アルゴリズム 75－6
(2000. 11. 10)

ネット割当てアルゴリズムの改良

小野 孝男, 平田 富夫
名古屋大学工学部電子工学科
464-8603 名古屋市千種区不老町
{ono,hirata}@nuee.nagoya-u.ac.jp

概要 本論文では論理エミュレータにおけるネット割当問題について考察する. この問題はボードレベル配線問題とも呼ばれている. 論理エミュレータのボードには複数の FPGA とクロスバが搭載されており, 各 FPGA はすべてのクロスバと配線によって接続されている. ネット割当て問題とは FPGA 間の配線要求であるネットに従って該当する FPGA 間を接続するため, 各ネットを適切なクロスバに割当てる問題である. この問題は一般には NP-完全であることが知られている. これに対し, すべてのネットが 2 端子ネットであり, しかも FPGA とクロスバが偶数本の配線で接続されている特殊な場合に対しては多項式時間で解くことが可能である. 本論文ではこの場合に対し実行時間を $O(n^2)$ から $O(n^2/c)$ に改良する. ここで n はネットの数, c はクロスバの数である.

キーワード: 論理エミュレータ, ネット割当て問題, 辺彩色, 均等辺彩色

An Improved Algorithm for the Net Assignment Problem

Takao Ono, Tomio Hirata
Department of Electronics, School of Engineering, Nagoya University
Furo-cho, Chikusa-ku, Nagoya 464-8603, Japan
{ono,hirata}@nuee.nagoya-u.ac.jp

abstract In this paper, we consider the net assignment problem in the logic emulation system. This problem is also known as the board-level-routing problem. There are field programmable logic arrays (FPGAs) and crossbars on an emulator board. Each FPGA is connected to each crossbar. Connection requests between FPGAs are called nets, and FPGAs are interconnected through crossbars. We are required to assign each net to the suitable crossbar. This problem is known to be NP-complete in general. A polynomial time algorithm is known for a certain restricted case, in which we treat only 2-terminal nets. In this paper we propose a new polynomial time algorithm for this case.

keyword: logic emulator, net assignment problem, edge coloring, nearly equitable edge coloring

1 はじめに

論理検証は特定用途向け IC (ASIC) や CPU チップなどの VLSI 設計において重要なステップである. このステップでは, 設計した回路が与えられた論理仕様を満たしているかどうかを検証する. これまで論理検証には論理シミュレータと呼ばれるソフトウエアが用いられてきた. しかし, 最近では回路が複雑かつ大規模化したため, 論理シミュレータでは計算時間が膨大になり論理検証が十分に行えなくなっている. そのため, FPGA を用いた論理エミュレータと呼ばれるハードウェアが用いられている.

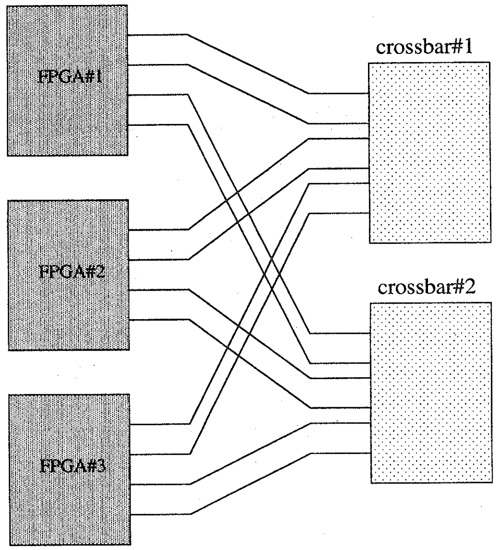

図 1: An example of a logic emulator

論理エミュレータ上には多くの FPGA とクロスバが配置されている. クロスバは一種のスイッチング回路であり, I/O ピンを任意に接続することができる. 論理エミュレータの例を図 1 に示す.

