

最小全域木を生成した遺伝的アルゴリズムによる 直線スタイナー問題

吉村 昌浩 渋谷 進

茨城大学工学部情報工学科

VLSI回路において配線のレイアウトを行う際に効率的になるように求める問題が考えられる。そこで、概略配線に使用されやすい直線スタイナー木を用いて、その効率的なレイアウトを求めた。しかし、この問題はNP-完全であるので最適解を得るのが困難である。そこで、本研究では遺伝的アルゴリズム(GA)を適用してこの解の近似値を求めた。また、今までの多くの研究はその評価基準に長さを用いているが、本研究では通信遅延で評価し、エルモア遅延と呼ばれるものを用いた。更に、初期集団を生成する際にその一部として最小全域木を含ませることによって、最小全域木のない場合と比較して、最小値を最大で45%、平均値を最大で52%に削減することができた。その際、ノード数が多ければ多いほど解の改善率が大きくなる結果を導いた。

A rectilinear Steiner problem using genetic algorithms by the generation of minimum spanning trees

Masahiro Yoshimura Susumu Shibusawa

Department of Computer and Information Sciences, Ibaraki University

Many efficient methods of wire layout problems have been studied for LSI circuits. This paper presents an efficient method of wire layout for rectilinear Steiner trees which are useful for loose routing. Since this problem is NP-complete and very hard to find the optimum solution, we obtained an approximate method using genetic algorithms. Many studies have been done using the total length of generated tree edges, but our method uses maximum communication delay as an evaluation criterion, which is a physical one and called 'Elmore delay'. Applying a minimum spanning tree to the initial population, the minimum value of solution was reduced to at most 45% of the solution without a minimum spanning tree, and the average value was reduced to at most 52%. The result shows this method becomes more effective for the rectilinear Steiner tree with large number of nodes.

1 はじめに

近年、電子機器の多様化に伴い、VLSI回路において配線のレイアウトを行う際に効率的になるように求める問題が多く考えられている。そこで、大まかな配線状況を知るために使用されることが多い直線スタイナー木を用いることによって、その効率的なレイアウトを求める。しかし、この問題はNP-完全であるということからその最適解を得るのが非常に困難であるとされている。そこで、本研究では確率的な探索の一手法である遺伝的アルゴリズム (GA)[1],[2] を適用することによって、この最適解の近似値を求めていく [3]。

また、今までの多くの研究では、その評価基準に生成された木の長さの総合計を用いているが、本研究では生成された木における通信遅延の最大値で評価している。これは、半導体加工技術の微細化に伴い、素子遅延に対する相対的な配線遅延の割合が高くなり、設計段階において動作速度や消費電力などといった性能面での要求がますます増加し、配線遅延をできるだけ正確に見積もる方法も必要とされてきているからである。しかし、最近の研究により木の辺の長さの総合計が最小であるからといって必ずしもその遅延での最大値が最小になるとは限らないとされている。そこで、物理的な遅延であるエルモア遅延 [4] と呼ばれるものを用いて評価を行なっている。

更に、シミュレーションを行う際、初期集団を生成するときとその中の一つとして最小全域木を当てはめることによって更なる解の改善を行うことを試みた。

2 直線スタイナー木

ユークリッド平面において n 個のノードを連結する木を見つけることを考える。与えられたノードのみでの連結を考えると、これが無向グラフであるならば全域木となることはよく知られている。このとき、全域木は $n-1$ 個の辺を含んでいる。図 1 はある 10 個のノードの集合上での全域木を示す。このように連結された木の中で最も木の辺の長さの総合計が短くなる木のことを最小全域木と呼ぶ。これを発見する手法はいくつか提案されており、簡単に求めることができる。

また、この全域木の辺の長さの総合計を更に短くするために追加のノードを与えるとき、この

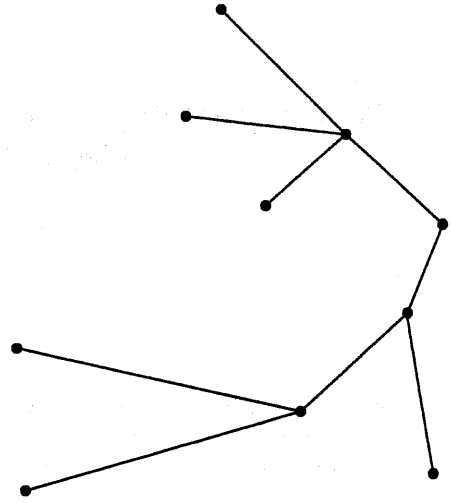


図 1: ある 10 点による全域木

ノードをスタイナー点と呼び、生成される木がスタイナー木である。この木におけるスタイナー点はすべて次数が 3 で、まわりにできる角度はすべて 120° となり、多くとも $n-2$ 個であるという特徴を持っている。図 2 は図 1 の全域木のノードと同じ 10 個のノード上でのスタイナー木である。

