

## スタイナ木生成問題に対する遺伝的アルゴリズムを 統合的に開発するフレームワーク

秦 純一† 若林 真一†

† 広島大学大学院工学研究科

〒 739-8527 広島県東広島市鏡山一丁目 4 番 1 号

E-mail: †{hata,wakaba}@bsys.hiroshima-u.ac.jp

あらかし 本稿では、VLSI レイアウト設計における概略配線に対し、遺伝的アルゴリズム (GA) を用いてグラフ上でスタイナ木問題を解くためのフレームワークを提案する。提案手法では GA が目的関数に対してブラインド探索であることを利用し、ユーザが評価関数を設定することにより多くの配線問題を統合的に扱うことを可能にする。また、GA 実行中に各オペレータのパフォーマンスに応じて適応的に交差手法等の適用確率を変更する。提案手法により種々のスタイナ木問題に対して個別に GA を開発する必要がなくなり、また、与えられた問題に適した GA を効率よく開発することが可能になる。本稿では、提案手法の有効性を示すために VLSI レイアウト設計におけるグラフスタイナ木問題に対して提案フレームワークを適用した結果を示す。

キーワード VLSI レイアウト設計, 遺伝的アルゴリズム (GA), スタイナ木問題

## A Generic Framework to Solve Steiner Tree Problems in Graphs using a Genetic Algorithm

Jun'ichi HATA† and Shin'ichi WAKABAYASHI†

† Graduate School of Engineering, Hiroshima University

4-1 Kagamiyama 1 chome, Higashi-Hiroshima, Hiroshima, 739-8527 JAPAN

E-mail: †{hata,wakaba}@bsys.hiroshima-u.ac.jp

**Abstract** This paper proposes a generic framework to solve Steiner tree problems in graphs using an adaptive genetic algorithm in VLSI layout design. The proposed method is based on the fact that a genetic algorithm adopts a blind search strategy for objective functions. By preparing evaluation functions by users, this proposed framework can solve many routing problems in a unified manner. With the proposed method, one can easily obtain an efficient GA for a given routing problem without developing a GA for that particular problem. In this paper, to show the effectiveness of the proposed framework, experimental results for the Steiner problem in a graph are presented.

**Key words** VLSI layout design, Genetic Algorithm, Steiner tree problem

### 1. ま え が き

VLSI レイアウトの設計工程は大きく分けて機能論理設計とレイアウト設計の 2 工程に分けることができる [1]。レイアウト設計においては、従来は配線長の最小化を目的とした設計手法によって仕様で与えられたチップ面積や動作周波数等の制約条件を満たすことができていたが、近年、半導体微細加工技術の進展に伴い、回路遅延における配線遅延の割合が支配的になるにつれて、レイアウト設計における配線遅延を重視した設計手法の開発が重要になってきている [2]。このように、半導体技術の進歩により、LSI 配線問題においては新たな目的関数や制約

を考慮することが必要になり、その度に問題の解法を新たに開発することが必要になる。一方、VLSI 設計において遺伝的アルゴリズム (GA) に基づく多くの手法が提案されている [4],[5]。しかし、GA は問題ごとに個体表現や交差、突然変異手法が異なるため、有効な GA の開発には多大な時間を必要とする。

本稿では、VLSI レイアウト設計における概略配線に対し、GA を用いてグラフ上のスタイナ木問題を解くためのフレームワークを提案する。このフレームワークを用いることによって目的関数や制約の異なる問題ごとに新たな GA を開発する手間を省き、また新たな問題に対しても有効な GA を短時間で開発できることを目的とする。提案手法では GA が目的関数に対し

てブライインド探索であることを利用する。すなわち、GAは個体の評価値を考慮せずに解探索を行うので、評価値計算時に必要となる情報をグラフ上にラベルとして節点やエッジに与え、各問題ごとにユーザが評価値関数を設定することにより多くの配線問題を統一的に扱うことが可能になる。

提案GAアルゴリズムは、現在の世代に対して撰択、交差、突然変異を行い、新しい世代を形成する。遺伝的操作の適用後に各2節点間の最短距離を求めることで配線経路を決定し、それに基づいてユーザが設定した評価値関数によって個体の評価値を求める。さらに得られた評価値に基づいてエリート戦略によるトーナメント撰択を行い、次世代に残す個体を決定する。また、有効な交差、突然変異手法は問題の種類やGA実行中で変わる可能性があるため、複数の交差、突然変異手法をあらかじめ用意しておき、GA実行中に各オペレータのパフォーマンスに応じて適応的に各交差手法等の適用確率を変更する。