設計した論理回路を複数のブロックに分割して FPGA に書き込み, ブロック間の接続は FPGA の I/O ピンを相互に接続することで実現する. FPGA の I/O ピン間の接続要求をネットと呼び, クロスバにネットを割当てることによって相互接続を実現する. ネット割当て問題はネットの集合であるネットリストが与えられたときに適切に個々のネットをクロスバに割当てる問題である. この問題は一般には NP-完全である [5]. これに対し, すべてのネットが 2 端子ネットであり, しかも FPGA とクロスバが偶数本の配線で接続されている特殊な場合に対しては多項式時間で解くことが可能である. 本論文ではこの場合に対し実行時間を $O(n^2)$ から $O(n^2/c)$ に改良する. ここで n はネットの数, c はクロスバの数である.

2 準備

ネット割当て問題を定式化するため, 以下の記法を用いる.

$\mathcal{F} = \{F_1, F_2, \ldots, F_f\}$ と $\mathcal{C} = \{C_1, C_2, \ldots, C_c\}$ をそれぞれ FPGA 及びクロスバの集合とする. FPGA とクロスバの間は m 本ずつ配線されている. 従って FPGA は mc 本の, クロスバは mf 本の I/O ピンを持つ.

ネット $N_i \subset F$ は接続したい FPGA の集合である. N_i の要素数 $|N_i|$ をこのネットの大きさと呼ぶ. t 端子ネットとは大きさが t のネットである. ネットリストはネットの多重集合 $\mathcal{N} = \{N_1, N_2, \ldots, N_n\}$ である.

ネット割当て問題の解はネットからクロスバへの割当て (写像) $a\colon \mathcal{N} \to \mathcal{C}$ である. 但し, この割当てにおいてどのクロスバにも同じ FPGA を含むネットは高々 m 個しか割当ててはならない.

この問題は $m = 1$ でも NP-完全であることがわかる. 問題例が与えられたときに, 次のようにグラフ $G = (V, E)$ を構成する. ネットリストの各ネットに G の頂点を対応させる. つまり $V = \mathcal{N}$ で

ある. 2 ネット N_i, N_j が同じ FPGA を共有する, つまり $N_i \cap N_j \neq \emptyset$ であるときに対応する 2 頂点の間に辺を持つ. また, クロスバにはグラフの頂点に対する色を対応させる. $m = 1$ であるから 2 ネット N_i, N_j が FPGA を共有するならば対応する頂点は異なる色でなければならない. このことからネット割当て問題はグラフの彩色問題に還元することができる. 逆に, 任意のグラフ G に対して上記の構成によって G が得られるようなネットリストが存在することがわかる. つまり, $m = 1$ に対するネット割当て問題は彩色問題と等価であり, 従って NP-完全である.

ネット割当て問題の入力は, 厳密には論理エミュレータとネットリストである. しかし, 実用上の観点から以下では論理エミュレータは固定であるものとしてネットリストのみを入力として考える.

一般的なネット割当て問題に対してはこれまでにもいくつかのグリーディアルゴリズムが知られている [1, 4, 3, 2]. これらはすべて 1 ステップごとに評価関数の値によってネットとそのネットを割当てるべきクロスバを選んでいくというアルゴリズムである.

一方, ネットリストのすべてのネットが 2 端子であり, かつ m が偶数のときには n ネットからなるネットリストに対して $O(n^2)$ 時間アルゴリズムが存在する [5]. 3 節では, この場合に対し, より高速なアルゴリズムを与える.

3 制限された場合に対する新しい近似アルゴリズム

Mak と Wong は, 2 節の最後に述べたような制約のある場合には, ネット割当て問題が多項式時間で解けることを示した [5]. このアルゴリズムは $O(n^2)$ 時間で実行できる. この節では彼らのアルゴリズムを修正し, 実行時間を $O(n^2/c)$ に削減可能であることを示す.

まず, 制限されたネット割当て問題を次のように記述する.

Problem 1 (m-辺彩色問題). 自己閉路を持たない多重グラフ $G = (V, E)$ ($|V| = f$, $|E| = n$) と色数 c が与えられたときに, どの頂点についても接続する同じ色の辺がたかだか m 本であるように辺に色を与えよ.

問題 1 において, 頂点, 辺, 色はそれぞれネット割当て問題における FPGA, ネット, クロスバに対応する.

