設 計 自 動 化 76－1
（1 9 9 5． 7． 2 0）

容量を固定した整数ビンパッキング問題の FFD 法による解法

横丸 敏彦 * †, 泉 知論 †, 高橋 篤司 †, 梶谷 洋司 † ‡

† 東京工業大学 工学部 電気・電子工学科
〒152 東京都 目黒区 大岡山 2-12-1
* 電子メール : yokomaru@ss.titech.ac.jp

‡ 北陸先端科学技術大学院大学
情報科学研究科
〒923-12 石川県 能美郡 辰口町 旭台

LUT をベースとした FPGA のテクノロジーマッピングに登場する，2 段部分論理回路を最小数の LUT に割り当てる問題は，共通信号線を考慮しない場合，整数ビンパッキング問題となる．整数ビンパッキング問題ではビンの容量を固定した場合，全探索アルゴリズムにより多項式時間で最適解が求まることがよく知られているが，実用性に欠ける．本文では，整数ビンパッキング問題の高速近似アルゴリズムである FFD 法が容量を 6 以下に固定した時に最適解を求めることを示す．また，入力を制限することにより，容量を 7 および 8 に固定した時に FFD 法が最適解を与えることを示し，容量を 8 以下に固定した場合に高速に最適解を求める多項式時間アルゴリズムを提案する．

Solution of Integer Bin Packing Problem with Fixed Capacity by FFD

Toshihiko Yokomaru * †, Tomonori Izumi †, Atsushi Takahashi †, Yoji Kajitani † ‡

† Dept. of Electrical and Electronic Engrg.,
Tokyo Inst. of Tech.
Ookayama, Meguro, Tokyo, 152 Japan
* E-mail : yokomaru@ss.titech.ac.jp

‡ School of Inf. Sci.,
Japan Advanced Inst. of Sci. and Tech.
Tatsunokuchi, Ishikawa, 923-12 Japan

In technology mapping of Look Up Table (LUT) based FPGA, the problem of mapping a two-level subcircuit into LUTs is formulated as the Integer Bin Packing Problem without taking account of the advantage of input signals connected to more than one gate. It is known that Integer Bin Packig Problem can be solved in polynomial time by exhaustive search when the size of bins is fixed, however, which is not practical. In this paper, we show that First Fit Decreasing (FFD) which is a fast approximation algorithm solves Integer Bin Packing Problem when the size of bins is fixed to less than or equal to 6. We show that FFD gives optimal solutions for some class of instances when the size of bins is fixed to 7 or 8, and suggest a polynomial time algorithm which solves fast when the size of bins is fixed to less than or equal to 8.

1 はじめに

FPGA はプログラムによって回路を実現する LSI チップである [1]. 回路を安価に短時間で実現できるので, 試作品や小量生産などを必要とする分野でよく用いられる. できるだけ大規模な回路を実現するためには, 回路を効率よく論理ブロックに割り当てることが重要である [2].

FPGA にルックアップテーブル方式の論理ブロックを用いると, 面積の観点から論理回路を効率良く実現できることが知られている. 論理ブロックが B 入力のルックアップテーブルからなる FPGA で論理回路を実現しようとする場合, 論理回路は B 入力以下の回路に分解され, それぞれが論理ブロックに割り当てられる. 最終目標は, できるだけ少ない論理ブロックで論理回路を実現することである.

今までに様々な論理ブロックへの割り当て手法が提案されているが, その中にビンパッキング [7] を用いて論理ブロックの最小化をはかる手法がある. 例えば, 2 段論理回路の 1 段目を構成するゲートを論理ブロックに割り当てる場合の論理ブロック数最小化問題は, 論理ブロックの入力数を容器 (bin) の大きさ, 1 段目ゲートの入力数を要素 (item) の大きさとすると, 整数ビンパッキング問題に定式化できる [3].

本研究では, [3] でも用いられているビンパッキング問題の近似解法である First Fit Decreasing(FFD) 法が容量 6 以下に限ると最適解を出力することを示す. さらに, 容量 8 以下で最適解を出力する多項式時間アルゴリズムを提案する.

2 整数ビンパッキング問題

整数ビンパッキング問題は次のように定式化できる.

Integer Bin Packing 問題

入力: $G = \{g_1, g_2, \ldots, g_n\}$, $s(g_i) \in Z^+$ $(1 \le i \le n)$, $B \in Z^+$, $K \in Z^+$.

質問: G を $\sum_{g \in G_i} s(g) \le B$ を満たす $G_1, G_2, \ldots, G_l (l \le K)$ に分割できるか?

以下では, $g \in G$ を要素, $s(g)$ を要素 g の大きさ, G_i を容器, B を容量と呼ぶことにする.

整数ビンパッキング問題は NP 完全であることが知られているが, 全探索アルゴリズムにより, $O(|G|^{2^B})$ の計算量で容器数最小の解が求まる, ということも知られている [4]. この計算量は B を固定すると $|G|$ の多項式となるが, 実用的とはいえない.

3 FFD 法と改良 FFD 法

入力 G 中の大きさ $m(\in Z^+)$ の要素の数を $n(G, m)$ とする. 容量より大きい要素はないものとして考える. 入力 G の任意の解において, 大きさが容量に一致する要素は単独で一つの容器に埋め込まれるので, 以下では $n(G, B) = 0$ である入力を考える.

整数ビンパッキング問題の近似アルゴリズムである, FFD 法の計算量は $O(|G| \log |G|)$ である. アルゴリズムは次のようになる.

