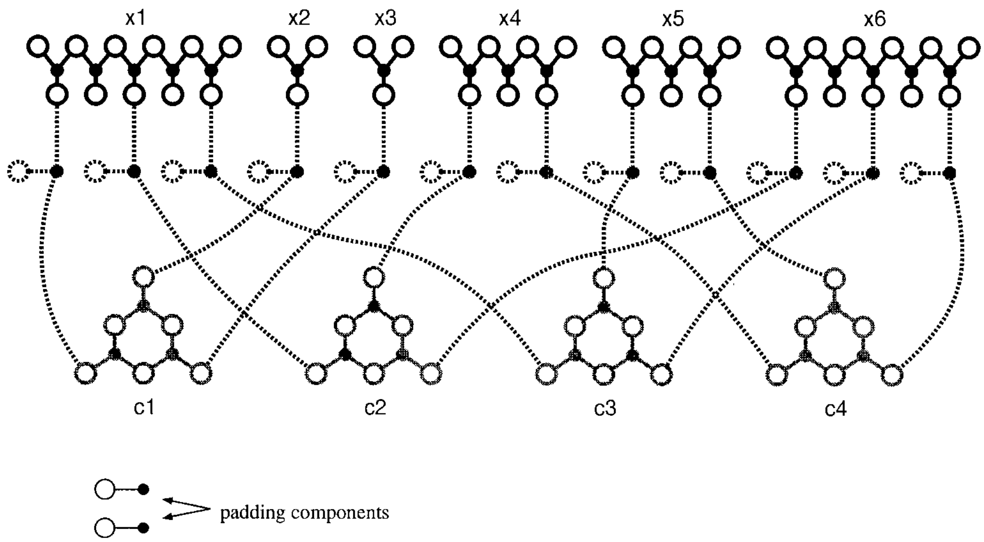


Fig. 7 The signals and gates constructed from the X3C instance. Element-components, set-components, connecting-components, and padding-components are drawn with solid, shaded, dotted, and thin lines, respectively.

are *padding-components* each of which is a gate of size 1. The number of padding-components is $p - q$. A padding-component together with a set-component of k-set which is not in the exact cover forms an α-cluster in Π. For example, the SBP instance constructed from the XkC instance in Example 1 is shown in Fig. 7. □

Lemmas 2 and 3 lead the theorem.

Theorem 2: SBP(α,γ,δ) is NP-complete for $\alpha \geqq 6$, $\gamma \geqq 3$, and $\delta \geqq 2$.

5. Polynomial Time Solvability

We present two theorems on polynomial time solvability of SBP in this section.

5.1 The case of $\forall\alpha$, $\forall\gamma$, and $\delta = 1$

First, we consider SBP$(\alpha,\gamma,1)$ which is equivalent to IBP. Although it is already known that IBP is polynomial time solvable when α is fixed, we present a proof which is a base of the proof of Theorem 4 on the polynomial time solvability of SBP$(\alpha,2,2)$.

Theorem 3: SBP$(\alpha,\gamma,1)$ is polynomial time solvable for any α and γ.

Proof: Let $I_{\rm SBP} = (S,G,\beta)$ be an instance of SBP$(\alpha,\gamma,1)$. A multi-set of non-negative integers whose sum is α is called a *divider* of α. A set π of gates is said to *fit* a divider d when the multi-set of gate sizes in π is a subset of d. For example, $\{2,1,1,1\}$ is a divider for $\alpha = 5$ and $\pi_1 = \{g_1,g_2,g_3\}$ where $|g_1| = 2$, $|g_2| = 1$, and $|g_3| = 1$ fits the divider. A multi-set a of dividers such that $|a| \leqq \beta$ is called an *accepter*. A partition Π of G into α-clusters is also said to *fit* an accepter a when there exists a one-to-one mapping from Π to a such that every α-cluster in Π fits the mapped divider. An accepter a is said to be *feasible* for G when there exists a partition of G which fits a. Given an accepter a, the feasibility of a can be checked by confirming if the number of gates of size i is at most the total number of integer 'i's contained in dividers in the accepter for each $1 \leqq i \leqq \alpha$.