本稿では、提案フレームワークの有効性を示すために、VLSIレイアウト設計におけるグラフスタイナ木問題に対する実験結果を示す。

本稿の構成は以下の通りである。まず2.ではスタイナ木問題と適応的GAについて述べる。3.では提案フレームワークについて、4.では提案手法の詳細について述べ、5.では計算機実験による提案手法の評価を示し、最後に6.で本稿をまとめる。

## 2. 準備

### 2.1 スタイナ木問題

以下では、スタイナ木問題の代表的な例としてグラフスタイナ木問題、直交スタイナ木問題、VLSI配線問題を定式化する。

#### 2.1.1 グラフスタイナ木問題 (SPG)

連結無向グラフ  $G=(V, E)$ 、正のエッジコスト関数  $c: E \rightarrow R_+$ 、部分節点集合  $W \subseteq V$  が与えられた時、 $W \subseteq V'$  かつ  $c(G')$  が最小であるような  $G$  の連結部分グラフ  $G'=(V', E')$  を求める。

$c(G')$  は  $G'$  の全てのエッジのコストの総和を示す。 $W \subseteq V'$  のような  $G$  の任意の連結部分グラフ  $G'$  が木であるとき  $G'$  は  $G$  での  $W$  に対するスタイナ木と呼ばれる。最小コストを持つ解  $G'$  は  $G$  での  $W$  に対する最小スタイナ木 (MStT) と呼ばれる。 $V'=W \cup S$  のような  $S \subseteq V \setminus W$  は  $G'$  のスタイナ節点と呼ばれる。 $n=|V|$ 、 $m=|W|$  とすると、 $2 < m < n$  において SPG は NP 完全である [7]。

#### 2.1.2 直交スタイナ木 (RST) 問題

一般に、平面上に与えられた複数個の点に対し、それらを接続し、かつ木を構成する枝が水平線分もしくは垂直線分のみから構成される場合、そのような木は直交スタイナ木 (Rectilinear Steiner tree) と呼ばれる。平面上に端点の集合  $T$  が与えられた時、枝の長さの総和が最小となる直交スタイナ木を見つける問題を直交スタイナ木問題という。このとき、直交スタイナ木上で端点でない点はスタイナ節点と呼ばれる。

#### 2.1.3 VLSI 配線問題

VLSI 概略配線経路は 2.1.2 の直交スタイナ木とみなすことができる。そこで、各端点にソース、シンクを割り当て、ソー

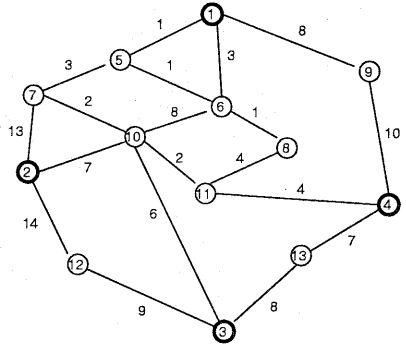


図 1 SPG の例

Fig. 1 An example instance of the SPG

ス、シンク間の最大遅延時間を最小化した上で、総配線長を最小化することを目的とする。ここで、配線遅延の見積りとして Elmore の提案した配線遅延モデル [10] を用いて計算する。

### 2.2 適応的 GA

遺伝的アルゴリズムにおいてパラメータ設定が難しいということが知られており、有効なパラメータ設定を行うための多くの研究が行われている [13], [14]。本稿では文献 [9] で提案されている適応的 GA を用いる。ここでは、交差確率等のパラメータ設定については頑健なものを使用し、交差等のオペレータを複数個用意しておき、各オペレータのパフォーマンスに応じて適用確率を生成し、その適用確率に応じて複数のオペレータの中からオペレータを撰択する。これにより、GA 実行中はもちろん問題に応じて最適なオペレータを適応的に用いることができる。

