木構造図式の描画問題†

海野　浩††*　安斎公士†††　小倉耕一††††
西野哲朗††**　中西美智子　夜久竹夫††***

木構造図式は各頂点が次の四つの属性を持った属性付き木である：すなわち属性は(1)頂点の幅，(2)頂点の深さ，(3)頂点の水平座標と，(4)頂点の垂直座標である．木構造図式を与えられた美的条件を満たすように配置する問題は“美的描画問題”といわれる．木構造図式に対する，プログラム図式を指向した美的条件は，木に対する美的条件を変形することにより定式化された．その美的条件を満たす配置を与える手法も提案された．本論文でわれわれははじめに，従来の美的条件を満たす配置を与える手法を定式化し $O(n^2)$ 時間アルゴリズムを詳細に定める．次に上の美的条件に条件をひとつおきかえて新たな美的条件を考えると，われわれのアルゴリズムが新たな美的条件を満たす最も狭い配置を与えることを示す．その結果美的条件と計算量に関する新たな関係を得る．

1. まえがき

“木構造図式”は各頂点が次の四つの属性を持った木である：すなわち四つの属性は(1)頂点の幅，(2)頂点の深さ，(3)頂点の水平座標，(4)頂点の垂直座標である．木構造図式は同一でない大きさを持ち木構造状に連結された“セル”という，2次元平面上に配置された頂点からなる図式を表している．この木構造図式はデータ，アルゴリズム，プログラム[1]など情報を表示するために多くの場面で用いられている．

一般にグラフを与えられた条件を満たすように配置する問題はグラフの配置問題といわれ，VLSI の配置問題がよく知られている．われわれが扱う“美的描画問題”[2),3)]はグラフの表示を指向した問題で，木構造図式を与えられた美的条件を満たすように配置する問題である．木構造図式から属性を取り去ると木になるが，1970年頃から木に対する美的描画問題が盛んに研究されてきた．まず，2分木に対する美的条件とその美的条件に対応する線形時間の近似描画アルゴリズムの例が示され[2),3)]，次にその美的条件のもとで最小領域に2分木を描画する問題が整数座標の場合 NP 完全であることが示された[6)]．土田は Hichart 型のプログラム図式の描画問題[4)]をヒントとして，一般の n 分木の描画問題を研究し，その結果，最小領域に描画する問題が $O(n^4)$，NP 困難となるような二つの美的条件を得た[11)]．プログラム図式の描画問題は属性文法の立場からも研究されている[14),16)]．

著者の一部らはプログラム図式の生成に関する研究[7)~10),12),13),15),17)]の中で，木に対する美的条件を拡張して，木構造図式に対する美的条件を導入した．従来の美的条件[5),11)]には，親頂点が子頂点の丁度中央に配置されるという条件が含まれるが，われわれの美的条件ではプログラム図式の編集途中の描画を考慮して，つまり，ディスプレイのように一度に表示できる領域の広さが限られている場合，親を子供の中央に位置させると，その親がディスプレイの表示領域から出てしまい，表示されている子供の親がどれなのかわからなくなる場合があるため，親頂点は一番上の子頂点から j レベル ($j \geq 0$) だけ下のレベルに配置されるようになっている．同時にその美的条件に対応した描画手法が一つ提案されている．本論文で扱われるのはこの新しい美的条件である．

本研究の動機は以上のように定式化されつつある木構造図式の美的描画問題に木の描画問題の成果を適用して，美的描画問題を体系化することである．本論文の目的は，最近提案された木構造図式の美的条件[17)]にもとづいて描画アルゴリズムを体系的に記述するとと

† Tidy Drawing of Tree Structured Diagrams by HIROSHI UNNO (Faculty of Science and Engineering, Tokyo Denki University), KOUSHI ANZAI (Faculty of Economics, Kanto Gakuen University), KOICHI OGURA (Hokkaido Tokai University), TETSURO NISHINO (Faculty of Science and Engineering, Tokyo Denki University), MICHIKO NAKANISHI, TAKEO YAKU (Faculty of Science and Engineering, Tokyo Denki University).
†† 東京電機大学理工学部
††† 関東学園大学経済学部
†††† 北海道東海大学
* 現在　中央大学理工学部
Presently with Faculty of Science and Engineering, Chuo University
** 現在　北陸先端科学技術大学院大学
Presently with Japan Advanced Institute of Science and Technology, Hokuriku
*** 現在　日本大学文理学部
Presently with College of Humanities and Sciences, Nihon University