頂点 v の次数を $d(v)$ で表し, その最大値 $\Delta = \max_{v \in V} d(v)$ を G の次数とする. 色 α を与えられた辺を α-辺と呼ぶ. $d_G(v, \alpha)$ は頂点 v に接続する α-辺の数を表し, グラフ G における α-辺の本数を $e_G(\alpha)$ で表す. 文脈から明らかなときには添字 G を省略する. これらの記法を使って問題 1 を次のように表現しなおす.

Problem 2 (m-辺彩色問題). グラフ G と c 色の色の集合 C が与えられたときに, 次の "容量制約" を満たすように辺に色を与えよ:

$$\text{任意の頂点 } v \in V \text{ と色 } \alpha \in C \text{ に対して } d(v, \alpha) \leq m. \tag{1}$$

m が偶数でかつ $\Delta \leq mc$ であるときには任意の m-辺彩色問題の問題例に対して解が存在する [5]. 以下ではこの仮定が満たされているものとする.

3.1 新しいアルゴリズム

Mak と Wong は m-辺彩色問題に対して $O(n^2)$ 時間アルゴリズムを示した [5]. 我々のアルゴリズムが Mak-Wong のアルゴリズムと異なるのは, 実際には次に示す平衡 m-辺彩色問題という, 問題 2 に制約を加えた問題を解いているという点である:

BALANCED-m-EDGE-COLORING$(G, \mathcal{C})$
$\triangleright$ $\mathcal{C} = \{C_1, C_2, \ldots, C_c\}$ に含まれる色を使って $G = (V, E)$ に対する平衡 m-辺彩色を解く
$\lceil m/c \rceil$ (または $\lfloor m/c \rfloor$) 本の辺が同じ色になるように m 本の辺に色 $C_1, C_2, \ldots, C_c$ を順に割当てる.
while $d(v, \alpha) > m$ であるような頂点 $v \in V$, 色 $\alpha \in \mathcal{C}$ が存在する間
    $d(v, \beta) < m$ である色 $\beta \in \mathcal{C}$ を選ぶ.
    RECOLOR$(G_{\alpha\beta}, \alpha, \beta)$

RECOLOR(G, α, β)
$\triangleright$ $e(\beta) \le e(\alpha) \le e(\beta) + 1$ が満たされるように G の辺に色 α, β を与える.
$x \leftarrow \alpha$, $y \leftarrow \beta$ とする.
for G の連結成分 H に対し
    RECOLOR-COMPONENT(H, x, y).
    if $|E_H|$ が奇数 **then**
        x と y を交換する.

RECOLOR-COMPONENT(G, α, β)
$\triangleright$ $e(\beta) \le e(\alpha) \le e(\beta) + 1$ が満たされるように G の辺に色 α, β を与える.
if G が奇数次の頂点を持つ **then**
    奇数次の頂点すべてに隣接する頂点 w を追加してグラフ G' を作る.
    頂点 w からオイラー閉路をたどり, その順に交互に色 β, α を辺に与える.
else
    if G が奇数本の辺を持つ **then**
        次数が $2m$ でない頂点 u を選ぶ.
    else
        u として任意の頂点を選ぶ.
    頂点 u からオイラー閉路をたどり, その順に交互に色 α, β を辺に与える.

図 2: 平衡 m-辺彩色に対するアルゴリズム

Problem 3 (平衡 m-辺彩色). この問題は辺彩色が次の “平衡制約” を満たさなければならないことを除いて問題 2 と同じである.

$$\text{任意の色 } \alpha, \beta \text{ に対して } |e(\alpha) - e(\beta)| \le 1. \tag{2}$$

平衡制約は均等辺彩色に対する Nakano-Suzuki-Nishizeki のアルゴリズムにヒントを得たものである.

Mak-Wong のアルゴリズムは初期割当てとしてすべての辺に同じ色を割当て, すべての頂点と色に対して容量制約が満たされるまで辺の色を更新する. これに対し, 図 2 に示す我々のアルゴリズムでは初期割当てにおいて個々の色を $\lceil m/c \rceil$ (または $\lfloor m/c \rfloor$) 本の辺に割当て, 平衡制約を満たすように Mak-Wong のアルゴリズムにおける色の更新を行う. ここで初期解において平衡制約 (2) が満たされていることに注意する. アルゴリズムにおいて, $G_{\alpha\beta}$ は α-辺, β-辺で誘導される G の部分グラフである.