## 3. 提案フレームワーク

本稿では、種々の配線問題を統一的に解くことができるフレームワークを提案する。

### 3.1 問題の定式化

提案フレームワークでは、以下に示すグラフに各問題で必要となる情報をラベルとして節点とエッジに与え、各問題に対するマッピングを定義することで、目的関数が異なる問題を解くことが可能になる。また、各問題における目的関数については各問題においてユーザが定義できるものとする。以下では提案フレームワークにおける問題の定式化を行う。

入力:  $G=(V, E)$ , ( $V$ : 節点集合,  $E$ : エッジ集合)

$V$  の部分集合  $W \subseteq V$

エッジラベル  $e_i (i=1 \sim k)$

節点ラベル  $v_j (j=1 \sim m)$

出力: 目的を満たしたスタイナ木  $T=f(G(V,E), W, e_i, v_j)$

目的: ユーザが定義した目的関数  $f(T)$  の最小化

### 3.2 各問題へのマッピング

以下では、3.1 で示した一般的なスタイナ木問題の定式化を用いて、2.1 で示した各種のスタイナ木問題がどのように定式化できるかを示す。

グラフスタイナ木問題では、グラフのエッジラベル  $e_1$  にエッ

ジコストを与える。

入力:  $G = (V, E)$ , ( $V$ : 節点集合,  $E$ : エッジ集合)

$V$  の部分集合  $W \subseteq V$

エッジラベル  $e_1$

出力: スタイナ木  $T = f(G(V, E), W, e_1)$

目的: 総配線長の最小化

直交スタイナ木問題では, Hanan [11] の結果より, 端点を通過する水平線分と垂直線分が交差するスタイナ節点だけを用いる最適な直交スタイナ木 (RST) が存在することが示されている。つまり, 与えられた端点集合に対して最小 RST を求めることは節点  $u, v$  間の直交距離  $d(u, v) = |x_u - x_v| + |y_u - y_v|$  で与えられるエッジコストを持つ Hanan グリッド上で最小スタイナ木を見つけることに帰着できる。ただし,  $(x_u, y_u), (x_v, y_v)$  はそれぞれ節点  $u, v$  の  $x, y$  座標である。そこで, 与えられた端点集合からハナングリッドによって求められるグラフと各節点間の直交距離をそれぞれ  $G = (V, E)$ , エッジラベル  $e_1$  として与える。

入力:  $G = (V, E)$ , ( $V$ : 節点集合,  $E$ : エッジ集合)

$V$  の部分集合  $W \subseteq V$

エッジラベル  $e_1$

出力: 目的を満たしたスタイナ木  $T = f(G(V, E), W, e_1)$

目的: 総配線長の最小化

VLSI 配線問題では, 直交スタイナ木問題同様に与えられた端点集合からハナングリッドによって求められるグラフと直交距離を表す配線長をそれぞれ  $G = (V, E)$ , エッジラベル  $e_1$  として与え, さらに節点ラベル  $e_2$  にソース, シンク情報を与える。また, 配線遅延を見積もるための Elmore 遅延の計算で必要となる配線幅をエッジラベル  $e_3$  として与える。

入力:  $G = (V, E)$ , ( $V$ : 節点集合,  $E$ : エッジ集合)

$V$  の部分集合  $W \subseteq V$

エッジラベル  $e_i (i=1 \sim 3)$

節点ラベル  $v_1$

出力: 目的を満たしたスタイナ木  $T = f(G(V, E), W, e_1, e_2, e_3, v_1)$

目的: (1) 総配線長の最小化

(2) 最大遅延時間の最小化

ここで, VLSI 配線問題における目的は最大遅延時間を最小化した上で総配線長を最小化する。

## 4. 提案アルゴリズム

### 4.1 アルゴリズムの概要

本稿で提案する手法は遺伝的アルゴリズム (Genetic Algorithm, GA) に基づいている。GA は複雑な最適化問題に対する頑健なヒューリスティック探索手法として知られており [3], VLSI 設計問題に対しても GA に基づく多くの手法が提案されている [4], [5]。