The algorithm to solve SBP$(\alpha,\gamma,1)$ is described as follows.

1. Enumerate the set D of all dividers of α.
2. Enumerate the set A of all accepters of α.
3. For every $a \in A$, check the feasibility and answer 'yes' if a is feasible.
4. Answer 'no.' (no feasible accepter is found)

The times to enumerate D and A are $O(\alpha|D|)$ and $O(|D||A|)$, respectively. The time to check feasibility is $O(\alpha|D|)$ for an accepter. The total time is $O(\alpha|D||A|)$. Since $|D| \leqq 2^{\alpha-1}$ and $|A| < (\beta+1)^{|D|}$, the total computational time for the algorithm is

$$O\left(\alpha \times 2^{\alpha-1} \times (\beta+1)^{2^{\alpha-1}}\right).$$

Since α is a fixed value, this is polynomial of the size of the instance. □

The time complexity of the algorithm which works for any α is a high order polynomial even if α is small. However, faster algorithms exist for specified α.

Theorem[9]: The FFD algorithm solves IBP in $O(|G|\log|G|)$ time for $\alpha \leqq 6$.

Theorem[9]: The enhanced FFD algorithm in [9] solves IBP in $O(|G|\log|G|)$ time for $\alpha \leqq 8$.

5.2 The case of $\forall\alpha$, $\gamma = 2$, and $\delta = 2$

We consider SBP$(\alpha,2,2)$. An instance of SBP$(\alpha,2,2)$ consists of only paths and cycles as shown in Fig. 8. We present two trivial facts as lemmas on the length of a path and a cycle. The proofs are omitted.

Lemma 4: Let $I_{\rm SBP} = (S,G,\beta)$ be an instance of SBP$(\alpha,2,2)$ with a cycle $(g_1,g_2,\ldots,g_\ell)$ such that $\ell > \alpha$. Let g'_ℓ be a gate whose input signals are a signal s'_1 not contained in S and the common signal of $g_{\ell-1}$ and g_ℓ. The answers for $I_{\rm SBP}$ and for $(S+s'_1, G-g_\ell+g'_\ell,\beta)$ are the same.

The lemma suggests that a cycle longer than α can be cut open to form a path as pre-processing.

Lemma 5: Let $I_{\rm SBP} = (S,G,\beta)$ be an instance of SBP$(\alpha,2,2)$ for $\alpha \leqq 7$ with a path $(g_1,g_2,\ldots,g_\ell)$ such that $\ell > \alpha - 1$. The answers for $I_{\rm SBP}$ and for $(S, G\backslash\{g_1,g_2,\ldots,g_{\alpha-1}\}, \beta-1)$ are the same.

The lemma suggests that if there is a path longer than $\alpha-1$, $\alpha-1$ gates from one end of the path can be clustered as pre-processing for the case that $\alpha \leqq 7$.

Based on the approach in the proof of Theorem 3, we have the following lemma.

Lemma 6: If the length of paths and cycles is bounded by the constant, SBP$(\alpha,2,2)$ is polynomial time solvable for any α.

Proof: Let $I_{\rm SBP} = (S,G,\beta)$ be an instance of SBP$(\alpha,2,2)$ and the bound of the length of paths and cycles be $\ell_{\max}$. By Lemma 4, we assume that $I_{\rm SBP}$ consists of only paths of length $\ell_{\max}$ or less and cycles of length α or less. As we do in the proof of Theorem

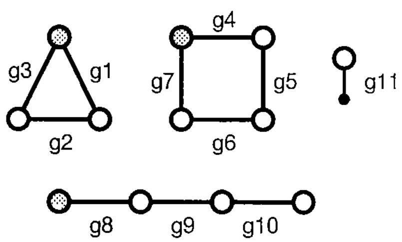


Fig. 8 An instance of SBP$(\alpha,2,2)$. A shaded signal is the origin of a path or cycle. The direction is clockwize or left-to-right.

