アルゴリズム　23－11
（1991.　9．20）

過半異色ラベル付問題の並列化可能性について†

岩本 宙造　太田 英憲　岩間 一雄

九州大学工学部

過半異色ラベル付問題 (Different Than Majority Labeling problem, DTML 問題) は，各頂点をその近傍の過半数を占める色と異なるように色 0,1 をラベル付する問題であり，NC に入るかどうかが判っていない著名な問題の一つである．本稿では，(i) 全ての頂点のラベルをある規則に従って一意に決定できるグラフに対して DTML を求める問題は P 完全であること，(ii) 入力グラフの次数を 3 以下に制限した DTML 問題は NC であることを示す．また，乱数を用いた発見的並列アルゴリズムを示し，次数が定数の場合このアルゴリズムが多項式対数時間で動作するのではないかという予想を可能にする実験結果を紹介する．

Parallelizability of the Different than Majority Labeling Problem†

Chuzo Iwamoto, Hidenori Ohta, and Kazuo Iwama

Department of Computer Science and Communication Engineering
Kyushu University

Fukuoka 812, Japan

The Different Than Majority Labeling (DTML) problem is to label all vertices by either color 0 or color 1 such that each vertex is colored differently from the majority of its neighbors. It is one of the common problems for which it is not known whether or not NC algorithms exist. We show that the DTML problem becomes P-complete for such graphs that (i) labels of two vertices are initially fixed and (ii) labeling of the rest of vertices can be carried out in a deterministic fashion. It is also shown that the DTML problem for graphs of degree three (or less) is in NC. A heuristic parallel algorithm is also shown which runs well especially for graphs of small degrees.

†科学研究費 01302059, 02302047, 02650278 の補助を受けた．

1 はじめに

過半異色ラベル付問題 (Different Than Majority Labeling problem, DTML 問題) は，各頂点に接続する枝の過半数を，その枝の両端の頂点の色が異なるように全頂点に色 0,1 をラベル付する問題である．頂点のラベル付問題がクラス NC に含まれていることを示す方法としては，多項式対数時間で少なくとも $\Omega(n/\log^s n)$ 頂点のラベルが決定できることを示す方法が良く用いられる [3]． DTML 問題には頂点のラベルを局所的に決定する高速なアルゴリズムは存在しないことが知られており [7]，並列化が難しい問題として認識され，クラス NC に含まれるかどうかは未解決の問題として残されている． [5] は，各頂点ごとではなく，単にグラフ全体の中で過半数の枝が両端の頂点の色が異なるようにラベル付する問題について， NC アルゴリズムが存在することを示した．一方， [7] は DTML を構成する単純な直列アルゴリズムを示し，辞書式順序優先 DTML 問題は， P 完全であることを示した．このように， DTML 問題に関しては，問題を難しくして P 完全を示すアプローチと易しくして並列アルゴリズムを示すアプローチによって問題の解決に迫るための幾つかの知見を得ているが，いまだ不十分であることは否定できない．

本稿では， DTML 問題の計算複雑さをより明らかにするため，前者のアプローチとして，ある特別な入力に対して DTML を構成する問題について考察する．そのために，全ての頂点のラベルをある規則に従って一意に決定できるグラフ（決定性グラフ）を定義し，決定性グラフに対して DTML を求める問題は P 完全であることを示す．また，後者のアプローチとして入力グラフの次数を 3 以下に制限した DTML 問題に対して NC アルゴリズムが存在することを示す．更に，乱数を用いた発見的並列アルゴリズムを示し，次数が定数の場合このアルゴリズムが多項式対数時間で動作するのではないかという予想を可能にする統計結果 を与える．