First Fit Decreasing(FFD) 法

入力: G, $s(g) \in Z^+$ for each $g \in G$, $B \in Z^+$.

Step 0: $B > s(g_1) \ge s(g_2) \ge \cdots \ge s(g_n)$ となるよう G をソートする $(g_i \in G, 1 \le i \le n)$.

Step 1: $m = 0, i = 1$ とする.

Step 2: $\sum_{g \in G_j} s(g) \le B - s(g_i)$ を満たす最小の正整数を j とし, $G_j = G_j \cup \{g_i\}$ とする. $j > m$ ならば $m = m + 1$ とする.

Step 3: $i \ne n$ ならば $i = i + 1$ とし, step2 へ.

Step 4: $G_1, G_2, \ldots, G_m$ を出力し終了.

以下では, 与えられた入力 $G = \{g_1, g_2, \ldots, g_n\}$ の中の要素は大きさの順にソートされているものとする. つまり, $s(g_1) \ge s(g_2) \ge \cdots \ge s(g_n)$ とする.

3.1 容量 6 以下の場合

容量を 6 以下に固定すると, FFD 法に関して, 次の定理が成立する.

定理 1 容量を 6 に固定した時, FFD 法は最適解を出力する.

証明は 3.1.2節に示す.

3.1.1 証明に必要な補題

まず, 証明に必要な補題をいくつか示す. ただしこの節で示す補題は, 容量を 6 以下に制限してないことを注意しておく. 容量を B とする.

補題 1 $G = \{g_1, g_2, \ldots, g_n\}, G' = \{g_1, g_2, \ldots, g_{n-1}\}$, $s(g_n) = 1$ とする. FFD 法が G' の最適解を出力するならば, FFD 法は G の最適解を出力する.

証明　G'の最適解を $A = \{G_1, G_2, \ldots, G_a\}$ とする．

G 中の最初から $n-1$ 個の要素の集合は G'に一致するので，入力 G に FFD 法を適用すると，g_1から g_{n-1}までを容器に埋め込んだ状態は A と同じになる．A で使われている容器の中身より

1. $\forall G_i \in A, \sum_{g \in G_i} s(g) = B$のとき

FFD 法で要素 g_nは $G_1, G_2, \ldots, G_a$のいずれにも埋め込まれず，$a+1$ 個目の容器 G_{a+1}に埋め込まれる．よって FFD 法で使用する容器は $a+1$ 個である．

ところで G に含まれる要素の大きさの総和は $aB+1$ であり，G のいかなる解においても $\lceil (aB+1)/B \rceil = a+1$ 個以上の容器を必要とする．よって FFD 法による G の解は最適解である．

2. $\exists G_i \in A, \sum_{g \in G_i} s(g) < B$のとき

FFD 法で要素 g_nは$\sum_{g \in G_i} s(g) < B$を満たす，$i$ が最小の G_iに埋め込まれるので，FFD 法で使用する容器は a 個である．

入力 G に対する最適解が $a-1$ 個以下の容器を使用したと仮定する．その解から g_nを取り除くと G'の解となり G'の最適解が a 個の容器を使用することに矛盾する．よって G の最適解は a 個以上の容器を使用する．したがって，FFD 法による解は最適解である．□

補題 2 $n(G,1) = 0$ かつ $n(G, B-1) = 0$ である任意の入力 G に対し FFD 法が最適解を出力するならば，FFD 法は任意の入力に対し最適解を出力する．

証明　任意の入力を G とする．G より，大きさ $1, B-1$ の要素をすべて取り除くことにより得られる入力を G'とする．FFD 法は G'に対し最適解を出力すると仮定する．

大きさ $B-1$ の要素を $n(G, B-1)$ 個 G'に加えることにより得られる入力を G''とする．G''は大きさ 1 の要素を含まないので，G''の任意の解において，大きさ $B-1$ の要素は単独で異なる容器に埋め込まれる．したがって，G'の最適解に大きさ $B-1$ の要素をそれぞれ 1 個含む容器を $n(G, B-1)$ 個加えた解は G''の最適解であり，G''に対し FFD 法は最適解を出力する．

帰納法により，FFD 法が G に対し最適解を出力することを示す．補題 1より，大きさ 1 の要素を 1 個 G''に加えることにより得られる入力に対し FFD 法は最適解を出力する．大きさ 1 の要素を k個 G''に加えることにより得られる入力に対し，FFD 法が最適解を出力すると仮定する．補題 1より，G'に大きさ 1 の要素を $k+1$ 個加えることにより得られる入力に対し，FFD 法は最適解を出力する．□

補題 3 与えられた入力 G から大きさ $i, B-i$ $(1 \le i \le \lceil B/2 \rceil)$ の要素を 1 個ずつ取り除いた入力を G'とする．FFD 法が G'の最適解を出力するならば，FFD 法は G の最適解を出力する．

証明　$s(g_a) = B - i, s(g_b) = i$ $(a < b)$ とする．一般性を失わず，$s(g_{a-1}) > s(g_a)$ とする．$i \ne B/2$ の場合，一般性を失わず，$s(g_{b-1}) > s(g_b)$ とする．$i = B/2$ の場合，一般性を失わず，$b = a+1$ とする．