もに，最小領域美的描画問題に関する美的条件と計算量との関係を体系的に与えることである．

われわれは2章で木構造図式の美的条件について知られている結果を紹介する．3章では，既に提案された描画手法を定式化しアルゴリズムの詳細を体系的に与える．そのアルゴリズムは $O(n^3)$ の時間計算量をもつ．4章では3章のアルゴリズムと美的条件との間の詳細な関係を調べる．そのためにわれわれは既に知られている美的条件[17]を変更した新たな美的条件を導入する．すると，3章のアルゴリズムが4章の新たな美的条件を満たす最も狭い配置を与えることが示される．すなわち，われわれは $O(n^3)$ の時間計算量に対応する美的条件を示す．

本論文の結果は Hichart, PAD, SPD および TSF 等の構造的プログラム図式の描画一般に応用される．

2. 準　　備

この章でわれわれは，大きさが一定でない“セル”からなる木構造図式を考え，その図式を描画する際の従来の“美的”条件[17]を概観する．なお，大きさ一定のセルからなる木構造型プログラム図式の描画に関しては，木の描画問題に関する手法が適用可能である[2),3),11)]．はじめに次の用語を定める．

定義1[17]．木構造は $T=(V, E, r, width, depth)$ である．ここで，(V, E) は木で V をセルの集合，E を辺の集合という．$r \in V$ は根セルである．$width$ は V から Z への写像で $width(p)$ はセル p の上下方向の長さ（幅という）を表す．$depth$ は V から Z への写像で $depth(p)$ は p の左右方向の長さ（深さという）を表す． □

定義2[17]．組 (T, π) を木構造図式という．ここで T は木構造で，π : T のセルの集合 $\longrightarrow Z \times Z$ は T の配置といわれる．$\pi(p)=(x, y)$ をセル p の座標という．このとき，その x 座標，y 座標は，$\pi_x(p)$, $\pi_y(p)$ でそれぞれ表される．ここで，セル p の座標はセル p の左上隅の位置を表す．プログラム図式への適用を考慮して x 軸は左から右へ，y 軸は上から下へ向かうものとする． □

木構造図式は各頂点に四つの属性 $width(p)$, $depth(p)$, $\pi_x(p)$ と $\pi_y(p)$ が割り当てられた根付き木である．われわれは子供セルに上から下へと順序がついている順序木構造図式を対象とする．木構造図式は長方形が木構造上に配置された図式を意味する．本論文では，プログラム図式への応用を考慮して，木構造図式 $D=(T, \pi)$ の幅 $width(T, \pi)$ を次のように定める：

$$width(T, \pi) \equiv \max\{\pi_y(p)+width(p)-\pi_y(q)-1 \mid \text{ただし，} p \text{と} q \text{は} T \text{の中のセルで，} \pi_y(p)>\pi_y(q)\}.$$

セル p のレベル $level(p)$ は，p と根セルとの間の辺の個数で定義される．また，セル p に対して，関数 $Index$ を次のように定める：

$$Index(p) \equiv \begin{cases} 0 & p \text{が根セルであるとき} \\ i & p \text{が} p \text{の親の} i \text{番目の子供であるとき.} \end{cases}$$

木構造の配置に関するいくつかの条件を以下のように定める．

条件C0[2),17)]．木構造図式 (T, π) において，セル p と q のレベルが等しく，$\pi_y(p)<\pi_y(q)$ であるならば $\pi_y(q\text{の長男})>\pi_y(p\text{の末っ子})+width(p\text{の末っ子})$，すなわち線が交差しない．また，セル p に対して，

$$area(p, \pi) \equiv \{(x, y) \mid \pi_x(p) \le x \le \pi_x(p)+depth(p)-1, \pi_y(p) \le y \le \pi_y(p)+width(p)-1\}.$$

としたとき，すべての p，q $(p \ne q)$ について

$$area(p, \pi) \cap area(q, \pi)=\phi.$$

すなわちセル同士が重ならない． □

次の条件C1はプログラム図式特有のものであり，論理上の階層レベルと見かけ上の階層レベルを一致させるための条件である．

条件C1[2),17)]．木構造図式 (T, π) において，セル p が子供セルを持つならば $depth(p)=1$．また，レベルが i であるすべてのセル p に対して，$\pi_x(p)=i+1$． □

条件C2[2),17)]．木構造図式 (T, π) の部分木構造 T_1 と T_2 が位相同型ならば，T_1 と T_2 は平行移動に関して同型に配置される． □