2 決定性 DTML 問題の P 完全性

本節では，一般の DTML 問題をより難しくして P 完全性を示す．このアプローチとして，辞書式順序優先 DTML 問題が P 完全であることの証明が知られている．さらに，自然な拡張として，入力グラフの頂点の一部を最初からある色に固定したグラフに対して DTML を構成する問題も考えられている [4]．ここで注意することとして，グラフの 1 頂点（v_s とする）のラベルを 1 に固定した場合は，その v_s 以外の全頂点のラベルを反転させることで， s のラベル 0 に反転させることと全く同等のことを行なうことが可能であるのでラベルを固定することに意味を持たない．ここでは， 2 頂点のラベルを固定したグラフを考察の対象とする．しかし， 2 頂点のラベルを固定したグラフの DTML を求める問題（2-DTML 問題）は，難しさが急激に増してしまい NP 完全になる [4]．そこで，本稿ではある特別な入力に対して 2-DTML を構成する問題について考察する．そのために，全ての頂点のラベルをある規則に従って決定してゆけるグラフ（決定性グラフ）を定義し，決定性グラフに対して 2-DTML を求める問題（決定性 2-DTML 問題）は P 完全であることを示す．

2.1 決定性グラフ

本稿では，頂点 v の近傍の頂点の集合を $N(v)$ で表す．頂点集合 S に対し， $S[l]$ は S に含まれかつ既に色 l にラベル付されている頂点の集合を表し， $|S|$ は S に含まれる頂点数を表すものとする．頂点 v の次数を $d(v)$ で表す．頂点 v のラベルを $l(v)$ で表す．このとき， $\overline{l(v)}$ は $l(v)$ の反転色を表し， $l(v)=0$ のとき色 1， $l(v)=1$ のとき色 0 を表す．頂点 v のラベルが *correct* であるとは， v の近傍の過半数の頂点が $\overline{l(v)}$ にラベル付されている状態（つまり， $\left|N(v)[\,\overline{l(v)}\,]\right| \geq \left\lceil \frac{d(v)}{2} \right\rceil$）をいう．そうでないときを *incorrect* という．

DTML に関して，次の性質が成り立つことは明らかである．

[性質 1]

(1) 近傍の頂点のうち，少なくとも $\left\lceil \frac{d(v)}{2} \right\rceil$ 個の頂点のラベルが色 l なら， v のラベルは $\bar{l}$ である．

(2) 次数が奇数で互いに同じ近傍を持つ 2 頂点に対し，一方の頂点のラベルが色 l なら，他方のラベルも l である．

この性質を用いて全ての頂点のラベルを一意に決定できるグラフを定義する．

[定義 1] 一部の頂点にラベル付されているグラフにおいて，まだラベル付されていない頂点 v のラベルを l_t に局

所決定可能であるというのは，v が次の (i),(ii) のいずれかを満たす場合を言う．

(i) $\left|N(v)[\,\overline{l_t}\,]\right| \geq \left\lceil \frac{d(v)}{2} \right\rceil$.

(ii) v，及び v と性質 1-(2) の関係を満たす頂点の全てのラベルを $\overline{l_t}$ にすると，$l(u) = \overline{l_t}$ かつ $\left|N(u)[\,\overline{l_t}\,]\right| \geq \left\lceil \frac{d(u)}{2} \right\rceil$ なる頂点 $u \in N(v)$ が存在する（頂点 u の既に決定済みのラベルが *incorrect* となる）．

[定義 2] ラベルが色 1 に最初から決定された特別な 2 頂点を含むグラフから始め，局所決定できる頂点の中で任意の一つの頂点のラベルを決定する，という手続きを再帰的に適用することにより全ての頂点のラベルを決定できるグラフを決定性グラフと呼ぶ．決定性グラフの DTML を求める問題を決定性 DTML 問題と呼ぶ．

[定理 1] 決定性グラフの DTML は一意に決まる．

(証明)　今，決定性グラフ $G = (V, E)$ に対し，$V_{uni} \subseteq V$ へのラベル付が一意であることが分かっており，それに従ってラベル付されているとする．このとき，定義 2 より，V_{uni} に含まれない少なくとも一つの頂点は局所決定可能であるか，または $V = V_{uni}$ である．局所決定可能な頂点 v が，定義 1-(i) を満たす場合は，そのラベルは（一意にラベルが決まった近傍の頂点より）一意に決まる．定義 1-(ii) を満たす場合も同様で，頂点集合 $V_{uni} \cup \{v\}$ のラベルも一意に決まることになる．このことから，定理は明らか．□