RECOLOR$(G_{\alpha\beta}, \alpha, \beta)$ は任意の頂点 v において容量制約と次の “α, β に対する強平衡制約” を満たすように $G_{\alpha\beta}$ の辺に色 α, β を与えなおす:

$$e_G(\beta) \le e_G(\alpha) \le e_G(\beta) + 1 \tag{3}$$

RECOLOR-COMPONENT(H, α, β) は連結成分 H に対し RECOLOR$(G_{\alpha\beta}, \alpha, \beta)$ と同じ処理を行う.

3.2 アルゴリズムの解析

ここでは図 2 に示したアルゴリズムの正当性と実行時間を示す.

Lemma 1 ([5]). 連結グラフ $H = (V_H, E_H)$ に対して RECOLOR-COMPONENT(H, α, β) を実行すると, 次の条件を満たす彩色が得られる.

1. H のすべての頂点が偶数次でかつ $|E_H|$ が偶数ならば任意の頂点 v に対して $d(v,\alpha)=d(v,\beta)$.

2. H のすべての頂点が偶数次でかつ $|E_H|$ が奇数ならば彩色の開始頂点 u に対しては $d(u,\alpha)-d(u,\beta)=2$, その他の頂点 v に対しては $d(v,\alpha)=d(v,\beta)$.

3. H に奇数次の頂点が存在するならば, 奇数次の頂点 v に対しては $|d(v,\alpha)-d(v,\beta)|=1$, 偶数次の頂点 v に対しては $d(v,\alpha)=d(v,\beta)$.

Lemma 2 ([5]). m は偶数であると仮定する. Recolor-Component(H,α,β) を実行して得られる H の彩色は次の条件を満たす: 任意の頂点 v に対し,

1. 実行前に $d(v,\alpha)+d(v,\beta)\leq 2m$ であれば $d(v,\alpha)\leq m$ かつ $d(v,\beta)\leq m$.

2. 実行前に $d(v,\alpha)+d(v,\beta)\geq 2m$ であれば $d(v,\alpha)\geq m$ かつ $d(v,\beta)\geq m$.

Corollary 1. 補題 2 は Recolor$(G_{\alpha\beta},\alpha,\beta)$ に対しても Recolor-Component(H,α,β) 同様成り立つ.

証明. 連結成分ごとに考えればよい. □

Lemma 3. Recolor-Component(H,α,β) を実行して得られる彩色は強平衡制約 $e_H(\beta)\leq e_H(\alpha)\leq e_H(\beta)+1$ を満たす.

証明. $H=(V_H,E_H)$ が奇数次の頂点を持たなければ補題が成り立つことは明らかである. つまり, $|E_H|$ が偶数本の辺を持てば Recolor-Component における彩色は α で始まり β で終わるため $e_H(\alpha)=e_H(\beta)$ である. $|E_H|$ が奇数ならば彩色は α で始まり α で終わるため $e_H(\alpha)=e_H(\beta)+1$ である.

そこで H が奇数次の頂点を持つときを考える. この場合 Recolor-Component の動作は w から始まる G' のオイラー閉路に交互に色 β, α を与え, その後 w と w に接続するすべての辺を取り除くとみなすことができる. $|E_H|$ が偶数 (従って $|E_{H'}|$ も偶数) ならば $e_{H'}(\alpha)=e_{H'}(\beta)$ である. 奇数次の頂点を持たないときの議論から $d_{H'}(w,\alpha)=d_{H'}(w,\beta)$ であり, 従って w を取り除いても $e_H(\alpha)=e_H(\beta)$ である. 最後に $|E_H|$ が奇数のときは $|E_{H'}|$ も奇数であり, 彩色は β で始まり β で終わるため $e_{H'}(\beta)=e_{H'}(\alpha)+1$ である. ここで彩色は w から始まるため $d_{H'}(w,\beta)=d_{H'}(w,\alpha)+2$ である. 従って w を削除すると $e_H(\alpha)=e_H(\beta)+1$ である. □

Corollary 2. 補題 3 は Recolor$(G_{\alpha\beta},\alpha,\beta)$ に対しても Recolor-Component(H,α,β) 同様成り立つ.