従来の美的条件[5),11)]には，親頂点が子頂点の丁度中央に配置されるという条件が含まれる．しかし，ディスプレイのように一度に表示できる領域の広さが限られている場合，親を子供の中央に位置させると，その親がディスプレイの表示領域から出てしまい，表示されている子供の親がどれなのかわからなくなる場合がある．そこで，中心へ配置するという条件の代わりにわれわれはつぎの条件を考える．

条件C3(j)[17]．木構造図式 (T, π) において，セル p が k 個の子供 $q_1, \cdots, q_k$ $(Index(q_i)=i, 1 \le i \le k)$ をもつならば，$\pi_y(p)=\pi_y(q_1)+\min\{j, \pi_y(q_k)-\pi_y(q_1)\}$． □

木構造図式の集合から整数の集合への関数 Inter-

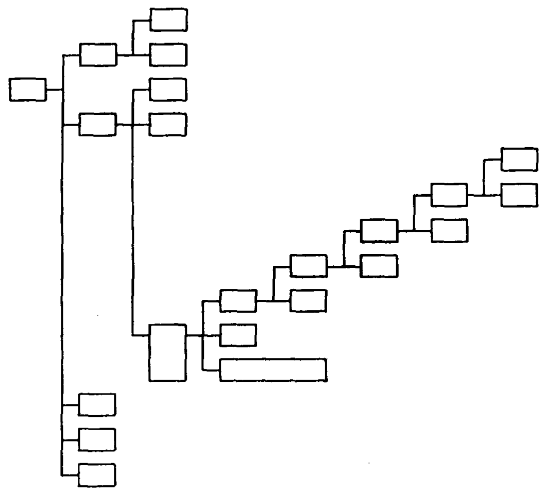


図 1 $E_0(1)$ を満たす木構造図式の例
Fig. 1 A tree structured diagram satisfying Condition $E_0(1)$.

sect を次のように定める：

$Intersect(T,\pi)\equiv\max\{\pi_y(p)-\pi_y(q)+1$；

T_1 と T_2 は，根セル同士が兄弟セルであり，$Index$ (T_2 の根セル) $<$ $Index$ (T_1 の根セル) であるような T の任意の部分木構造．p，q は T_2 と T_1 の任意のセルとする$\}$．

次の条件は部分木同士の水平方向の重なりに関するものである．n 進木の描画問題ではこの重なりの値により計算量が異なることが知られている[11]．

条件 C 4(k)[11),17)]．(T,π) を木構造図式とする．$k\geq 0$ に対して，配置 π が

$Intersect(T,\pi)\leq k.$

を満たす． □

以上の条件を組み合わせて"美的条件"を導入する．

記法 1 木構造図式の美的条件 $E_0(j)$ と $E_*(j)$ を以下のように定める：

$E_0(j)\equiv C0\wedge C1\wedge C2\wedge C3(j)\wedge C4(0).$

$E'_*(j)\equiv C0\wedge C1\wedge C3(j).$ □

図 1 に条件 $E_0(1)$ を満たす木構造図式の例を示す．

3. 描画アルゴリズム

この章では前章で紹介された美的条件 $E_*(j)$ に注目し，それを満たすような配置をあたえるアルゴリズムについて考える．プログラム生成手法[17)]の中で述べられた方法を定式化し，詳細に定める．ただし，そのアルゴリズム[17)]では概略部分は土田[11)]に従うことを仮定しているので時間計算量は $O(n^4)$ である．ここでは，今回の詳細化にともない $O(n^3)$ 時間アルゴリズムを新たに導入する．

木構造 T を美的条件 $E_0(j)$，$(j\geq 0)$ を満たすように最小幅に配置することを T の配置初期化といいその配置 π_{init} を初期配置という．アルゴリズム $Init(T,j,v,\pi_{init})$ に対し，$T=(V,E,r,width,depth)$ として $Init(T,j,r,\pi_{init})$ を実行すれば T の初期配置が得られる．