FFD 法で G を容器に埋め込む場合を考える．$s(g_1) \ge s(g_2) \ge \cdots \ge s(g_{a-1}) > s(g_a) \ge B/2$ より，$g_1, g_2, \ldots, g_a$はそれぞれ異なる容器に埋め込まれる．g_aが埋め込まれた容器を G_aとする．$s(g_a) + s(g_i) > s(g_a) + s(g_b) = B$ $(a+1 \le i \le b-1)$ より，g_iは $g_1, g_2, \ldots, g_a$とは異なる容器に埋め込まれる．また $s(g_i) + s(g_b) > s(g_a) + s(g_b) = B$ $(1 \le i \le a-1)$ により，g_bは $g_1, g_2, \ldots, g_{a-1}$と同じ容器には埋め込めず，$G_a$に埋め込まれる．

また，G'を FFD で容器に埋め込む場合，容器 G_aを除き G を埋め込む場合と動作は同じであり，FFD 法が G を x 個の容器に埋め込むとすると，FFD 法は G'を $x-1$ 個の容器に埋め込む．

入力 G に対する解が $x-1$ 個以下の容器を使用したと仮定する．g_aと g_bが異なる容器に埋め込まれた場合，g_a, g_bが埋め込まれた容器をそれぞれ G_a, G_bとすると，G_a, G_bに埋め込まれた他の要素の大きさの和は B以下であり，g_a, g_bを同一容器に，他の要素を高々1 個の容器に埋め込むことにより，容器数が同じであり，g_aと g_bが同一容器に埋め込まれた解が構成できる．よって一般性を失わず，g_aと g_bは同じ容器にあると仮定できる．その解において，g_a, g_bの入った容器を取り除くことにより，G'を $x-2$ 個以下の容器に埋め込む解を得ることができる．これは FFD 法が G'を最適に埋め込むという仮定に矛盾する．よって G を収めるのに必要な容器は x 個以上である．したがって，入力 G に対して FFD 法は最適解を出力する．□

系 1 与えられた入力 G から大きさ $i, B-i$ の要素を k個ずつ取り除いた入力を G'とする．FFD 法が G'の最適解を出力するならば，FFD 法は G の最適解を出力する．

証明　FFD 法が G'に対し最適解を出力すると仮定する．補題 3より，大きさ $i, B-i$ の要素を 1 個ずつ G'に加えることにより得られる入力に対し，FFD 法は最適解を出力する．帰納法により，大きさ $i, B-i$ の要素を k個ずつ G'に加えることにより得られる入力に対し，FFD 法は最適解を出力する．□

補題 4 $n(G,1) = n(G, B-1) = 0$，すべての $i(1 < i < \lfloor B/2 \rfloor)$ に対し $n(G,i) = 0$ または $n(G, B-i) = 0$，かつ

$n(G,B/2) \leq 1$ である任意の入力 G に対し FFD 法が最適解を出力するならば，FFD 法は任意の入力に対し最適解を出力する．

証明 任意の入力を G とする．大きさ $1, B-1$ の要素をすべて，大きさ $i, B-i$ の要素をそれぞれ $\min(n(G,i), n(G,B-i))$ 個 $(1 < i < \lfloor B/2 \rfloor)$，大きさ $B/2$ の要素を $2\lfloor n(G,B/2)/2 \rfloor$ 個 G から取り除くことによって得られる入力を G' とする．

G' は補題の条件を満足する．FFD 法は G' に対し最適解を出力すると仮定する．大きさ $1, B-1$ の要素をそれぞれ $n(G,1)$ 個，$n(G,B-1)$ 個 G' に加えることにより得られる入力を G'' とする．補題2より，G'' に対し FFD 法は最適解を出力する．系1より，大きさ $2, B-2$ の要素をそれぞれ $\min(n(G,2), n(G,B-2))$ 個 G'' に加えることにより得られる入力に対し FFD 法は最適解を出力する．同様に，大きさ $i, B-i$ の要素をそれぞれ $\min(n(G,i), n(G,B-i))$ 個 $(1 < i < \lfloor B/2 \rfloor)$，大きさ $B/2$ の要素を $2\lfloor n(G,B/2)/2 \rfloor$ 個 G'' に加えることにより得られる入力，つまり G に対し FFD 法は最適解を出力する． □

3.1.2 定理 1 の証明

証明 容量 $B(\leq 6)$ の容器を考える．

補題2より，FFD 法は $B = 1, 2, 3$ で最適解を出力する．以下では $B = 4, 5, 6$ の場合について，FFD 法が最適解を出力することを示す．

- $B = 4$ のとき

$n(G,1) = n(G,3) = 0$ である入力 G を考える．FFD 法は大きさ 2 の要素を 2 個ずつ容器に埋め込む．よって FFD 法が使用する容器の数は $\lceil n(G,2)/2 \rceil$ 個である．大きさ 2 の要素は一つの容器に高々2 個しか入らないので，少なくとも $\lceil n(G,2)/2 \rceil$ 個以上の容器が必要である．よって FFD 法は G に対し最適解を出力する．

補題4より，任意の入力に対して FFD 法は最適解を出力する．

- $B = 5$ のとき

$n(G,1) = n(G,4) = 0$ であり，かつ $n(G,2) = 0$ または $n(G,3) = 0$ である入力 G を考える．

$n(G,2) = 0$ の場合，FFD 法は大きさ 3 の要素を異なる容器に埋め込む．よって FFD 法が使用する容器の数は $n(G,3)$ 個である．大きさ 3 の要素は一つの容器に高々1 個しか埋め込めないので，少なくとも $n(G,3)$ 個以上の容器が必要である．よって $n(G,2) = 0$ の場合，FFD 法は G に対し最適解を出力する．