2.2　P 完全性の証明

本稿では，P 完全性の証明は，P 完全問題として知られている単調論理回路値問題 (Monotone Circuit Value Problem, MCVP)[7] を対数領域で帰着することで行なう．MCVP は，次の様に定義される．問題のインスタンスを有向非巡回グラフ $G = (V, E)$ とする．但し，頂点は関数頂点または入力頂点のいずれかで，$\beta_1, \beta_2, \cdots, \beta_n$ で表す．関数頂点は，関数ラベル AND, OR にラベル付されている．AND（または OR）にラベル付された頂点を ∧ 頂点（∨ 頂点）と記す．関数頂点の入次数は 2，出次数は高々 2 とする．入力頂点は論理値 *true* または *false* なる値にラベル付され，入次数を 0，出次数を 1 である．各関数頂点 β_i に対し (y, β_i) を β_i へ入ってくる枝とする．このとき，各 y は入力頂点または $y = \beta_j$（但し $j < i$）である．MCVP はこのようなインスタンスに対し，その頂点を指す頂点の値と割り当てられた論理関数に従って頂点の値を計算してゆくとき，頂点 β_n の値は *true* となるかを問う問題である．

[定理 2] 決定性 DTML 問題は，クラス P に含まれる．

(証明)　定義 1 によりラベルが決定できる頂点を多項式時間で見つけることが可能であることを示すだけで十分である．定義 1-(i) は容易であるので説明は省略する．(ii) も同じ近傍をもつ頂点ごとにグループ分けしておけば容易に求まる．

[定理 3] 決定性 DTML 問題は P 完全である．

(証明)　MCVP のインスタンスを $G = (V, E)$ を決定性 DTML 問題のインスタンス $G' = (V', E')$ に対数領域で変換する．ここでは，G の関数頂点の数を n，入力頂点の数を高々 $n_i \leq n$ とする．G' のラベルが色 1 に固定された特別な 2 頂点を v_s, v_t とする．頂点集合 S_1, S_2 をそれぞれ $16n^2 + 66n + 1$, $8n + 34$ 個の頂点からなる集合とし，S_1 に含まれる各頂点と頂点 v_s を枝で結ぶ．更に，S_1 と S_2 間を完全 2 部グラフ構造となるように枝で結ぶ．

次に，関数頂点 β_n, β_{n-1}, $\cdots$, β_1 の順に以下のことを行なう．今，$G = (V, E)$ の ∧ 頂点 β_i の番とする．β_i を Fig.1 に示すグラフに置き換える．但し，Fig.1 で k なる値が添えられた○印は，k 個の頂点からなる頂点集合を表すものとし，k_1, k_2 個の頂点からなる二つの頂点集合間の枝は，完全 2 部グラフ構造をした $k_1 \times k_2$ 本からなる枝集合を表す．次に，各頂点集合 $V_{i,8}$, $V_{i,18}$, $V_{i,20}$, $V_{i,25}$, $V_{i,28}$ と S_2 間をそれぞれ完全 2 部グラフで接続する．G において β_i から $\beta_h, \beta_l (h, l > i)$ へ枝が存在するとき，頂点集合 $V_{i,12}$ から $V_{h,1}$, $V_{h,2}$ の一方へ，$V_{i,12}$ から $V_{l,1}$, $V_{l,2}$ の一方へ，それぞれ完全 2 部グラフ構造で接続する．