アルゴリズム Init：配置初期化

［形式］ $Init(T,j,v,\pi_{init})$

［入力］

$T=(V,E,r,width,depth)$,
j : $E_*(j)$ の j,
$v\in V$

［出力］ π_{init} : $E_0(j)$ を満たす T の最小幅配置

［方法］ 以下のアルゴリズムにおいて，l，$free$ を全域変数とする．また，$free$ の初期値を 1 とする．

```
begin
  if v は葉かつ
      これ以前に葉は見つかっていない then
      begin
        l:=level(v)+1;
        π_init(v):=(l, free);
        free:=free+width(v)
      end
  else if v は葉 then
      begin
        π_init(v):=(level(v)+1, free);
        free:=free+width(v)
      end
  else
      begin
        s1:=v の最初の子供;
        s2:=v の最後の子供;
        for s:=s1 to s2 do Init(T, j, s, π_init);
        π_init(v):=(level(v)+1, π_init_y(s1)+
          min{j, π_init_y(s2)−π_init_y(s1)})
      end
end.
```

上のアルゴリズム $Init$ の時間計算量は明らかに $O(n)$ である．

美的条件 $E_0(j)$，$(j\geq 0)$ を満たすように初期配置された木構造 T の配置 π_{init} に対して，条件 $E_*(j)$，$(j\geq 0)$ を満たしながらさらに狭い領域に再配置する

ことを配置改良と呼ぶことにする．はじめに配置改良のためにいくつかの用語を導入する．

定義 3 木構造 $T=(V, E, q, width, depth)$, 配置 π と $j \in Z$ に対して，集合 $U_j=\{v \mid v \in V, \pi_x(v)=j\}$ を考える．このとき，

$$U_j=\{v_1, \cdots, v_k \mid \pi_y(v_i)<\pi_y(v_{i+1})(1 \leq i<k)\}.$$

と書けるとき，v_i の上方移動可能距離 $Move\_v_i\_\pi$ を次のように定義する．

$$Move\_v_i\_\pi \equiv \begin{cases} \pi_y(v_i)-1 & (i=1) \\ \pi_y(v_i)-\pi_y(v_{i-1})-width(v_{i-1}) & (2 \leq i \leq k). \end{cases}$$

□

定義 4 木構造 $T=(V, E, q, width, depth)$ と配置 π に対して $i=\min\{\pi_x(v) \mid v \in V\}, k=\max\{\pi_x(v) \mid v \in V\}$ とおく．各 $j\ (i \leq j \leq k)$ に対して，$U_j=\{q \mid q \in V, \pi_x(q)=j\}$ とする．このとき集合

$$S=\bigcup_{i \leq j \leq k}\{z \in U_j \mid v \in U_j \text{ に対して，} \pi_y(z) \leq \pi_y(v)\}$$

を (T, π) の上辺をなすセルの集合という． □

定義 5 集合 $\{z_1, \cdots, z_k\}$ $(k \geq 1)$ を，木構造図式 (T, π) の上辺をなすセルの集合とし，各 i $(1 \leq i \leq k)$ に対して m_i を z_i の上方移動可能距離とする．このとき次のように定義される $Move\_T\_\pi$ を (T, π) の上方移動可能距離という，ここで

$$Move\_T\_\pi \equiv \min\{m_i \mid 1 \leq i \leq k\}.$$ □

セル q を根とする部分木構造 T_q の配置改良は以下の方法で決まる：

アルゴリズム SubOpt：部分木構造の配置改良

［形式］ $SubOpt(T_q, \pi_{in}, j, \pi_{out})$

［入力］

$T_q=(V, E, q, width, depth)$

π_{in}：最適化前の配置

j ：整数，$E_*(j)$ の j

［出力］ π_{out}：$E(j)$ を満たす π_{in} より幅が狭いか等しい T_q の配置

［方法］

begin
　if q は葉 **then**
　　begin
　　　（＊*Shift* により (T_q, π_{in}) に上方移動を行う＊）
　　　$Shift(T_q, \pi_{in}, j, \pi_{out})$
　　end
　else（＊ q は子を持つ＊）
　　begin
　　　$Sons(q)=\{p_1, \cdots, p_k\}$ $(k \geq 1)$ を q の子供セルの集合とする．
　　　（＊ q の子供を根とする部分木構造 T_{p_i} $(1 \leq i \leq k)$ の配置改良を行う＊）
　　　for $i=1$ **to** k **do**
　　　　begin
　　　　　（＊部分木構造 T_{p_i} の配置改良を行う＊）
　　　　　$SubOpt(T_{p_i}, \pi_{in}, j, \pi_{out})$;
　　　　　$\pi_{in} := \pi_{out}$
　　　　end;
　　　$\pi''_x(q) := \pi_{in_x}(q)$;
　　　$\pi''(q)$ を条件 $C3(j)$ により設定する；
　　　$\pi''(s) := \pi_{in}(s)$ $(s \neq q)$;
　　　（＊*Shift* により (T_q, π'') に上方移動を行う＊）
　　　$Shift(T_q, \pi'', j, \pi')$;
　　　$\pi_{out} := \pi'$
　　end {else}
end. {*SubOpt*}