$n(G,3) = 0$ の場合，FFD 法は大きさ 2 の要素を 2 個ずつ容器に入れる．よって FFD 法が使用する容器の数は $\lceil n(G,2)/2 \rceil$ 個である．大きさ 2 の要素は一つの容器に高々2 個しか埋め込めないので，少なくとも $\lceil n(G,2)/2 \rceil$ 個以上の容器が必要である．よって $n(G,3) = 0$ の場合，FFD 法は G に対し最適解を出力する．

したがって，FFD 法は G に対し最適解を出力する．補題4から，FFD 法は任意の入力に対して，最適解を出力する．

- $B = 6$ のとき

$n(G,1) = n(G,5) = 0$ であり，かつ，$n(G,2) = 0$ または $n(G,4) = 0$，かつ $n(G,3) \leq 1$ である入力 G を考える．

$n(G,2) = 0$ の場合，各要素は別々の容器に埋め込まれる．明らかに FFD 法は最適解を出力する．

$n(G,4) = 0$ の場合を考える．$n(G,3) = 0$ ならば，FFD 法は大きさ 2 の要素を 3 個ずつ容器に埋め込む．よって FFD 法が使用する容器の数は $\lceil n(G,2)/3 \rceil$ 個である．$n(G,3) = 1$ ならば，FFD 法は大きさ 3 の要素が埋め込まれた容器に大きさ 2 の要素を 1 個埋め込んだ後，残りを 3 個ずつ同じ容器に入れる．よって FFD 法で使用する容器数は $\lceil (n(G,2)+2)/3 \rceil$ となる．入力中の要素の大きさの総和から，いずれの場合も必要な容器数は $\lceil (2n(G,2)+3n(G,3))/6 \rceil$ 個以上である．これは FFD 法が使用する容器の数に一致する．したがって FFD 法は G に対し最適解を出力する．

以上により，FFD 法は G に対し最適解を出力する．補題4から，FFD 法は任意の入力に対して，最適解を出力する． □

3.2 容量 7 以上での FFD 法に対する反例

前節で，容量を 6 以下に固定すると FFD 法が最適解を出力することを示したが，容量を 7 以上に固定した場合は最適解を出力しない例が存在する．

6 個の要素からなる入力 $G = \{g_1, g_2, g_3, g_4, g_5, g_6\}$ を考える．ただし各要素の大きさは，容量に応じて，次のように定める．

- B が 7 以上の奇数のとき

$s(g_1) = s(g_2) = (B-1)/2, s(g_3) = s(g_4) = (B-3)/2$, $s(g_5) = s(g_6) = 2$

- B が 8 以上の偶数のとき

$s(g_1) = B/2, s(g_2) = s(g_3) = B/2-1$, $s(g_4) = B/2-2, s(g_5) = s(g_6) = 2$

いずれの場合も，FFD 法は $\{g_1, g_2\}, \{g_3, g_4, g_5\}, \{g_6\}$ という解を出力するが，$\{g_1, g_4, g_5\}, \{g_2, g_3, g_6\}$ という解が存在し，FFD 法が最適解を出力しないことがわかる．

3.3 容量 7 の場合の改良アルゴリズム

FFD 法は容量 7 では必ずしも最適解を出力しない．ここでは，容量 7 で最適解を出力するアルゴリズム FFD-r-7 法を定義する．

定義 1 FFD-r-7 法
容量 7 の時の FFD 法の入力を G とする．
$M_7 = \max\left(\min\left(\left\lfloor\frac{n(G,2)-n(G,5)}{2}\right\rfloor, n(G,3)-n(G,4)\right), 0\right)$
とする．入力 G から大きさ 2 の要素を $2M_7$個，大きさ 3 の要素を M_7個取り除いた残りを G'とする．G'に対して FFD 法を行い，得られた解に大きさ 3 の要素が 1 個，大きさ 2 の要素が 2 個入った容器を M_7個加え，G の解として出力する．

定理 2 FFD-r-7 法は最適解を出力する．

証明は 3.3.2節に示す．

3.3.1 証明に必要な補題

今節で示す補題は容量 7 の場合に成立するものである．

補題 5 容量を 7 とする．$n(G,1) = n(G,6) = 0$，かつ，$n(G,2) = 0$ または $n(G,3) = 0$，かつ，$n(G,2) = 0$ または $n(G,5) = 0$，かつ，$n(G,3) = 0$ または $n(G,4) = 0$，である任意の入力を G とする．FFD 法は G に対し最適解を出力する．

証明 $n(G,2) = 0$ と仮定する．$n(G,3) = 0$ ならば，G は大きさ 4，5 の要素のみからなり，いかなる解においても，各要素は単独で容器に埋め込まれる．明らかに FFD 法は最適解を出力する．$n(G,3) \neq 0$ ならば，G は大きさ 3，5 の要素のみからなる．いかなる解においても，大きさ 5 の要素は単独で容器に埋め込まれ，大きさ 3 の要素は各容器には高々2 個ずつ埋め込まれる．明らかに FFD 法は最適解を出力する．