提案 GA は世代を持つ GA であり,  $m$  個の個体を個体集合 (人口)  $P_C$  として保持し, それぞれの個体はスタイナ木を表現している。アルゴリズムの実行が開始されると個体集合  $P_C$  がランダムに生成され, 各個体の適応度が評価される。現世代の個体集合  $P_C$  の個体には交差確率  $p_c$  で交差, および突然変異確

```

initialize( $V_c''(p_{c1}, p_{c2}), V_m''(p_{m1}, p_{m2}, p_{m3}, p_{m4})$ );
 $P_C = \text{generate}(m)$ ;
evaluate( $P_C$ );
repeat  $G_{max}$  times;
   $P_N = \emptyset$ ;
  repeat  $(m \times g) / 2$  times;
     $i_a = \text{select2}(P_C)$ ;
     $i_b = \text{select2}(P_C)$ ;
     $P_N = P_N \cup \text{mutate}(\text{crossover}(i_a, i_b, P_C, V_c''), P_m, V_m'')$ ;
     $P_S = \text{record}(i_a, i_b, c1, c2, m1, m2, m3)$ ;
  end;
  evaluate( $P_C, P_N$ );
   $P_C = \text{select}(P_C, P_N, m)$ ;
  operator_fit_cal( $P_N$ );
  adaptive_cal( $M_c, M_m$ );
end;

```

図 2 提案アルゴリズムの概要

Fig. 2 Overview of the algorithm

率  $p_m$  で突然変異が適用され, 次世代の個体候補の集合  $P_N$  が形成される。ここで, 交差手法, 突然変異手法はそれぞれあらかじめ複数用意されており, それぞれの手法の適用確率のリスト ( $V_c'', V_m''$ ) が与えられ, これに基づいて適応的に選択される。世代間ギャップを  $g (0 \leq g \leq 1.0)$  とすると  $|P_N| = g \times m$  である。最後に現世代の個体集合と次世代の個体候補の集合の各個体の適応度を評価し, 優良解  $m$  個を選択し, 次世代の個体集合が形成される。アルゴリズムは以上の手続きを世代数があらかじめ与えられた数  $G_{max}$  に達するまで繰り返す。

アルゴリズムの概要を図 2 に示す。図 2 において, 関数  $\text{evaluate}(P_C)$  は個体集合  $P_C$  の要素である各個体の適応度を評価する関数, 関数  $\text{select2}(P_C)$  は個体集合  $P_C$  からランダムに異なる 2 個体を選択し, 適応度の優れた 1 個体を返す関数, 関数  $\text{crossover}(i_a, i_b, p_c, V_c'')$  は個体  $i_a$  と個体  $i_b$  を交差確率  $p_c$  で複数の手法の中から適用確率リスト  $V_c''$  に基づいて適応的に手法を選択して交差し, 得られた 2 個体を返す関数, 関数  $\text{mutate}(i_a, i_b, p_m, V_m'')$  は個体  $i_a$  と個体  $i_b$  に対してそれぞれ突然変異確率  $p_m$  で複数の手法の中から適用確率リスト  $V_m''$  に基づいて適応的に手法を選択して突然変異を行い, 得られた 2 個体を返す関数, 関数  $\text{select}(P_C, m)$  は個体集合  $P_C$  の中から適応度の優れた  $m$  個の個体を返す関数である。関数  $\text{operator\_fit\_cal}(P_N)$  は交差, 突然変異が施された次世代に残る個体に対して各オペレータの評価値を計算する関数である。関数  $\text{adaptive\_cal}(M_c, M_m)$  では各オペレータの評価値を基に各オペレータの適用確率リスト  $V_c'', V_m''$  を計算する関数である。それぞれの関数の動作の詳細については以下の節で説明する。

### 4.2 個体表現

一般に GA においては, 問題の解をどのように個体としてコーディングするかが重要である。一般にスタイナ木を表現するためにはトポロジ情報と幾何学的な情報の 2 つが必要である。前者は節点の親子関係を表し, 後者はスタイナ木の配線セグメ

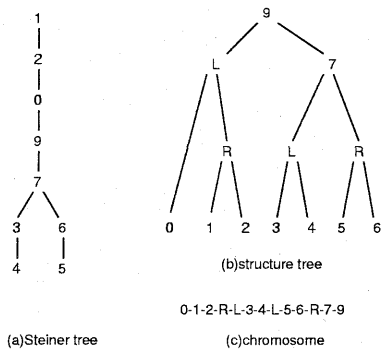


図3 個体表現  
Fig.3 Genotype

ントをどのようにレイアウトするかを表す。本稿では、個体のトポロジ情報をコーディングし、幾何学的な情報はその後の適応度評価計算中に決定される。また、スタイナ節点をトポロジ情報に含めることにより、個体表現からより詳しい配線経路を求めることを可能にしている。