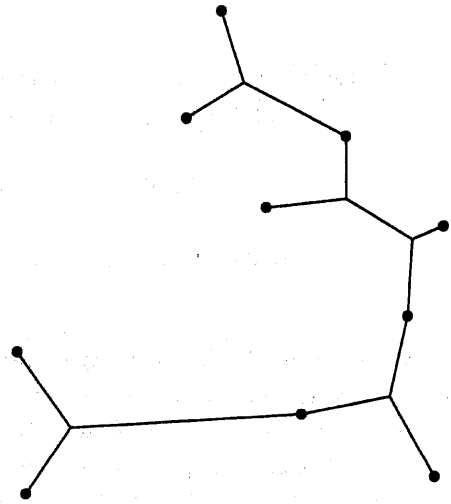


図 2: ある 10 点によるスタイナー木

そして、全域木の辺が垂直方向と水平方向のみで結ばれている木のことを直線スタイナー木と呼ぶ。ただし、直線スタイナー木のスタイナー点は、一般に $n-1$ 個存在する。つまり、全域木に

おけるすべての辺 $n - 1$ 本を結ぶ際に水平方向と垂直方向の変わり目の位置がスタイナー点となる。図3は図1,2の全域木、スタイナー木と同じ10個のノード上での直線スタイナー木である。

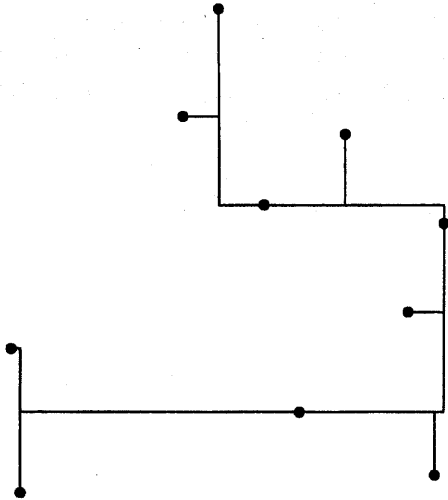


図3: ある10点による直線スタイナー木

与えられたノードの集合上での直線スタイナー木の辺の長さの総合計が最も短くなる木が最小直線スタイナー木であり、最小直線スタイナー木を見つけることが直線スタイナー問題である。ここで、 n 個のノードからなる全域木の本数は Cayley の公式 [5] から n^{n-2} であるとされ、直線スタイナー木は全域木の辺1本1本に対して水平方向から垂直方向へ辺を結ぶ方法と垂直方向から水平方向へ辺を結ぶ方法の2通りの結び方があるので、直線スタイナー木の総計数は $2^{n-1}n^{n-2}$ 通りあることがわかる。

3 遅延

ここで述べている遅延には、計算が簡単であり、かつ精度が高いとして有効とされているエルモア遅延を使用している。それは木のコストと半径の両方の観点から考慮した抵抗 (resistance) と静電容量 (capacitance) による RC 遅延の近似である。ここで、木のコストとは生成された木の辺の長さの総合計のことであり、半径とは生成した木のノードにおいてすべての2点間の経路 (パス) の長さの総合計が最も長くなる時のものをいう。つまりコスト以外にこの半径を小さくする

ことによって回路中の任意の端子対とその時の部分回路の最大遅延が小さくなる。

図4のように出発点を n_0 とする木 $T(N)$ を与えて、あるノード v から n_0 方向への辺を e_v と表す。その辺における抵抗と静電容量をそれぞれ r_{e_v}, c_{e_v} と表す。また v から n_i 方向にできた $T(N)$ の部分木を T_v と表し、 v での負荷容量を c_v と表す。ただし、 v がスタイナー点であるならば、 $c_v = 0$ とする。また、部分木 T_v での静電容量、即ち部分木 T_v における負荷と辺の静電容量の総計を表すのに C_v を使用している。以上から、ここでは遅延として次式を用いている。

$$t(n_i) = r_d C_{n_0} + \sum_{e_v \in \text{path}(n_0, n_i)} r_{e_v} \left(\frac{c_{e_v}}{2} + C_v \right)$$