$n(G,3) = 0$ と仮定する．$n(G,2) \neq 0$ の場合を考えれば十分であり，G は大きさ 2，4 の要素のみからなるとする．FFD 法では，大きさ 4 の要素を異なる容器に埋め込み，大きさ 2 の要素を，大きさ 4 の要素が埋め込まれた容器に 1 個ずつ埋め込んだ後，残りを 3 個ずつ容器に埋め込む．したがって，FFD 法では $n(G,2) \leq n(G,4)$ の場合，$n(G,4)$ 個，$n(G,2) > n(G,4)$ の場合，$n(G,4) + \lceil (n(G,2) - n(G,4))/3 \rceil$ 個の容器を使用する．また，どのような解においても，大きさ 4 の要素は異なる容器に埋め込まれ，各容器に埋め込まれる要素の大きさの和は高々6 である．したがって，$n(G,2) \leq n(G,4)$ の場合，$n(G,4)$ 個，$n(G,2) > n(G,4)$ の場合，$\lceil (n(G,2) + 2n(G,4))/3 \rceil$ 個以上の容器が必要である．これは，FFD 法で用いる容器数に一致する．よって，FFD 法は G に対し最適解を出力する．□

補題 6 容量を 7 とする．$n(G,2) \leq n(G,5)$ または $n(G,3) \leq n(G,4)$ である任意の入力 G に対して FFD 法は最適解を出力する．

証明 $n(G,2) \leq n(G,5)$ または $n(G,3) \leq n(G,4)$ である任意の入力 G を考える．G より，大きさ 1,6 の要素をすべて，大きさ 2,5 の要素を $\min(n(G,2), n(G,5))$ 個，大きさ 3,4 の要素を $\min(n(G,3), n(G,4))$ 個，取り除くことにより得られる入力を G'とする．G'は補題 5の条件を満足するため FFD 法は G'に対して最適解を出力する．大きさ 1,6 の要素をそれぞれ $n(G,1)$ 個，$n(G,6)$ 個 G'に加えることにより得られる入力を G''とする．補題 2より，G''に対し FFD 法は最適解を出力する．系 1より，大きさ 2,5 の要素をそれぞれ $\min(n(G,2), n(G,5))$ 個，大きさ 3,4 の要素をそれぞれ $\min(n(G,3), n(G,4))$ 個 G''に加えることにより得られる入力，つまり G，に対し FFD 法は最適解を出力する．□

3.3.2 定理 2 の証明

証明 任意の入力 G を考える．$n(G,2) \leq n(G,5)$ または $n(G,3) \leq n(G,4)$ であるならば，$M_7 = 0$ となり，G'は G と同一となる．補題 6より，FFD 法は G に対して最適解を出力する．

したがって，$n(G,2) > n(G,5)$ かつ $n(G,3) > n(G,4)$ であると仮定する．

G'は G より大きさ 2 の要素を $2M_7$個，大きさ 3 の要素を M_7個取り除くことにより得られる．G'より大きさ 1,6 の要素をすべて，大きさ 2,5 の要素を $n(G,5)$ 個，大きさ 3,4 の要素を $n(G,4)$ 個，取り除くことにより得られる入力を G^*とする．G^*は，大きさ 2,3 の要素以外は含まず，さらに，$n(G^*,3) = 0$，または，$n(G^*,2) \leq 1$，である．$n(G^*,3) = 0$ の場合，FFD 法は G^*を $\lceil n(G^*,2)/3 \rceil$ 個の容器に埋め込む．また，どのような解においても，$\lceil n(G^*,2)/3 \rceil$ 個の容器が必要である．$n(G^*,2) \leq 1$ の場合，FFD 法は G^*を $\lceil (n(G^*,2) + n(G^*,3))/2 \rceil$ 個の容器に埋め込む．また，どのような解においても，$\lceil (2n(G^*,2) + 3n(G^*,3))/6 \rceil$ 個の容器が必要である．これは，FFD 法で用いる容器数に一致する．したがって，いずれの場合も，G^*に対し FFD 法が最適解を出力する．補題 2，系 1より，FFD 法は，G'に対して最適解を出力する．

したがって，$M_7 = 0$ の場合，G は G'と一致し，FFD-r-7 法は，G に対して最適解を出力する．

$M_7 > 0$ の場合を考える．G'の最適解が a 個の容器を用いたと仮定する．FFD-r-7 法は，G'の最適解に，M_7個の容器を加えた解を G の解として出力する．したがって，FFD-r-7 法は，G に対して $a + M_7$個の容器を用いた解を出力する．

大きさ 2 の要素 2 個，大きさ 3 の要素 1 個が埋め込まれた容器数が最大である G の最適解を考える．そのとき，そのような容器数が M_7未満であると仮定する．

大きさ 5 の要素と異なる容器に埋め込まれた大きさ 2 の要素数は $2M_7$個以上，大きさ 4 の要素と異なる容器に埋め込まれた大きさ 3 の要素数は M_7個以上存在する．したがって，大きさ 2 の要素 2 個，大きさ 3 の要素 1 個が埋め込まれた容器，大きさ 2 の要素 1 個，大きさ 5 の要素 1 個が埋め込まれた容器，のいずれにも含まれない大きさ 2 の要素が 2 個以上存在する．したがって，そのような大きさ 2 の要素を g_x, g_yとする．同様に大きさ 2 の要素 2 個，大きさ 3 の要素 1 個が埋め込まれた容器，大きさ 3 の要素 1 個，大きさ 4 の要素 1 個が埋め込まれた容器，のいずれにも含まれない大きさ 3 の要素が 1 個以上存在する．そのような大きさ 3 の要素を g_zとする．