スタイナ木のトポロジ情報は以下のように再帰的にコード化される。まず、 $T$ をスタイナ木とし、 $T_u$ を $T$ の部分木とする。ここで、 $T_u$ は部分木 $T_v$ 、 $T_w$ から構成されているとし、それぞれ少なくとも1つの葉節点を含んでいるとする。この時、 $code(T_v)$ 、 $code(T_w)$ をそれぞれ $T_v$ 、 $T_w$ のコードとする。そのとき、内部節点を+で表すとすると、 $code(T_u) = code(T_v)code(T_w)+$ と定義する。+はL、R、節点のいずれかであり、このコードから配線の木構造が得ることができ、これよりスタイナ木の幾何学情報を生成することができる。ここで、L、Rはその内部節点が部分木での左下、右下に対応する節点であることを意味し、+が節点の場合はその節点がスタイナ節点であることを示す。これは、スタイナ木 $T$ に対し、適当な内節点を根とした場合の $T$ 内の枝で接続された節点同士の親子関係を逆ポーランド記法で表現した場合に対応している。逆ポーランド記法で表された部分木同士の関係は2分木で表現できる。スタイナ木のトポロジ情報を逆ポーランド記法で表現し、さらに2分木で表現したものを以下では構造木と呼ぶ。図3にスタイナ木の例と、それに対応する構造木、およびポーランド記法を示す。

#### 4.3 適応度評価

遺伝的アルゴリズムにおいては個体を評価するために、適応度関数が必要である。提案アルゴリズムにおいて個体は配線経路のトポロジ情報のみを指定するので、個体を評価するためにはトポロジ情報に基づいてスタイナ木として実際に経路を決定する必要がある。??で示したように、提案アルゴリズムにおいて個体はスタイナ木のトポロジ情報を構造木として保持している。以下では個体が表現する構造木 $S$ に対し、 $S$ に対応するスタイナ木 $T$ を求める手順を示す。

ここで、構造木 $S$ で示された接続関係(親子関係)をできる限り保持したいので、構造木 $S$ の各部分木内での2節点間の最短経路を順次求めていき、最終的に全体の配線経路を決定する。

まず、構造木において内部節点を根とする部分木ごとに経路を決定する。部分木内での配線経路の決定はまず左子から親への最短経路を決定し、次に右子から親への最短経路を決定する。そこで、重複経路があれば、その経路は既に通過した経路として配線経路には加えない。これらを個体表現における内部節点の出現順に行う。また、親がスタイナ節点でない場合、つまりL、Rである場合は子同士で経路を決定し、L、Rに当たる部分をそれぞれ左子、右子に置き換える処理を行う。各2節点間の最短経路はあらかじめFloydのアルゴリズム[6]を用いて求める。

#### 4.4 交差

次世代の個体集合を生成するために、個体集合に交差を適用して次世代の個体の候補集合を生成する。交差の適用対象となる個体はトーナメントサイズが2のトーナメント選択により選択した2個体に対し、交差確率 $p_c$ で交差を適用する。提案アルゴリズムで適用する交差手法は2種類用意し、適用確率によって選択される。また、これらの交差手法を部分木交差という。まず、大きさが同じ部分木を交換する部分木交差について説明する。以下では交差の対象となる個体を構成木とみなし、 $S_1$ 、 $S_2$ で表す。まず、構成木 $S_1$ の中からランダムに内部節点を選択する。ただし、根の内部節点は選択の対象から除外する。次に構成木 $S_2$ の中からランダムに内部節点を選択し、選択した内部節点を根とする部分木が構成木 $S_1$ の部分木と大きさが同じであるものを選択する。選択したそれぞれの内部節点を $v$ 、 $w$ とする。次に $v$ 、 $w$ を根とする部分木を交換する。部分木を交換するとそれぞれの木において葉節点のラベルとスタイナ節点に当たる内部節点が重複する。これを修復するために葉節点に関しては重複を除去し、それによって不足した葉節点に加え、スタイナ節点に当たる内部節点に関しては重複しない他のスタイナ節点を新たに加える。