3, we enumerate dividers and accepters and check the feasibility for every accepter.

We introduce a *label* representing a sub-path of a path or cycle. The direction of a path and the direction and the origin of a cycle is decided arbitrarily. A label $\mathsf{P}_\ell\langle i:j\rangle$ represents the sub-path from i-th gate to j-th gate of a path which consists of ℓ gates of size 2. A label T represents a gate of size 1. A label $\mathsf{C}_\ell\langle i:j\rangle$ represents the sub-path from i-th gate to j-th gate of a cycle of length ℓ. The size of a label is defined by the number of signals in the sub-path. Table 2 summarizes the range of parameters and the size of a sub-path.

A divider of α is defined by a multi-set of labels such that the sum of sizes of labels in the multi-set is at most α. A set π of gates is said to *fit* a divider d when the multi-set of labels representing sub-paths in π is a subset of d. For example, a set of gates $\{g_1, g_2, g_3, g_9\}$ shown in Fig. 8 fits a divider $\{\mathsf{C}_3\langle 1:3\rangle, \mathsf{P}_3\langle 2:2\rangle\}$. Let D be the set of all dividers of α. An accepter is defined as same in the proof of Theorem 3. The function to check the feasibility of an accepter is shown in Fig. 9. The framework of the algorithm is same as one in the proof of Theorem 3.

The times to enumerate D and A are $O(\alpha|D|)$ and $O(|D||A|)$, respectively. The time to check feasibility is $O(|G|\beta\alpha|D|)$ for a partitioning-candidate and repeated for $|A| < (\beta+1)^{|D|}$ times. The total computational time is

Table 2 The range of indices and the size of a sub-path.

	range	size
$\mathsf{P}_\ell\langle i:j\rangle$	$1 \leqq i \leqq j \leqq \ell \leqq \ell_{\max}$	$j-i+2$
T	—	1
$\mathsf{C}_\ell\langle i:j\rangle$	$3 \leqq i \leqq j \leqq \ell \leqq \alpha$	$\begin{cases} \ell \cdots i=1 \wedge j=\ell \\ j-i+2 \cdots \text{o.w.} \end{cases}$

```
Function Check-Feasibility;
Input an accepter a;
{
  Label all gates 'unaccepted';
  For each divider d ∈ a, do {
    For each label ℓ in d, do {
      Find a sub-path represented by ℓ
        which consists of only unaccepted gates.
      Label them 'accepted' if exists;
    }
  }
  If all gates are accepted, answer 'feasible',
  otherwise, answer 'infeasible';
}
```

Fig. 9 The function to check feasibility of an accepter.

$$O\left(\alpha|D||G|\beta(\beta+1)^{|D|}\right).$$

Since $|D|$ depends only on the constants α and $\ell_{\max}$, this is polynomial of the size of the instance. □

Lemmas 5 and 6 lead the following theorem.

Theorem 4: SBP$(\alpha,2,2)$ SBP$(\alpha,2,2)$ is polynomial time solvable for $\alpha \leqq 7$.

6. Conclusion

We analyzed the computational complexity of the Set-Bin-Packing problem with limitations by capacity α of LUTs, upper bound γ of the gate size, and upper bound δ of the fanout of signals. It is summarized in the 3D map. Our main results are Theorems 1, 2, and 4 whose contributions are to fill almost the area for $\delta \geqq 2$. However, the 3D map has not been completed remaining some areas still open. Among them, SBP$(4,3,3)$, SBP$(5,3,3)$, SBP$(5,4,3)$ and SBP$(\alpha,2,2)$ for $\alpha \geqq 8$ are essential since SBP$(4,4,3)$ and SBP$(5,5,3)$ are reduced to SBP$(4,3,3)$ and SBP$(5,4,3)$, respectively. As opposed to our initial expectation that SBP is solvable for small parameters, it was revealed that SBP is mostly hard. The non-trivial solvable cases are only SBP$(\alpha,2,2)$ which may cover few practical cases.