g_x, g_y, g_zがそれぞれ異なる容器 G_x, G_y, G_zに埋め込まれ，それら容器内には大きさ 2，3 の要素は他に埋め込まれていないと仮定する．G_x, G_yに埋め込まれた他の要素の大きさは 1，または 4 である．G_zに埋め込まれた他の要素の大きさはすべて 1 である．したがって，G_x, G_y, G_z中の g_x, g_y, g_z以外の要素は高々2 個の容器に埋め込むことができる．したがって，g_x, g_y, g_zを同一容器，他の要素を高々2 個の容器に埋め込むことにより，容器数を増やさずに，大きさ 2 の要素 2 個，大きさ 3 の要素 1 個が埋め込まれた容器数がより多い解が構成でき仮定に矛盾する．また，g_x, g_y, g_zが 2 容器 G_x, G_yに埋め込まれていたとすると，G_x, G_yに埋め込まれた他の要素の大きさの和は 7 以下であり，g_x, g_y, g_zを同一容器に，他の要素を高々1 個の容器に埋め込むことにより，やはり，容器数を増やさずに，大きさ 2 の要素 2 個，大きさ 3 の要素 1 個が埋め込まれた容器数がより多い解が構成でき矛盾する．

したがって，大きさ 2 の要素 2 個，大きさ 3 の要素 1 個が埋め込まれた容器数が M_7以上である G の最適解が存在する．また，その解から大きさ 2 の要素 2 個，大きさ 3 の要素 1 個が埋め込まれた容器を M_7取り除いた解は G'の解である．したがって，G の最適解が $a + M_7$個未満の容器を用いたとすると，a 個未満の容器を用いた G'に対する解が存在し，G'の最適解が a 個の容器を用いることに矛盾する．したがって，G の最適解が $a + M_7$個以上の容器を用いる．FFD-r-7 法の解は G に対する最適解である． □

3.4 容量 8 の場合の改良アルゴリズム

容量 7 の場合と同様に，FFD 法は容量 8 では必ずしも最適解を出力しない．ここでは，容量 8 で最適解を出力するアルゴリズム FFD-r-8 法を定義する．

定義 2 FFD-r-8 法
容量 8 の時の FFD 法の入力を G とする．
$M_8 = \max\left(\min\left(\left\lfloor\frac{n(G,3)-n(G,5)}{2}\right\rfloor, n(G,2)-n(G,6)\right), 0\right)$
とする．入力 G から大きさ 3 の要素を $2M_8$個，大きさ 2 の要素を M_8個取り除いた残りを G'とする．G'に対して *FFD* 法を行い，得られた解に大きさ 3 の要素が 2 個，大きさ 2 の要素が 1 個入った容器を M_8個加え，G の解として出力する．

定理 3 FFD-r-8 法は最適解を出力する．

証明は 3.4.2節に示す．

3.4.1 証明に必要な補題

今節で示す補題は容量 8 の場合に成立するものである．

補題 7 容量を 8 とする．$n(G,1) = n(G,7) = 0$，かつ，$n(G,2) = 0$ または $n(G,3) = 0$，かつ，$n(G,2) = 0$ または $n(G,6) = 0$，かつ，$n(G,3) = 0$ または $n(G,5) = 0$，かつ，$n(G,4) \leq 1$ である任意の入力に対し，FFD 法は最適解を出力する．

証明 $n(G,2) = 0$ と仮定する．$n(G,3) = 0$ の場合を考える．このとき，G は大きさ 4,5,6 の要素のみからなり，しかも大きさ 4 の要素は高々1 個なので，いかなる解においても，各要素は単独で容器に埋め込まれる．明らかに FFD 法は最適解を出力する．$n(G,3) \neq 0$ の場合を考える．このとき，G は大きさ 3,4,6 の要素のみからなる．FFD 法では，大きさ 6 の要素は単独で容器に埋め込まれる．$n(G,4) = 1$ ならば，大きさ 3 の要素は大きさ 4 の要素が埋め込まれた容器に 1 個埋め込まれ，残りは 2 個ずつ埋め込まれる．$n(G,4) = 0$ ならば，大きさ 3 の要素は 2 個ずつ埋め込まれる．いずれの場合も，FFD 法では $n(G,6) + \lceil(n(G,3) + n(G,4))/2\rceil$ 個の容器を使用する．また，いかなる解においても，大きさ 6 の要素は単独で容器に埋め込まれ，大きさ 4,3 の要素は各容器には高々2 個ずつ埋め込まれる．したがって，$n(G,6) + \lceil(n(G,3) + n(G,4))/2\rceil$ 個以上の容器が必要である．明らかに FFD 法は最適解を出力する．よって，$n(G,2) = 0$ ならば，FFD 法は最適解を出力する．