証明. Recolor$(G_{\alpha\beta},\alpha,\beta)$ は各連結成分 H に対して Recolor-Component(H,x,y) を呼び出し, H が奇数本の辺を持つ場合には 2 つの色の役割を交替させる. H が偶数本の辺を持つときには $e_H(\alpha)=e_H(\beta)$ であるため, 奇数本の辺を持つ連結成分のみを考えればよい.

$G_{\alpha\beta}$ には奇数本の辺を持つ連結成分が $H_1, H_2, \ldots, H_s$ の s 個あると仮定する. 奇数の i に対しては $x=\alpha$, $y=\beta$ に対して Recolor-Component(H_i,x,y) を実行するため得られる彩色において $e_{H_i}(\alpha)=e_{H_i}(\beta)+1$ が成り立つ. 偶数の i に対して得られる彩色では $e_{H_i}(\beta)=e_{H_i}(\alpha)+1$. これらの等式をすべて加えると, s が偶数ならば $e_G(\alpha)=e_G(\beta)$, s が奇数ならば $e_G(\alpha)=e_G(\beta)+1$ である. □

Corollary 3. アルゴリズム Balanced-m-Edge-Coloring$(G,\mathcal{C})$ の実行中の任意の時点において, 任意の色 α, β に対して不等式 $|e(\alpha)-e(\beta)|\leq 1$ が成り立つ.

証明. RECOLOR の実行回数に関する帰納法で証明する.

0 回目, つまりアルゴリズムの開始時点においては辺を均等に各色に割当てているため任意の色 α, β に対して $|e(\alpha)-e(\beta)| \leq 1$ である.

RECOLOR を k 回実行した後に補題が成り立つと仮定して $k+1$ 回目の実行を考える. 色 α, β が等式 $e(\alpha)=e(\beta)$ を満たすならば系 2 から RECOLOR を実行したあとにおいても $e(\alpha)=e(\beta)$ が満たされる. 一方 $e(\alpha) \neq e(\beta)$ ならば帰納法の仮定から $|e(\alpha)-e(\beta)|=1$ であり, やはり系 2 から $|e(\alpha)-e(\beta)|=1$ となる. 従って RECOLOR を $k+1$ 回実行したあとにおいても補題は成り立つ. □

ここで次のように頂点 $v \in V$ における超過 $\Phi(v)$ を定義する.

$$\Phi(v) = \sum_{\alpha \in C:\ d(v,\alpha)>m} (d(v,\alpha)-m).$$

ある彩色の超過 Φ はすべての頂点 v における $\Phi(v)$ の和とする. 任意の彩色に対して $\Phi \geq 0$ であり, $\Phi = 0$ である彩色はすべての頂点において容量制約を満たす, つまり m-辺彩色問題の解であることに注意する.

次の補題を使ってアルゴリズムが有限の時間で停止することが証明できる.

Lemma 4. $d(v_0,\alpha) > m$, $d(v_0,\beta) < m$ であるような頂点 v_0, 色 α, β を取る. このとき RECOLOR$(G_{\alpha\beta},\alpha,\beta)$ を実行すると Φ の値は必ず減少する.

証明. RECOLOR$(G_{\alpha\beta},\alpha,\beta)$ を実行する前と実行した後の $\Phi(v)$ の変化量を $\Delta\Phi(v)$ で表す. 以下では RECOLOR の実行前, 実行後に頂点 v に接続する α-辺の本数をそれぞれ $d(v,\alpha)$, $d'(v,\alpha)$ で表す.

$d(v,\alpha) \leq m$ かつ $d(v,\beta) \leq m$ ならば補題 2 より $d'(v,\alpha) \leq m$, $d'(v,\beta) \leq m$ である. 従って $\Delta\Phi(v) = 0-0=0$.