References

[1] S. Martello and P. Toth, "Knapsack problems," John Wiley & Sons Ltd., 1990.

[2] R. Murgai, R.K. Brayton, and A.S.-Vincentelli, "Cube-packing and two-level minimization," Proc. International Conf. on CAD, pp.115–122, 1993.

[3] J. Cong and Y.-Y. Hwang, "Structural gate decomposition for depth-optimal technology mapping in LUT-based FPGA design," Proc. 33rd Design Automation Conf., pp.726–729, 1996.

[4] S.D. Brown, R.J. Francis, J. Rose, and Z. Vranesic, "Field-Programmable Gate Arrays," Kluwer Academic Publishers, 1992.

[5] R.J. Francis, J. Rose, and K. Chung, "Chortle: A technology mapping program for lookup table-based field programmable gate arrays," Proc. 27th Design Automation Conf., pp.613–619, 1990.

[6] R.J. Francis, J. Rose, and Z. Vranesic, "Chortle-crf: Fast technology mapping for lookup table-based FPGAs," Proc. 28th Design Automation Conf., pp.227–233, 1991.

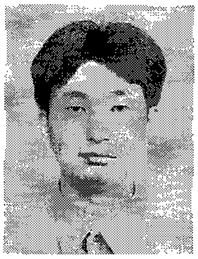
[7] R.J. Francis, J. Rose, and Z. Vranesic, "Technology mapping of lookup table-based FPGAs for performance," Proc. International Conf. on CAD, pp.568–571, 1991.

[8] M.R. Garey and D.S. Johnson, "Computers and intractability: A guide to the theory of NP-completeness," W.H. Freeman and Company, 1979.

[9] T. Yokomaru, T. Izumi, A. Takahashi, and Y. Kajitani, "Solution of integer bin packing problem with fixed capacity by FFD," Technical Report, 95-DA-76-1, IPSJ, 1995.

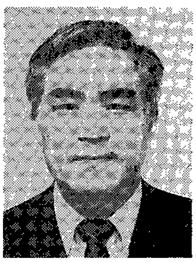
[10] R.M. Karp, "Reducibility among combinatorial problems," in eds. R.E. Miller and J.W. Thatcher, Complexity of Computer Computations, pp.85–103. Plenum Press, New York, 1972.

Tomonori Izumi received his B.E. degree in computer science and M.E. degree in electrical and electronic engineering from the Tokyo Institute of Technology, Tokyo, Japan, in 1992 and 1994, respectively. He is currently a Ph.D. candidate of electrical and electronic engineering at the Tokyo Institute of Technology. His research interests are in combinatorial algorithms applied to VLSI layout design.



Toshihiko Yokomaru received his B.E. and M.E. degrees in electrical and electronic engineering from the Tokyo Institute of Technology, Tokyo, Japan, in 1994 and 1996, respectively. He is currently a Ph.D. candidate of electrical and electronic engineering at the Tokyo Institute of Technology. His research interests are in combinatorial algorithms applied to VLSI layout design.

Atsushi Takahashi received his B.E., M.E., and D.E. degrees in electrical and electronic engineering from the Tokyo Institute of Technology, Tokyo, Japan, in 1989, 1991, and 1996, respectively. He had been with the Tokyo Institute of Technology as a research associate from 1991 to 1997 and has been an associate professor since 1997 in the Department of Electrical and Electronic Engineering. His research interests are in VLSI layout design and combinatorial algorithms. He is a member of the Information Processing Society of Japan and IEEE.



Yoji Kajitani received his B.E., M.E. and D.E. degrees from the Tokyo Institute of Technology, Tokyo, Japan, all in electrical engineering, in 1964, 1966 and 1969, respectively. He has been a professor in the Department of Electrical and Electronic Engineering at the Tokyo Institute of Technology since 1985, and has been a professor at the Japan Advanced Institute of Science and Technology from 1992 to 1995. His current research interests are in combinatorial algorithms applied to VLSI layout design. He was awarded IEEE Fellowship in 1992.