$n(G,3) = 0$ と仮定する．$n(G,2) \neq 0$ の場合を考えれば十分である．G は大きさ 2,4,5 の要素のみからなる．$n(G,4) = 0$ ならば，FFD 法では，大きさ 5 の要素は異なる容器に埋め込まれ，大きさ 2 の要素は，大きさ 5 の要素が埋め込まれた容器に 1 個ずつ埋め込まれ，残りは 4 個ずつ埋め込まれる．したがって，FFD 法では $n(G,2) \leq n(G,5)$ の場合 $n(G,5)$ 個，$n(G,2) > n(G,5)$ の場合 $n(G,5) + \lceil (n(G,2) - n(G,5))/4 \rceil$ 個の容器を使用する．ところで，いかなる解においても，大きさ 5 の要素は異なる容器に埋め込まれる．また，大きさ 5 の要素が埋め込まれた容器に埋め込まれる要素の大きさの総和は高々7 であり，入力中の要素の大きさの総和は $2n(G,2) + 5n(G,5)$ である．したがって，容器数を x とすると，$8x - n(G,5) \geq 2n(G,2) + 5n(G,5)$ が成立する．よって，$n(G,2) \leq n(G,5)$ の場合 $n(G,5)$ 個，$n(G,2) > n(G,5)$ の場合 $\lceil (n(G,2) + 3n(G,5))/4 \rceil$ 個以上の容器が必要である．これは FFD 法で用いる容器数に一致する．$n(G,4) = 1$ ならば，FFD 法では，大きさ 5 の要素は異なる容器に埋め込まれ，大きさ 2 の要素は，大きさ 5 の要素が埋め込まれた容器に 1 個ずつ埋め込まれ，大きさ 4 の要素が埋め込まれた容器に 2 個埋め込まれる．残りは 4 個ずつ埋め込まれる．したがって，FFD 法では $n(G,2) \leq n(G,5)+2$ の場合 $n(G,5)+1$ 個，$n(G,2) > n(G,5)+2$ の場合 $n(G,5) + \lceil (n(G,2) - n(G,5) - 2)/4 \rceil + 1$ 個の容器を使用する．また，いかなる解においても，大きさ 4,5 の要素は異なる容器に埋め込まれ，大きさ 5 の要素が埋め込まれた容器に埋め込まれる要素の大きさの和は高々7 である．したがって，$n(G,2) \leq n(G,5) + 2$ の場合 $n(G,5) + 1$ 個，$n(G,2) > n(G,5)+2$ の場合 $\lceil (n(G,2)+3n(G,5)+2)/4 \rceil$ 個以上の容器が必要である．これは FFD 法で用いる容器数に一致する．よって，$n(G,3) = 0$ ならば，FFD 法は最適解を出力する．

以上より，FFD 法は G に対し最適解を出力する．□

補題 8 容量を 8 とする．$n(2) \leq n(6)$ または $n(3) \leq n(5)$ である任意の入力 G に対して FFD 法は最適解を出力する．

証明 $n(G,2) \leq n(G,6)$ または $n(G,3) \leq n(G,5)$ である任意の入力 G を考える．G より，大きさ 1,7 の要素をすべて，大きさ 2,6 の要素を $\min(n(G,2), n(G,6))$ 個，大きさ 3,5 の要素を $\min(n(G,3), n(G,5))$ 個，大きさ 4 の要素を $n(G,4) - 2\lfloor n(G,4)/2 \rfloor$ 個，取り除くことにより得られる入力を G' とする．G' は補題 7の条件を満足するため，FFD 法は G' に対して最適解を出力する．大きさ 1,7 の要素をそれぞれ $n(G,1)$ 個，$n(G,7)$ 個 G' に加えることにより得られる入力を G'' とする．補題 2より，G'' に対し FFD 法は最適解を出力する．系 1より，大きさ 2,6 の要素をそれぞれ $\min(n(G,2), n(G,6))$ 個，大きさ 3,5 の要素をそれぞれ $\min(n(G,3), n(G,5))$ 個，大きさ 4 の要素を $n(G,4) - 2\lfloor n(G,4)/2 \rfloor$ 個 G'' に加えることにより得られる入力に対し FFD 法は最適解を出力する．つまり G に対し FFD 法は最適解を出力する．□

3.4.2 定理 3 の証明

証明 任意の入力 G を考える．$n(G,2) \leq n(G,6)$ または $n(G,3) \leq n(G,5)$ であるならば，$M_8 = 0$ となり，G' は G と同一となる．補題 8より，FFD 法は G に対して最適解を出力する．

したがって，$n(G,2) > n(G,6)$ かつ $n(G,3) > n(G,5)$ であると仮定する．

G' は G より大きさ 2 の要素を M_8 個，大きさ 3 の要素を $2M_8$ 個取り除くことにより得られる．G' より大きさ 1,7 の要素をすべて，大きさ 2,6 の要素を $n(G,6)$ 個，大きさ 3,5 の要素を $n(G,5)$ 個，大きさ 4 の要素を $n(G,4) - 2\lfloor n(G,4)/2 \rfloor$ 個取り除くことにより得られる入力を G^* とする．G^* には，大きさ 2,3,4 の要素以外は含まず，さらに，$n(G^*,2) = 0$ または $n(G^*,3) \leq 1$，かつ，$n(G^*,4) \leq 1$ である．

$n(G^*,2) = 0$ の場合，FFD 法は G^* を $\lceil (n(G^*,3) + n(G^*,4))/2 \rceil$ 個の容器に埋め込む．また，どのような解においても，$\lceil (n(G^*,3) + n(G^*,4))/2 \rceil$ 個の容器が必要である．$n(G^*,3) \leq 1$ の場合，FFD 法は G^* を $\lceil (n(G^*,2) + 2n(G^*,3) + 2n(G^*,4))/4 \rceil$ 個の容器に埋め込む．また，どのような解においても，$\lceil (2n(G^*,2) + 3n(G^*,3) + 4n(G^*,4))/8 \rceil$ 個の容器が必要である．これは，FFD 法で使用する容器数に一致する．したがって，いずれの場合も，G^* に対し FFD 法が最適解を出力する．補題 2，系 1より，FFD 法は，G' に対して最適解を出力する．