$d(v,\alpha) > m$ かつ $d(v,\beta) \geq m$ ならば $d'(v,\alpha) \geq m$, $d'(v,\beta) \geq m$ となる. 従って $d(v,\alpha)+d(v,\beta) = d'(v,\alpha)+d'(v,\beta)$ より

$$\begin{aligned}\Delta\Phi(v) &= \big[\big(d'(v,\alpha)-m\big)+\big(d'(v,\beta)-m\big)\big]-\big[\big(d(v,\alpha)-m\big)+\big(d(v,\beta)-m\big)\big]\\ &= \big(d'(v,\alpha)+d'(v,\beta)\big)-\big(d(v,\alpha)+d(v,\beta)\big)\\ &= 0.\end{aligned}$$

$d(v,\alpha) > m$ かつ $d(v,\beta) < m$ のときにはさらに 2 つに場合分けをする. まず $d(v,\alpha)+d(v,\beta) \geq 2m$ ならば $d'(v,\alpha) \geq m$, $d'(v,\beta) \geq m$ である. 従って

$$\begin{aligned}\Delta\Phi(v) &= \big[\big(d'(v,\alpha)-m\big)+\big(d'(v,\beta)-m\big)\big]-\big(d(v,\alpha)-m\big)\\ &= \big(d'(v,\alpha)+d'(v,\beta)\big)-d(v,\alpha)-m\\ &= d(v,\beta)-m<0\end{aligned}$$

であり, この場合には $\Phi(v)$ は減少する. 最後に $d(v,\alpha)+d(v,\beta) \leq 2m$ ならば $d'(v,\alpha) \leq m$, $d'(v,\beta) \leq m$ となるので

$$\Delta\Phi(v) = 0-\big(d(v,\alpha)-m\big) = m-d(v,\alpha) < 0.$$

どの場合においても $\Phi(v)$ が増加することはない. さらに, v_0 は最後の 2 つの場合のいずれかにあてはまるため $\Phi(v_0)$ は必ず減少する. 従って Φ は必ず減少する. □

Φ の値は次のように上から押さえることができる:

$$\begin{aligned}\Phi &= \sum_{v\in V}\Phi(v)\\ &= \sum_{v\in V}\sum_{\alpha:\ d(v,\alpha)>m}(d(v,\alpha)-m)\\ &\le \sum_{v\in V}\sum_{\alpha\in C} d(v,\alpha)\\ &= \sum_{v\in V} d(v) = 2|E| = 2n.\end{aligned}$$

補題 4 から RECOLOR を実行すると Φ の値は少なくとも 1 減少する. 従って RECOLOR を高々 $2n$ 回実行すると Φ は必ず 0 になる.

ここで実行時間に関する定理を証明する.

Theorem 1. BALANCED-m-EDGE-COLORING は平衡 m-辺彩色問題を $O(n^2/c)$ 時間で解く.

証明. 系 3 から平衡制約は常に満たされている. そして $\Phi = 0$ となる割当ては平衡 m-辺彩色問題の解であるため, アルゴリズムは RECOLOR を高々 $2n$ 回実行すると解を見付けて停止する.

次に RECOLOR の実行時間を解析する.

Lemma 5. RECOLOR$(G_{\alpha\beta},\alpha,\beta)$ の実行時間は $O(e(\alpha)+e(\beta))$ である.

証明. まず連結グラフ $H=(V_H,E_H)$ に対する RECOLOR-COMPONENT(H,α,β) の実行時間について考える. オイラー閉路をたどる時間はグラフの辺の数に比例する. 従って H のオイラー閉路をたどる時間は $O(|E_H|)$ である. 一方, H' の辺の数は高々 V_H+E_H であるため, オイラー閉路をたどる時間は $O(V_H+E_H)$ となる. どちらの場合であっても RECOLOR-COMPONENT(H,α,β) の実行時間は $O(V_H+E_H)$ である.