上式において r_d を出力抵抗とすると、 $t(n_i)$ は、出発点 n_0 から点 n_i への遅延を表している。そして、上式から出発点 n_0 を定めたときにそれぞれの点 n_i までの遅延を導き出す。そこで求めた値の中で最大となったものを最大遅延という。

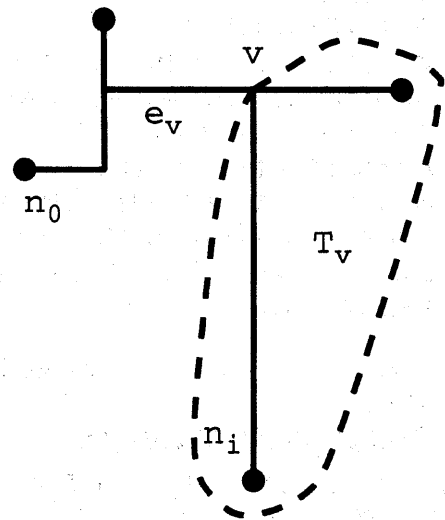


図4: 出発点 n_0 の直線スタイナー木

ここで、現在は回路における出発点を定め、その点からの最大となる遅延を求め、それを最小にしようとしているが、本研究では、出発点を任意に定め、その中で最大の遅延を最小にすることを目的としている。これにより、評価基準を従来のような長さではなく、遅延で行なうことができると考える。

4 遺伝的アルゴリズムとは

自然界における生物の進化過程において、ある世代を形成している個体の集合、すなわち個体群の中で、環境への適応度の高い個体が高い確率で生き残れるように再生していく。更に、交叉や突然変異によって、次の世代の個体群が形成し、最終的にその問題の最適解の近似値を求めていく手法であり、その流れは以下ようになる。

1. 初期化：決められた個体数の染色体を初期集団としてランダムに生成し、初期世代の個体群として設定する。
2. スケーリング：ある個体の適応度に対して、そのまま選択時に反映させるよりも効果的な選択を行うように関数を用いる。
3. 選択：集団中の適応度の高い個体に、より多く交配機会を与えることを目的として交配を行う親染色体を選ぶ操作である。
4. 交叉：選択によって選ばれた2つの親染色体の遺伝物質を組み合わせる新しい子染色体を次世代に残す操作である。
5. 突然変異：局所解からの脱出をはかり探索空間を広げるために確率的に選ばれた遺伝子を変化させる操作である。
6. 終了判定：終了条件を満たせば、その時に得られている最良の個体を問題における最適解とする。満たさなければ、ステップ2に戻る。この終了条件としては、世代数等があげられる。

ここで、ステップ2からステップ6までを遺伝的操作と呼ぶ。

5 遺伝的アルゴリズムの適用

本節では、遺伝的アルゴリズムを直線スタイナー問題に適用する方法について述べる。

5.1 準備

与えられたノードには、水平方向と垂直方向の座標データを与える。

ここで述べている遺伝的アルゴリズムは、交叉や突然変異については、ごく一般的な方法を用いているが、コード化については全域木を生成するものとその全域木と組み合わせる直線スタイナー木を生成するものの2種類の染色体を新たに使用している。

5.2 コード化

直線スタイナー木を生成するためにはまず全域木を生成する。全域木が生成される手順は下に示した通りであり、図5のような順で行われる。この図における頂点の上の数は頂点番号で、下の数はその頂点の次数を表す。また、結ばれた辺の隣の矢印は辺が引かれる方向を表し、その隣の数は辺の引かれた順番を表す。