［時間複雑度］ *SubOpt* 自身を n 回，各 *SubOpt* で *Shift* を 1 回呼ぶ．*Shift* の複雑度については以下で述べる．

次のアルゴリズム *Shift* は部分木構造を上方へ平行移動する．$Move\_T_q\_\pi_{in}$ は，(T_q, π_{in}) が他の部分木構造の配置と重ならないで，かつ最も上の配置となるまでの y 軸方向の平行移動可能距離である．

アルゴリズム Shift：部分木構造の上方移動

［形式］ $Shift(T_q, \pi_{in}, j, \pi_{out})$

［入力］

$T_q=(V, E, q, width, depth)$

π_{in}：T_q の配置

j ：整数，$E_*(j)$ の j

［出力］ π_{out}：T_q の配置：再配置

［方法］

begin
　if $\pi_{in}(q)=(l, 1)$ **then**
　　begin
　　　$\pi_{out} := \pi_{in}$;
　　　return
　　end;

if セル q は長男でない **then**

begin

- $\{z_1, \cdots, z_k\}$ $(k\geq 1)$ を (T_q, π_{in}) の上辺をなすセルの集合とするとき，各 i $(1\leq i\leq k)$ に対して (z_i, π_{in}) の上方移動可能距離 m_i を求め，それらの最小値を (T_q, π_{in}) の上方移動可能距離 $Move\_T_q\_\pi_{in}$ とする (時間複雑度：各 i に対して $O(n)$ 全体で $O(n^2)$)；
- 部分木構造 T_q の配置 π_{in} を y 軸負方向へ値 $Move\_T_q\_\pi_{in}$ だけ平行移動させ π_{out} とする (時間複雑度：$O(n)$).

end

else $\pi_{out}:=\pi_{in}$

end. {*Shift*}

［時間複雑度］ $O(n^2)$

以上により木構造 T の配置はつぎのアルゴリズム *LayOut* で得られる．

アルゴリズム LayOut：木構造の配置

［形式］ $LayOut(T, j, \pi)$

［入力］

$T=(V, E, r, width, depth)$,

j ：整数，$E_*(j)$ の j

［出力］ π : $E_*(j)$ を満たす T の配置

［手法］

begin

$Init(T, j, r, \pi_{init})$;

$SubOpt(T, \pi_{init}, j, \pi)$

end.

［時間複雑度］ 定理2参照

定理1 アルゴリズム *LayOut* は条件 $E'_*(j)$ を満たすアルゴリズム *Init* より狭いか等しい幅の木構造図式を出力する．

定理2 木構造図式のセルの個数を n としたとき，アルゴリズム *LayOut* の計算量は $O(n^3)$ である．

証明．セルを深さ優先に訪れ，各セルで *SubOpt* を1回呼ぶ．□

4. 最　適　性

この章では，3章で導入されたアルゴリズム *LayOut* の出力が最小幅配置となるような美的条件を示す．はじめに *LayOut* が必ずしも条件 $E'_*(j)$ に対する最小幅配置を与えるとは限らないことを示す．

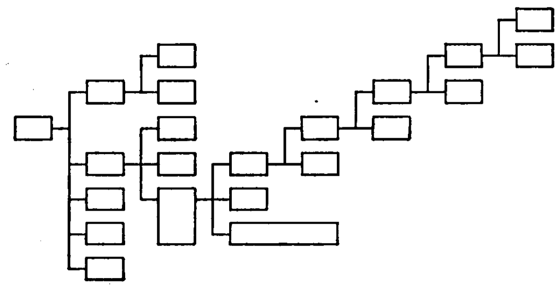


図2 図1に対するアルゴリズム *LayOut* の出力
Fig. 2 The output of Algorithm *LayOut* for Fig. 1.

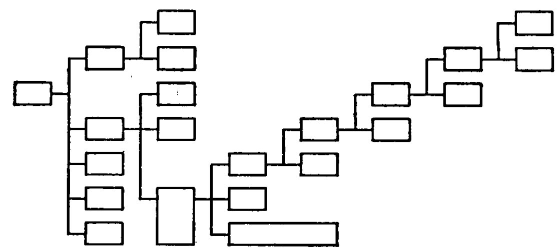


図3 図1に対する $E_*(1)$ のもとでの最小幅配置
Fig. 3 The narrowest placement for Fig. 1 under Condition $E_*(1)$.