$G_{\alpha\beta}$ には $e(\alpha)+e(\beta)$ 本の辺と高々 $|V|=f$ 個の頂点が含まれることから, $G_{\alpha\beta}$ のすべての連結成分に対するこの結果を加えると RECOLOR$(G_{\alpha\beta},\alpha,\beta)$ の実行時間は $O(e(\alpha)+e(\beta)+f)$ である. さらに, 任意のグラフ $G=(V,E)$ に対して $|V|\le 2|E|$ であるから補題は成り立つ. □

平衡制約 (2) を加えたことにより, 次の補題に示すように部分グラフの辺の数を制限することができる.

Lemma 6. アルゴリズムのどの時点においても任意の色 α に対して $e(\alpha)=O(n/c)$.

証明. 任意の色 $x\in C$ に対して $e(\alpha)\ge e(x)$, つまり辺の数が最大となる色を α とする. 平衡制約が満たされているので, 任意の色 x に対して $e(\alpha)\le e(x)+1$ である. ここで $e(\alpha)>n/c+1$ と仮定すると平衡制約から $e(x)\ge e(\alpha)-1>n/c$ であり, 従って

$$\begin{aligned}n=|E| &\ge \sum_{x\in C} e(x)\\ &= e(\alpha)+\sum_{x\in C\setminus\{\alpha\}} e(x)\\ &\ge \left(\frac{n}{c}+1\right)+(c-1)\cdot\frac{n}{c} = n+1\end{aligned}$$

となるが, これは矛盾である. 従って $e(\alpha)\le n/c+1$ より補題は証明された. □

補題 5, 6 から RECOLOR$(G_{\alpha\beta},\alpha,\beta)$ は $O(n/c)$ 時間で実行することができる.

以上のことから BALANCED-m-EDGE-COLORING の実行時間は $O(n\times n/c)=O(n^2/c)$ である. □

4 まとめ

本論文ではネット割当て問題の特殊な場合に対する新しいアルゴリズムを提案し，その実行時間がネット数 n，クロスバ数 c に対して $O(n^2/c)$ であることを示した．Mak, Wong による従来のアルゴリズムの実行時間は $O(n^2)$ である [5]．実際の応用ではエミュレータ上に 100 個以上のクロスバが存在することから，今回提案したアルゴリズムによって実行時間を大幅に減らすことができる．

m が奇数の場合，または 3 端子以上のネットが存在する場合には，[5] と同じように扱うことができる．m が奇数のときには FPGA とクロスバの間の配線を高々 $m-1$ しか使わないように制限して考える．従って m が奇数であっても $\Delta \le (m-1)c$ であれば提案したアルゴリズムを適用して解を求めることができる．多端子ネットが存在する場合には，クロスバを多層に配置し，多端子ネットを複数の 2 端子ネットに分解することによって扱うことができる．詳細は [5] を見てほしい．

謝辞

本研究の一部は立松財団と文部省科学研究費の補助を受けた．

参考文献

[1] M. Butts, J. Batcheller, and J. Vargheseq: "An efficient logic emulation system," in *Proc. IEEE Int. Conf. Computer Design*, pp. 138–141, Oct, 1992.

[2] Daisuke Ito, Takao Ono, and Tomio Hirata, "A Study on the Heuristic Algorithm for the Net Assignment Problem," *Technical Report of IEICE*, Vol. 100, No. 144, COMP2000-14, pp. 1–8.

[3] N. Funabiki, J. Kitamichi, "A neural-greedy combination algorithm for board-level routing in FPGA-based logic emulation systems," *Transcations of IEICE*, Vol. E81-A, No. 5, pp. 866–872, 1998.

[4] S.-S. Lin, Y.-J. Lin, and T.-T. Hwang, "Net assignment for the FPGA-based logic emulation system in the folded-clos network structure," *IEEE Trans. Computer-Aided Design*, Vol. 16, No. 3, pp. 316–320, 1997.

[5] Wai-Kei Mak, D.F. Wong, "On optimal board-level routing for FPGA-based logic emulation," *IEEE Trans. Computer-Aided Design*, Vol. 16, No. 3, pp. 282–289, 1997.

[6] Shin-ichi Nakano, Yasuhiro Suzuki, and Takao Nishizeki, "An algorithm for the nearly equitable edge-coloring of graphs," *Transactions of IEICE*, Vol. J78-D-I, No 5, pp. 437–444, 1995.