交差がどのように行われるかを図4に示す例を用いて説明する。構成木 $S_1$ 、 $S_2$ をそれぞれ親1、親2とし、破線で囲まれた部分木を交換するものとする。部分木を交換すると子1に関して葉節点で4、6が内部節点で9が重複する。そこで、部分木内には重複した葉節点4、6を0、2に変更する。また、内部節点において9が重複するので、重複が起らない他のスタイナ節点をランダムに選択する。子2においても同様の操作を行う。

同様の方法で大きさの異なる部分木同士を交換を許す部分木交差も用意する。

#### 4.5 突然変異

交差適用後の個体に対し、突然変異が突然変異確率 $p_m$ で適用される。突然変異手法は4種類用意し適用確率によって選択される。それらは葉節点交換、スタイナ節点変更、L-R変更、部分木交換である。以下では、部分木交換手法について説明する。まず、構成木 $S$ の相異なる2節点 $u$ 、 $v$ を選択する。 $S_u$ 、 $S_v$ を $u$ 、 $v$ を根とする $S$ の部分木とする。もし $S_u$ 、 $S_v$ が同じ葉節点を1つも含まなければ、 $S_u$ 、 $S_v$ を交換し、突然変異は終了する。もし、同じ葉節点を含む場合はもう一度始めから突然変異をやり直す。

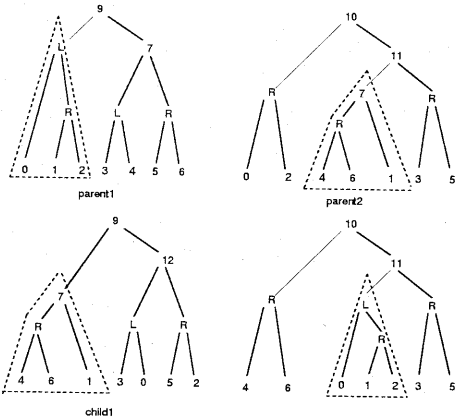


図4 交差  
Fig.4 crossover

#### 4.6 選択

選択は個体集合の中から優良個体を選択し、次世代に残す操作である。提案アルゴリズムでは選択を2カ所で行っている。まず、交差の対象となる2個体を選択する場合はGAの選択手法として一般的であるトーナメントサイズが2のトーナメント選択を行う。すなわち、ランダムに選択された個体を  $i_1, i_2$  とするとき、 $i_1$  と  $i_2$  の内の優れた個体を交差の対象とする。この操作を2回繰り返して、交差の対象となる2個体を選択する。これにより優れた個体ほど、交差を通して次世代に個体の優れた性質を残しやすくなる。

一方、次世代の個体集合を生成する場合は、現世代の個体集合  $P_G$  と、交差、突然変異により生成された次世代の個体集合の候補集合  $P_N$  の和集合を作成し、優良なものから個体集合のサイズ ( $m$ ) 分の個体を選択し、そのようにして生成された個体集合を改めて次世代の個体集合  $P_C$  とする。

#### 4.7 適応的GA

関数  $operator\_fit\_cal(P_N)$  では、オペレータのパフォーマンスを計算する。まず、オペレータにより子が生成された時は常にその親と子を生成したオペレータを記録する。次にもし現在の人口における最良解より良い解をもつ子を生成した時はその子を生成したオペレータにパフォーマンス評価値  $100 \times (\text{子の評価値}/\text{最良評価値})$  を与える。また、有効な個体を生成するための準備を行ったかもしれないオペレータを評価するためにパフォーマンス評価値の一部を親を生成したオペレータに与える。最後に  $M$  世代間でのパフォーマンスを計算するために、 $M$  世代間でオペレータが生成した各子の評価値を合計し、子の数で割ることにより、各オペレータのパフォーマンスを計算する。

関数  $adaptive\_cal(M_c, M_m)$  では、上記で計算されたオペレータのパフォーマンス評価値を基に各オペレータの適用確率を計算する。もし、全てのオペレータのパフォーマンス評価値が0であるなら、適用確率はそのままにしておく。そうでなければ、適用確率のリスト  $V$  にある定数  $((100-S)/100)$  を掛けることによって合計を  $100-S$  になるようにする。これを基本適用確率