定理3 アルゴリズム *LayOut* は美的条件 $E'_*(j)$ を満たす最小幅の木構造図式を与えるとは限らない．

証明．次の二つの例により示される．

図1に対する *LayOut* の出力（**図2**）および図1に対する $E_*(1)$ のもとでの最小幅配置（**図3**）．□

次に新たな美的条件を考えるため，以下の条件C5 (j) を導入する：

はじめに木構造図式 $C=(U, F, s, width', depth', \pi')$ と $D=(V, E, r, width, depth, \pi)$ の<u>距離</u>を

$$Dist(C, D) \equiv \min_{x\in Z}\{|\pi'_y(p)-\pi_y(q)|;\ p\in U, q\in V, \pi'_x(p)=\pi_x(q)=x\}$$

と定める．距離は<u>森構造図式</u>に対しても同様に定められる．

定義6 $D=(V, E, r, width, depth, \pi)$ を木構造図式とし，$D(v)=(V_v, E_v, v, width_v, depth_v, \pi_v)$ を D のセル v を根とする D の部分木構造図式とする．$j\geq 0$ に対して $D(v)$ が<u>準 j-局所最小である</u>とは次の条件を満たすことである．

1. $D(v)$ が D の葉である時 $D(v)$ は準 j-局所最小である．
2. $D(v)$ が D の葉でない時 $D(v)$ は次の (1),

(2), (3)をすべて満たす.

(1) v の子供を長男セルから順に $v_1, v_2, \cdots, v_n$ とする. $v_2, \cdots, v_n$ を根とする $D(v)$ の部分木構造図式 $D(v_2), \cdots, D(v_n)$ が準 j-局所最小であり

(2) $\pi_y(v)=\pi_y(v_1)+\min\{j, \pi_y(v_n)-\pi_y(v_1)\}$

(3) 各 $k(1\leq k\leq n-1)$ と $D(v_{k+1})$ より上にある D の部分森構造図式 $\bar{D}(v_{k+1})$ に対して

$$Dist(\bar{D}(v_{k+1}), D(v_{k+1}))=1 \qquad \square$$

定義 7 $j\geq 0$ に対し, 木構造図式 $D=(V, E, r, width, depth, \pi)$ が j-局所最小であるとは (1) D が準 j-局所最小であり, かつ (2) 長男セルを優先する深さ優先探索において最初に見つかる葉セル v に対して

$$\pi_y(v)=1, \quad \pi_x(v)=level(v)+1$$

が成り立つことである. □

条件 C5(j) 木構造図式 D が j-局所最小である.

上の条件 C5(j) にもとづき美的条件 $E'_*(j)$ を強めて, 次の美的条件 $F_*(j)$ を導入する:

記法 2 $F_*(j)\equiv E'_*(j)\wedge C5(j)$. □

このとき美的条件 $F_*(j)$ に対して次が成り立つ.

定理 4 与えられた $j\geq 0$ に対してアルゴリズム *LayOut* は美的条件 $F_*(j)$ を満たす最小幅配置を出力する.

証明. T に対するアルゴリズム *LayOut* の出力を $D=(T, \pi)$ とする. D は定理 1 より美的条件 $E'_*(j)$ を満たす. またアルゴリズム *Shift* により D は条件 C5(j) も満たしている. なぜなら, もし美的条件 $E'_*(j)$ を満たす D より狭い木構造図式 (T, π') が存在したとすると (T, π') は C5(j) を満たさない. □

なお, 美的条件 $E'_*(j)$ において親セルを子供セルの上から j 番目へ配置する代わりに子供セルの中心へ配置することを考え, なおかつ部分木間の重なりについてある種の制限をつけた条件を $E_*(\$)$[11] とする. このときセルの大きさを一定とすると $E_*(\$)$ を満たす最小幅配置は $O(n^4)$ 時間で得られることが知られている[11].