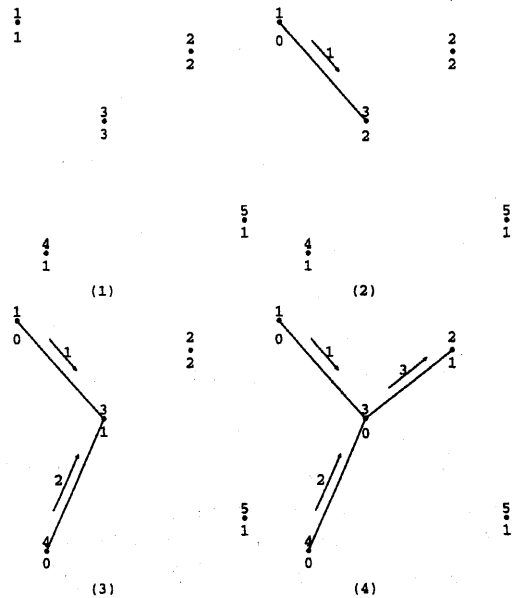


図5: 頂点列332の全域木の生成方法

1. 各頂点に番号をつけ、その次数を1とする。頂点の数が n のとき、頂点の番号を $n-2$ 個、任意に取り出し、その番号の頂点の次数を1ずつ増やし、取り出した順に並べる。以後、これを頂点列と呼ぶ。
2. 次数が1の頂点を番号順に見つけ、ステップ1での頂点列の頂点と順番に辺を結ぶ。そのときに次数1の頂点に辺で結ばれた頂点の水平方向と垂直方向のデータをもう一つ与える。辺が引かれた2つの頂点の次数を1ずつ減らす。
3. 2つの頂点のみ次数が1で、それ以外の頂点の次数が0になるまでステップ2を繰り返す。そして最後にこの2頂点間を辺で結ぶ。この頂点列と下に示した手順とを組み合わせることから直線スタイナー木を生成する。図6は、

その手順によって生成されていく直線スタイナー木の例である。この図における辺の隣の矢印は辺が引かれる方向を表し、その隣の数は矢印の方向に対応する方向列の数を表す。

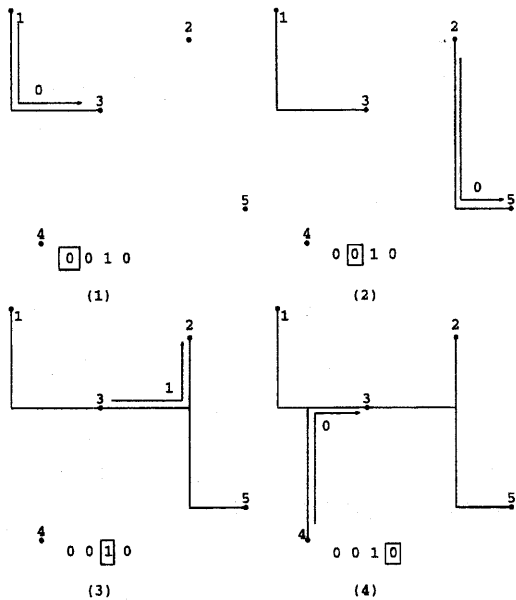


図 6: 頂点列 332 の全域木における方向列 0010 の直線スタイナー木の生成方法

1. 頂点の数が n のとき、任意に $n-1$ 個の 0 か 1 かのどちらかを選んで並べる。以後、これを方向列と呼ぶ。
2. ステップ 1 での方向列の順と頂点番号を対応させ、方向列の i 番目の数が 0 ならば頂点番号が i となる頂点からの辺の出る方向を垂直方向から水平方向へ結び、1 ならば水平方向から垂直方向へ結ぶ。
3. 最後の方向列による辺を結ぶまでステップ 2 を繰り返す。

5.3 遺伝的操作

5.3.1 選択

ここでは、その世代において最も良い解を持つ個体よりも次世代でそれ以上の解を持つ個体がでてこなければ、その個体を次世代に確実に残すエリート保存戦略とそれぞれの個体の適応度に比例してその個体が次世代に残る確率が高くなる適応度比例戦略を使用した。

5.3.2 交叉

ここでは、最も一般的な交叉である単純交叉を用いた。また、その例として図 7(a),(b) に示した。図 7(a),(b) では、4 番目と 5 番目の遺伝子座の間に交叉位置があり、親 P1 の染色体の先頭から 4 番目までと親 P2 の 5 番目から最後までまでの遺伝子が子 O1 に受け継がれ、親 P2 の染色体の先頭から 4 番目までと親 P1 の 5 番目から最後までまでの遺伝子が子 O2 に受け継がれる。