$V'$  とする。つぎに  $M$  世代間でのオペレータのパフォーマンス評価値のリスト  $D$  を生成する。このリスト  $D$  にある定数を掛けることによって合計が  $S$  になるようにする。これをパフォーマンス評価値のリスト  $D'$  とする。 $V', D'$  から新しい適用確率  $V'' = V' + D'$  とする。

以下に簡単な例を用いて適用確率の計算方法を示す。以下では4つのオペレータがあるとし、まず、各オペレータの現在の適用確率(%で表現)を  $V=(10,30,20,40)$ 、パフォーマンス評価値を  $D=(25,12,33,9)$  と仮定する。さらに、 $S=15$  と仮定すると、現在の適用確率は15%変動する。これより、 $V'=0.85 \times V = (8.5, 25.5, 17, 34)$  となる。 $D'$  は  $D$  が合計15となるようにし、 $D'=(4.74, 2.28, 6.27, 1.71)$  となる。これらより、 $V'' = V' + D' = (13.24, 27.78, 23.27, 35.71)$  となる。以上より、 $V$  と  $V''$  を比較すると、1番目と3番目のオペレータの適用確率が増加し、2番目と4番目のオペレータの適用確率は減少していることが分かる。

本稿では、交差を2つ、突然変異を4つ用意し、交差、突然変異それぞれに対して適用確率を用いて適応的に選択するようにする。こうすることで交差、突然変異それぞれに適用確率を更新する世代数を設定することができ、それぞれの特性を活かすことができると考えられる。

### 5. 実験的評価

提案手法をC言語でプログラムとして実現し、ワークステーション上で実行することにより実験的評価を行った。提案手法の有効性を示すために、2種類の従来手法と比較した。従来手法1は個体表現がスタイン節点の有無を表し、一点交差、突然変異、ルーレット選択を行う遺伝的アルゴリズムであり、従来手法2はダイクストラ最短経路手法を基にしたスタイン木ヒューリスティック [15] である。従来手法1においては、スタイン節点と  $W$  で指定される端子節点で構成される部分グラフの最小木を求めることでスタイン木を求めている。また、本実験においては提案アルゴリズムの実現において、適応的オペレータの選択はまだ実現しておらず、交差手法として同じ部分木を交換する部分木交差を、突然変異として部分木突然変異を用いている。提案アルゴリズムと従来手法は Beasley [12] の OR-Library から利用できるグラフスタイン問題から3つの問題に対して実験を行った。これらの問題における問題サイズを表1に示す。表1において1行目にグラフ問題名を表し、次の3行では入力グラフでの節点数  $|V|$ 、端点数  $|W|$ 、エッジ数  $|E|$  を示し、最後の行で最適解を示す。

遺伝的アルゴリズムのパラメータとしては以下のパラメータを用いた。個体集団サイズ40、交差確率0.6、突然変異確率

表1 問題のサイズと最適解

Graph	Problem size			Optimal value
	$ V $	$ W $	$ E $	
B-5	50	13	100	61
B-10	75	13	150	86
B-18	100	50	200	218

表2 実験結果

Graph	提案アルゴリズム				従来手法 1				従来手法 2	
	best	worst	ave	CPU[sec]	best	worst	ave	CPU[sec]	best	CPU[sec]
B-5	61	87	74.4	2.11	61	71	63.10	2.58	75	0.01
B-10	86	105	92.2	3.20	86	105	93.50	4.43	117	0.04
B-18	225	256	242.7	13.52	218	223	220.20	16.90	307	0.03

0.01, 世代間ギャップ 0.6, 最大世代数 200. 各テストデータに対して提案アルゴリズム, 従来手法 1 を用いて 10 回, 従来手法 2 を用いて 1 回実行する. 表 2 に各テストデータに対する提案アルゴリズム, 従来手法を用いた実験結果を示す. 表 2 において best は最良解, worst は最良解の中での最悪解, ave は最良解の平均値, CPU は計算時間を表す.