したがって，$M_8 = 0$ の場合，G は G' と一致し，FFD-r-8 法は，G に対して最適解を出力する．

$M_8 > 0$ の場合を考える．G' の最適解が a 個の容器を用いたと仮定する．FFD-r-8 法は，G' の最適解に，M_8 個の容器を加えた解を G の解として出力する．したがって，FFD-r-8 法は，G に対して $a + M_8$ 個の容器を用いた解を出力する．

大きさ 2 の要素 1 個，大きさ 3 の要素 2 個が埋め込まれた容器数が最大である G の最適解を考える．そのとき，そのような容器数が M_8 未満であると仮定する．

大きさ 6 の要素と異なる容器に埋め込まれた大きさ 2 の要素数は M_8 個以上，大きさ 5 の要素と異なる容器に埋め込まれた大きさ 3 の要素数は $2M_8$ 個以上存在する．したがって，大きさ 2 の要素 1 個，大きさ 3 の要素 2 個が

埋め込まれた容器，大きさ2の要素1個，大きさ6の要素1個が埋め込まれた容器，のいずれにも含まれない大きさ2の要素が1個以上存在する．したがって，そのような大きさ2の要素をg_xとする．同様に大きさ2の要素1個，大きさ3の要素2個が埋め込まれた容器，大きさ3の要素1個，大きさ5の要素1個が埋め込まれた容器，のいずれにも含まれない大きさ3の要素が2個以上存在する．そのような大きさ3の要素をg_y, g_zとする．

g_x, g_y, g_zがそれぞれ異なる容器G_x, G_y, G_zに埋め込まれ，それら容器内には大きさ2，3の要素は他に埋め込まれていないと仮定する．G_xに埋め込まれた他の要素の大きさは1，または4，または5である．G_y, G_zに埋め込まれた他の要素の大きさは1，または4である．したがって，G_x, G_y, G_z中のg_x, g_y, g_z以外の要素は高々2個の容器に埋め込むことができる．したがって，g_x, g_y, g_zを同一容器，他の要素を高々2個の容器に埋め込むことにより，容器数を増やさないで，大きさ2の要素1個，大きさ3の要素2個が埋め込まれた容器数がより多い解が構成でき仮定に矛盾する．また，g_x, g_y, g_zが2容器G_x, G_yに埋め込まれていたとすると，G_x, G_yに埋め込まれた他の要素の大きさの和は8以下であり，g_x, g_y, g_zを同一容器に，他の要素を高々1個の容器に埋め込むことにより，やはり，容器数を増やさないで，大きさ2の要素1個，大きさ3の要素2個が埋め込まれた容器数がより多い解が構成でき矛盾する．

したがって，大きさ2の要素1個，大きさ3の要素2個が埋め込まれた容器数がM_8以上であるGの最適解が存在する．また，その解から大きさ2の要素1個，大きさ3の要素2個が埋め込まれた容器をM_8取り除いた解はG'の解である．したがって，Gの最適解が$a + M_8$個未満の容器を用いたとするとa個未満の容器を用いたG'に対する解が存在しG'の最適解がa個の容器を用いることに矛盾する．したがって，Gの最適解が$a + M_8$個以上の容器を用いる．FFD-r-8の解はGに対する最適解である．□

4 結論

本文では，整数ビンパッキング問題の近似アルゴリズムであるFFD法が，容量を6以下に制限した時に任意の入力に対して最適解を出力することを示した．また，FFD法を改良することによって，容量が7,8の場合に任意の入力に対して最適解を出力するアルゴリズムを提案した．入力数が9以上のルックアップテーブル型論理ブロックは現実的でないと考えられるため，2段論理回路の1段目を構成するゲートを論理ブロックに割り当てる場合の論理ブロック数最小化問題を，提案したアルゴリズムにより，実用範囲において効率よく解くことができる．

共通信号線を考慮した論理ブロック数最小化問題は，整数ビンパッキング問題を一般化した，集合ビンパッキング問題に定式化される．集合ビンパッキング問題についても様々な研究がなされている[5][6][8]．また，多段論理回路に対する論理ブロック数最小化問題に関する研究が必要である．

謝辞

なお，本研究の一部は，文部省科学研究費一般研究(B)(05452209)及び東工大CAD21研究体の援助による．

参考文献

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

[2] K. Chung and J. Rose, "TEMPT: Technology Mapping for the Exploration of FPGA Architectures with Hard-Wired Connections", Proc. 29th DAC, pp.361-367, 1992.

[3] R. Francis, J. Rose, and Z. Vranesic, "Chortle-crf: Fast Technology Mapping for Lookup Table-Based FPGAs", Proc. 28th DAC, pp. 227 - 233, 1991.

[4] M. R. Garey and D. S. Johnson, "Computers and Intractability", W. H. FREEMAN AND COMPANY, 1983.

[5] T. Izumi, T. Yokomaru, A. Takahashi and Y. Kajitani, "Cube-Packing Problem with Fixed Bin-Capacity($\geq$ 3) is NP-complete", Technical Report FTS94-44, VLD94-63,IEICE, 1994.

[6] T. Izumi, T. Yokomaru, A. Takahashi and Y. Kajitani, "Computational Complexity Map of the Set Bin-Packing Problem", Proc. 1995 IEICE GENERAL CONF., A-110, 1995.

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

[8] R. Murgai, R. K. Brayton and A. Sangiovanni-Vincentelli, "Cube-packing and Two-level Minimization", "International Conference on CAD", Proc. ICCAD,IEEE, pp.115-122, 1993.