5. おわりに

木に対しては扇型に配置される美的条件とその美的条件に対応する時間計算量の研究が行われ, 既に得られていた成果[11]はセルの大きさが一定で扇型に配置された木構造図式に対して適用可能であった. セルの大きさが一定でない木構造図式に対しては, プログラム図式を指向した美的条件 ($E_*(j)$) と非定式的な描画手法だけが知られていた[17]. 本論文でわれわれは後者の美的条件 $E_*(j)$ を満たす配置を与える $O(n^3)$ 時間アルゴリズムを得た. さらにわれわれは従来の美的条件 $E_*(j)$ を変更して新たな条件を加えた美的条件 $F_*(j)$ を導入した. この時, われわれの $O(n^3)$ アルゴリズムが新たな美的条件 $F_*(j)$ を満たす最小の配置を与えることを証明した. その結果美的条件と計算量についての新たな関係を得た. すなわち, 木の描画問題の立場から考えると従来の結果は $O(n^4)$ 時間[11]と NP 完全性[3]に対応する美的条件であり, 今回新たに $O(n^3)$ 時間に対応する美的条件が知られたことになる.

本来の美的条件 $E_*(j)$ に対応する計算量を評価する課題は未解決である. そのほかの今後の課題としては, さらに美的条件のバリエーションを増やしそれらと計算量との関係を調べることがある. その中には例えば, 位相同型な部分木構造を同型に配置するという条件 C2 が NP 完全性の十分条件かどうかという問題がある.

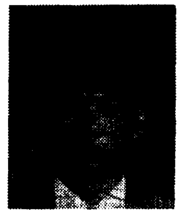
謝辞 著者らはアルゴリズムの設計に際して助言をくださった郷信義氏 (東海大学計算センタ), 近藤理彦氏 (早稲田大学, 現日本アイビーエム(株)), 岸本美紀氏 (東海大学, 現富士通(株)), 竹内一浩氏 (東海大学, 現日本電気(株)), 桑原博氏 (早稲田大学, 現富士通(株)), 今井一昭氏 (東海大学, 現(株)CSK) に感謝します. またアルゴリズム *LayOut* の検証に際して助言をくださった杉田公生氏 (東海大学), 土田賢省氏 (神奈川大学, 現 東洋大学), 米田信夫氏・宮寺庸造氏 (東京電機大学), 細井潔氏 (東京電機大学電子計算機センタ) に感謝いたします. 第 3 著者と第 6 著者は (株)ガロアの委託研究, および東京電機大学総合研究所学術研究援助を受けています.

参考文献

1) 夜久竹夫, 二木厚吉: フローチャートの木構造型記法, 信学技報, Vol. AL 78-47, pp. 61-66 (1978).
2) Wetherell, C. and Shannon, A.: Tidy Drawing of Trees, *IEEE Trans.*, Vol. SE-5, pp. 514-520 (1979).
3) Reingold, E. M. and Tilford, J. S.: Tidy Drawing of Trees, *IEEE Trans.*, Vol. SE-7, pp. 223-228 (1981).
4) 郷 信義, 岸本美紀, 高橋美智子, 長田芳一, 夜久竹夫: 木フローチャート記述言語 Hichart の処理系実現について, 第 27 回情報処理学会全国大会論文集, pp. 549-550 (1983).

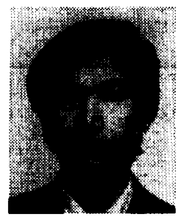
5) Supowit, K. J. and Reingold, E. M.: The Complexity of Drawing Trees Nicely, *Acta Inf.*, Vol. 18, pp. 377-392 (1983).
6) 岡田謙一，北川 節：PAD によるソフトウェア開発システム，情報処理学会論文誌，Vol. 26, No. 5, pp. 898-904 (1985).
7) 郷 信義，小倉耕一，竹内一浩，土田賢省，近藤理彦，岸本美紀：階層的フローチャート言語 Hichart に対するオートフローチャータの実現，日本ソフトウェア科学会構造エディタに関するワークショップ予稿，15 p. (1986).
8) 夜久竹夫，二木厚吉，足立暁生，番場 浩：階層的流れ図言語 Hichart の情報処理記号，早稲田大学情報科学研究教育センター紀要，Vol. 3, pp. 92-107 (1986).
9) 大原茂之：木構造化チャートによるプログラム開発・保守技法，情報処理学会論文誌，Vol. 27, No. 10, pp. 1019-1026 (1986).
10) Miyadera, Y., Imai, K., Kuwabara, H., Unno, H. and Yaku, T.: ETA87—An Extension of a Hichart Flowchart Processing System, 第 35 回情報処理学会全国大会論文集，pp. 1201-1202 (1987).
11) Tsuchida, K.: The Complexity of Tidy Drawings of Trees, *Topology and Computer Science* (Suzuki, S. ed.), pp. 487-520, Kinokuniya, Tokyo (1987).
12) Yaku, T., Futatsugi, K., Adachi, A. and Moriya, E.: Hichart—A Hierarchical Flowchart Description Language—, *Proc. IEEE COMPSAC*, Vol. 11, pp. 157-163 (1987).
13) 夜久竹夫，杉田公生，二木厚吉，守屋悦朗：Hichart とプログラム開発環境 ETA_AIDE, 原田賢一(編)，構造エディタ，第Ⅲ部6章，共立出版，東京 (1987).
14) 西野哲朗：属性グラフ文法とその Hichart 型プログラム図式に対するエディタへの応用，コンピュータソフトウェア，Vol. 5, pp. 81-92 (1988).
15) 郷 信義，小倉耕一，土田賢省，夜久竹夫，岸本美紀：Hichart 流れ図の描画アルゴリズム，日本ソフトウェア科学会第5回大会論文集，pp. 181-184 (1988).
16) Nishino, T.: Attribute Graph Grammars with Applications to Hichart Program Chart Editors, *Advances in Software Science and Technology*, Vol. 1, pp. 426-433 (1989).
17) 郷 信義，岡田直之，岸本美紀，土田賢省，宮寺庸造，夜久竹夫：Hichart プログラム図式の生成手法，情報処理学会論文誌，Vol. 31, No. 10, pp. 1463-1473 (1990).