表 2 より, 提案アルゴリズムと従来手法 1 は B-5, B-10 に対して共に最適解を求めることができていない. B-18 に対して提案アルゴリズムは最適解を求めることができていない. 従来手法 2 ではダイクストラ最短経路手法を用いて逐次的に求めているために解が悪くなっている. また, worst では問題によって提案アルゴリズムの方が従来手法 1 より悪くなっている. これらの原因として, 交差手法として同じ大きさの部分木を交換する交差を用いているために, うまく解の更新が行われていないからだと考えられる. 計算時間に関しては逐次探索である従来手法 2 の計算時間が一番短く, 同じ遺伝的アルゴリズムである従来手法 1 と比べて提案アルゴリズムの計算時間のほうが短い. 以上より, 提案アルゴリズムは従来手法と比べて同等の結果を得ることができていると考えられる.

## 6. あとがき

本稿では, VLSI レイアウト設計における概略配線に対し, 遺伝的アルゴリズム (GA) を用いてグラフ上でスタイナ木問題を解くためのフレームワークを提案した. 実験結果より, グラフスタイナ木問題に対して従来手法と比較して同等の結果を得ることができた.

今後の課題としては, 提案アルゴリズムに適応的 GA を導入し, 同じ大きさの部分木交換を行う交差手法以外に異なる大きさの部分木交換を行う交差手法を適用可能にする. これにより, 最適解をより短い時間で求めることが可能になると考えられる. また, グラフスタイナ木問題以外の直交スタイナ木問題や VLSI 配線問題に対しても実験を行い, 提案フレームワークの有効性を示すことも今後の課題である.

## 謝辞

従来手法のプログラム開発に御協力いただいた本学学部生の河内智彦君に感謝します.

## 文 献

- [1] M.Sarrafzadeh, and C.K. Wong, "An Introduction to VLSI Physical Design", McGraw-Hill, New York, 1996.
- [2] J.Cong, "An interconnect-centric design flow for nanometer technologies" Proc. IEEE, vol.89, no.4, pp.505-528, April 2001.
- [3] D.E. Goldberg, "Genetic Algorithm in Search, Optimization, and Machine Learnig" Addison-Wesley, Reading, 1989.
- [4] J. Lienig, and J.P. Cohoon, "Genetic Algorithms applied to the physical design of VLSI circuits: A survey" Proc.Parallen Problem Solving from Nature IV, pp839-848, Berlin, Germany, Sept. 1990.
- [5] P. Mazumber, and E. M. Rudnick, "Genetic Algorithms for VLSI Design, layout & Test Automation" Prentice-Hall PTR, Upper Saddle River, USA, 1999.
- [6] A. V. Aho, and J. E. Hopcroft and J. D. Ullman, "Data Structures and Algorithms" Addison-Wesley, Reading, Mass, 1983.
- [7] R. M. Karp, "Reducibility among Combinatorial Problems" In R. E. Miller, J. W. Thatcher(Eds.), Complexity of Computer Computations, Plenum Press, New York, pp.85-103, 1972.
- [8] A. Kapsalis, and V. J. Rayward-Smith, and G. D. Smith, "Solving the Graphical Steiner Tree Problem Using Genetic Algorithms" Journal of the Operational Reserch Society, vol.44, No.4, pp.397-406, 1993.
- [9] Davis and Lawrence, "Adapting operator probabilities in genetic algorithms" Proceeding of the Third International Conference on Genetic Algorithms, San Mateo, Calif: Morgan Kaufmann Publishers, 1989.
- [10] W. C. Elmore, "The transient response of damped linear network with particular regard to wideband amplifier" J. Applied Physics, 19, pp.55-63, 1948.
- [11] M. Hanan, "On Steiner's problem with rectilinear distance" SIAM J. appl. Math, vol. 14, pp.255-265, 1966.
- [12] J. E. Beasley, "OR-Library: Distributing test problems by electronic mail" J Oper Res Soc41, pp.1069-1072, 1990.
- [13] DeJong, and A. Kenneth, "An Analysis of the Behavior of a Class of Genetic Adaptive Systems" University of Michigan, Ph.D. Dissertation, 1975.
- [14] Grefenstette, and J. John, "Optimization of control parameters for genetic algorithms" IEEE Transactions on Systems, Man, and Cybernetics, SMC-16(1), 1986.
- [15] Sadiq M. Sait, and Habib Youssef, "VLSI Physical design automation", IEEE Press, pp.259-261.