- 図 7(a) は頂点列の並びにおいて交叉を行うときに適用する。全域木で使う。
- 図 7(b) は方向列の並びにおいて適用する。直線スタイナー木で使う。

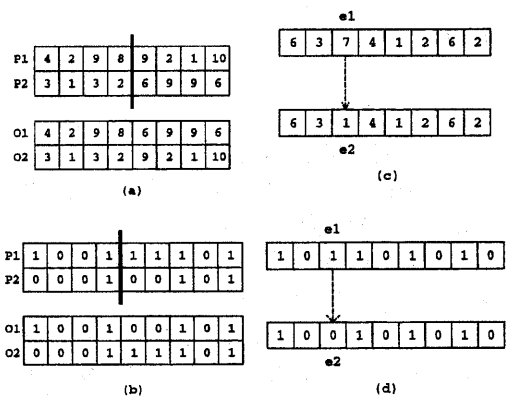


図 7: 単純交叉と突然変異

5.3.3 突然変異

突然変異は、個体の各遺伝子座に対して、一定の確率である突然変異確率によって適用した。また、その例として図 7(c),(d) に示した。図 7(c) では、頂点列の並びに対しては、その頂点番号 $e1$ を $e1$ 以外のランダムな頂点番号 $e2$ に変化させる。方向列の並びに対しては、 $e1$ が 0 ならば $e2$ を 1 に、 $e1$ が 1 ならば $e2$ を 0 に置き換える。

- 図 7(c) は頂点列の並びにおいて突然変異を行うときに適用する。全域木で使用する。
- 図 7(d) は方向列の並びにおいて適用する。直線スタイナー木で使用する。

5.3.4 評価関数

コード化において新たに2種類の染色体を用いたので、ここではそれぞれの染色体である頂点列と方向列の両方の評価関数を用意する。まず、方向列の評価関数は生成した直線スタイナー木におけるコストと遅延、半径の値の逆数でそれぞれ表す。これは、コストと遅延、半径の値が小さければ小さいほど良好な結果となるからである。頂点列の評価関数は、方向列との兼ね合いを持たせるためにそれぞれの頂点列を方向列と組み合わせていき、方向列の適応度の最も良い値となったものをその方向列と組み合わせる頂点列として与えるので、その頂点列が使われる割合で表す。また、ここで使う特徴量とは、直線スタイナー木におけるコストと遅延、半径のそれぞれの値のことを意味する。

$$f_i = \frac{l_i}{\text{個体数}} \quad g_j = \frac{10000}{m_j}$$

f_i : i 番目の頂点列の適応値

l_i : i 番目の頂点列が方向列に使われる回数

g_j : j 番目の方向列の適応値

m_j : j 番目の直線スタイナー木の特徴量

6 シミュレーション

シミュレーションを行なった状況としては、実験回数はそれぞれ10回ずつ行い、世代数は50世代まで、個体数についてはそれぞれノード数と同じ数で行った。また、GAの各パラメータである交叉確率は1.0、突然変異確率は0.01で行っている。また、以下の表の中で使っている長さとはコストのことを意味している。

そこでまず世代数に対する各特徴量の推移を図8に示す。ただしここでのノード数は10としている。ここでの比率とは、変化した世代における特徴量を第一世代で得た特徴量で割ったものに100をかけたものであり、式で表すと以下のようになる。

$$\text{特徴量の比率} = \frac{\text{変化した世代の特徴量}}{\text{第一世代の特徴量}} \times 100$$

またここで、直線スタイナー問題に対して遺伝的アルゴリズムで得た結果がどのようになったかを表1に示す。この表では、ある特徴量を評価基準とした場合でのそれぞれの特徴量を示してい