(平成3年8月13日受付)
(平成4年4月 9 日採録)



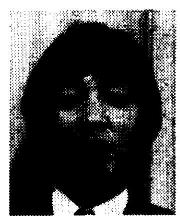
海野 浩（正会員）

1963 年生．1987 年東京電機大学理工学部数理学科卒業．1987 年～92 年大原簿記学校税理士科在学．1992 年2月東京電機大学理工学部情報科学科研究生．1992 年4月より中央大学理工学部数学科技術職員として勤務．コンピュータおよびコンピュータネットワークの運用と教育に従事．オートマトン・言語理論，図形型プログラム言語，計算量理論に興味を持つ．



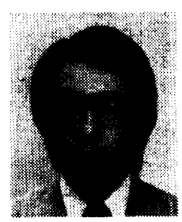
安斎 公士（正会員）

1961 年生．1984 年東海大学理学部情報数理学科卒業．1986 年同大学院理学研究科数学専攻博士課程前期修了．同年関東学園大学経済学部経営学科助手．1991 年同講師．グラフ言語，ソフトウェアの開発環境の研究に従事．LA, JAPAN-FIG 各会員．



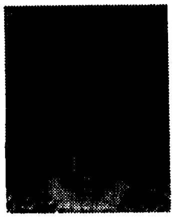
小倉 耕一

1963 年生．1986 年東海大学理学部情報数理学科卒業．現在，北海道東海大学電子計算センターに勤務．グラフ言語，オートマトン理論およびソフトウェアの開発環境等に興味を持つ．



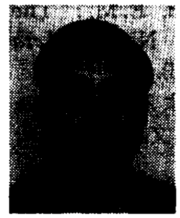
西野 哲朗（正会員）

昭和 34 年生．昭和 57 年早稲田大学理工学部数学科卒業．昭和 59 年同大学院理工学研究科博士前期課程修了．同年日本アイ・ビー・エム(株)入社．昭和 62 年東京電機大学理工学部情報科学科助手．平成4年北陸先端科学技術大学院大学助教授．理学博士．属性文法，自然言語の基礎理論，計算論的学習理論の研究に従事．情報処理学会学会誌編集委員会主査．電子情報通信学会，日本ソフトウェア科学会，LA 各会員．



中西美智子（旧姓高橋）

1955年8月15日生．1978年早稲田大学教育学部理学科数学専修卒業．1980年4月～1986年3月東海大学理学部情報数理学科技術職員．ソフトウェア工学の研究に従事．



夜久　竹夫（正会員）

1947年生．1970年自由学園大学部理学科卒業．1972年早稲田大学大学院数学専攻修士課程修了．1976年同大学院博士課程単位取得退学．理学博士．1976年東海大学理学部情報数理学科専任講師．1979年同助教授．1985年東京電機大学理工学部数理学科助教授．1986年同情報科学科助教授．1992年日本大学文理学部応用数学科教授，現在に至る．セルオートマトン，図形型プログラム言語，グラフアルゴリズム等の研究に従事．日本数学会，電子情報通信学会，ACM 各会員．