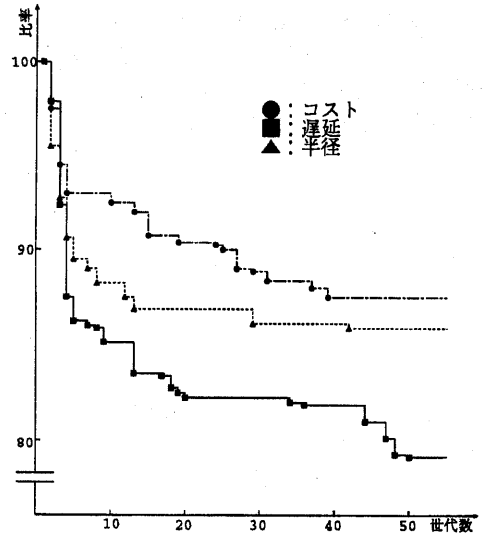


図8: 世代数に対する特徴量の推移

る。また、この表における対象とは、それを評価基準とした場合のことである。

表1: それぞれの評価に対する結果

対象	点	長さ		遅延		半径	
		平均	最小	平均	最小	平均	最小
長さ	5	14.4	13	109	89.5	12.4	11
遅延	5	14.2	13	102	85.0	11.7	10
半径	5	16.4	14	126	90.0	12.0	10
長さ	8	30.9	28	447	318	21.6	17
遅延	8	34.2	32	372	327	21.6	14
半径	8	36.4	29	463	349	18.9	17
長さ	10	63.7	59	1731	1227	43.3	37
遅延	10	64.7	57	1209	959	35.3	27
半径	10	79.2	72	1850	1293	34.9	30
長さ	12	86.3	80	2876	2455	56.5	57
遅延	12	92.0	88	2388	1961	49.2	45
半径	12	97.6	111	2498	2440	42.1	37

ここで2種類の染色体を生成した新たなコード化による手法が以前の頂点列と方向列を一つの染色体として表したコード化による手法と比べてどのようになるかをパーセンテージで表2に示す。ここでの比率とは、新たなコード化による手法の各特徴量を以前のコード化による手法の各特徴量で割ったものに100をかけたものであり、式で表すと以下のようになる。

$$\text{コード化の比率} = \frac{\text{新たなコード化}}{\text{以前のコード化}} \times 100$$

表 2: 新たなコード化の性能 (単位は%)

対象	点	長さ		遅延		半径	
		平均	最小	平均	最小	平均	最小
長さ	5	95	93	90	87	92	85
遅延	5	90	93	85	97	89	100
半径	5	94	88	87	75	98	91
長さ	8	98	104	96	103	85	81
遅延	8	99	97	83	79	94	64
半径	8	97	88	89	91	91	94
長さ	10	97	100	78	103	80	100
遅延	10	92	92	81	84	94	93
半径	10	107	101	101	99	101	107
長さ	12	90	101	89	93	91	102
遅延	12	93	116	86	99	92	96
半径	12	96	112	84	96	92	88

また、ここで新たに頂点列で初期集団の中の一つにおいて最小全域木を生成させることによって解の改善を試みたシミュレーションを行なった。表1での値に対してこの最小全域木を用いた方法がどのようになったかをパーセンテージで表3に示す。ここでの対比率とは、最小全域木を生成した場合を生成しなかった場合(表1の場合)で割ったものに100をかけたものであり、式で表すと以下のようになる。

$$\text{対比率} = \frac{\text{最小全域木を生成した場合}}{\text{生成しない場合}} \times 100$$

表 3: 対比結果 (単位は%)

対象	点	長さ		遅延		半径	
		平均	最小	平均	最小	平均	最小
長さ	5	94	100	88	100	90	100
遅延	5	96	100	93	105	96	110
半径	5	88	93	83	94	90	100
長さ	8	83	89	64	75	67	82
遅延	8	78	78	77	67	78	100
半径	8	73	86	70	63	78	82
長さ	10	70	75	70	63	73	84
遅延	10	72	79	71	76	92	115
半径	10	63	71	78	68	88	97
長さ	12	68	73	68	46	66	63
遅延	12	65	67	52	54	77	75
半径	12	64	53	77	45	79	89

表3の結果からノード数と対比率の推移が各特微量においてどのようになるかを図9,10,11に示す。

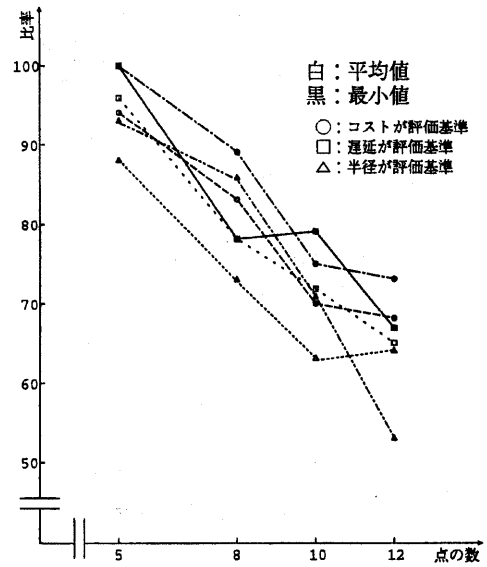


図 9: 長さの対比率の推移

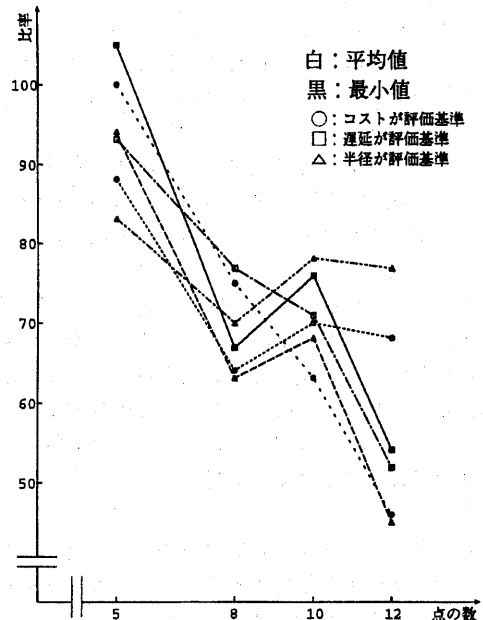


図 10: 遅延の対比率の推移

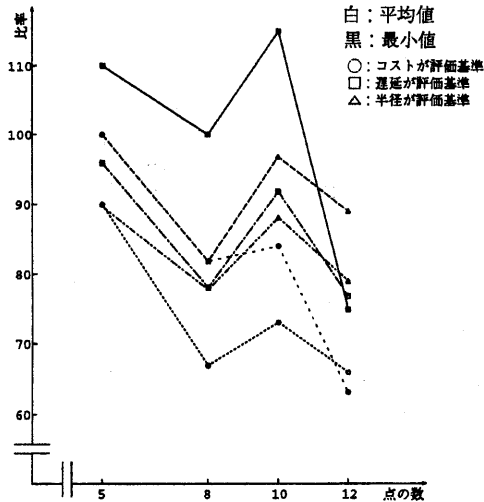


図 11: 半径の対比率の推移

7 考察

表1の実験の結果からノード数が少ない時には評価の対象がどれであっても同じような結果を得るが、ノード数が多くなると評価の対象がコストと半径はその値のみしか良い結果を得られないのに対し、遅延はすべての値に対して良い結果を得ることができている。ここでの遅延とはコストと半径の両方を含むのですべての値が良い結果を得ることができる。この結果から、現在は直線スタイナー問題の評価の対象は主にコストであるが、遅延を用いてもコストの量は同程度の結果が得られると同時に、更に半径を小さくできる。

また、表2から、新たなコード化によって、最小値は上回ったり、下回ったりしているが、平均値から全体的に安定して良い値を得られることがわかる。これは、評価するときに1種類の染色体だと1つで行っていたものを2種類の染色体では2つで行うことによって得られたものだと思うられる。

本報告では新たに頂点列の初期集団の一つとして最小全域木を生成することによって更なる各特徴量の値の改善ができた。これは、辺の連結においてあまりにも長い部分がないためだと思うられる。また、ノード数と改善率との関係からノード数が多くなればなるほど、本手法が有効に働く

ように思われる。今後は更にノード数を増やして、計算時間の短縮を行ったり、他の研究との比較などを行ない、更に効率良い方法を考えなければならない。

謝辞

ご討論いただいた茨城大学工学部渋沢研究室の皆様へ深く感謝いたします。

参考文献

- [1] 北野 宏明: 遺伝的アルゴリズム, 産業図書, 1993.
- [2] 坂和 正敏, 田中 雅博: 遺伝的アルゴリズム, 朝倉書店, September 1995.
- [3] Bryant A. Julstom: A Genetic Algorithm for the Rectilinear Steiner Problem, Proceedings of the Fifth International Conference on Genetic Algorithms, pp.474-480, July 1993.
- [4] K. D. Boese, et al.: Rectilinear Steiner Trees with Minimum Elmore Delay, Proceedings of the 31st ACM/IEEE Design Automation Conf., pp.381-386, 1994.
- [5] R. J. ウィルソン: グラフ理論入門, pp.46-55, 近代科学社, March 1985.