β_i が ∨ 頂点である場合は，$V_{i,8}$ を $4n + 12$, $4n + 13$, $4n + 13$, $4n + 14$ 頂点からなる頂点集合 $V_{i,8}^1$, $V_{i,8}^2$, $V_{i,8}^3$, $V_{i,8}^4$ に置き換え，$V_{i,8}V_{i,8}^1$ 間，$V_{i,8}^1 V_{i,8}^2$ 間，$V_{i,8}^2 V_{i,8}^3$ 間，$V_{i,8}^3 V_{i,8}^4$ 間，$V_{i,8}^4 S_2$ 間を完全 2 部グラフ構造で接続する．

β_i が β_j へ枝を持ち *false* にラベル付された入力頂点の場合，β_i をそれぞれ $8n + 32, 4n + 16$ 頂点からなる $V'_{i,11}$, $V'_{i,12}$ に置き換え，S_2 と $V'_{i,11}$ 間，$V'_{i,11}$ と $V'_{i,12}$ 間，$V'_{i,12}$ と $V_{j,1}$ または $V_{j,2}$ 間を完全 2 部グラフ構造で接続する．β_i

のラベルが $true$ の場合は，β_i を頂点集合 $V'_{i,12}$ に置き換え，S_2 と $V'_{i,12}$ 間，$V'_{i,12}$ と $V_{j,1}, V_{j,2}$ の一方間を完全２部グラフで接続する．以上の操作の後，$i < n$ の場合は $V_{i,29}$ と $V_{i+1,30}$ 間を完全２部グラフで接続し，$i = 1$ の場合は $V_{i,30}$ を特別な頂点 v_t（ラベルは１）とみなし，更に，$v_{1,31}$ を取り除く．

以上の変換では，G の一つの頂点を n と i によって決まる $O(n^2)$ の大きさのグラフに置き換えるだけで良いので，対数領域で可能である．このようにして得られたグラフ G' に対し以下の補題が成り立つ．

[補題 1] S_1 に含まれるどの２頂点も同じ色にラベル付される．

(証明)　S_1 に含まれる頂点は，奇数個の同じ近傍をもつので性質１より明らかである．□

[補題 2] S_1, S_2 に含まれる頂点はそれぞれ，全て０，全て１に決定される．

(証明)　補題１より，S_1 に含まれる頂点の任意の一つのラベルを１とすると，残りの頂点も全て１とならなければならない．これと，v_s の近傍の少なくとも半数のラベルを０にすることとは両立できないことは明らか．故に，定義１の (1) より S_1 に含まれる全頂点のラベルは０である．S_2 の全ての頂点は，S_1 に近傍を $16n^2{+}66n{+}1$ で個もち，それ以外に高々 $16n^2{+}66n$ 個の近傍を持つ．S_1 に含まれる頂点のラベルが０であることから，S_2 に含まれる全頂点のラベルは全て１に決定できる．□

[補題 3] $1 \leq i < n$ を満たす全ての i に対し，$V_{i,8}$, $V_{i,18}$, $V_{i,20}$, $V_{i,25}$, $V_{i,28}$ のラベルは０に決定される．

(証明)　$V_{i,8}$, $V_{i,18}$, $V_{i,20}$, $V_{i,25}$, $V_{i,28}$ のそれぞれに含まれる頂点の近傍の過半数は S_2 に含まれていること，及び定義 1-(1) より明らか．□

[補題 4] $V_{i,j} (1 \leq i \leq n,\ 1 \leq j \leq 31)$ に含まれるどの２頂点も同じ色にラベル付される．

(証明)　$V_{i,8}$, $V_{i,18}$, $V_{i,20}$, $V_{i,25}$, $V_{i,28}$ に関しては補題３で証明済み．その他の頂点集合に関しては，補題１と同様に証明できる．□

[補題 5] $1 \leq i \leq n$ を満たす全ての i について，$V_{i,30}$ に含まれる全ての頂点のラベルは１に決定される．

(証明)　i に関して帰納的に証明する．

初期段階：$i = 1$ のとき，頂点 $V_{1,30}$ に含まれる頂点はラベルが１に固定された特別な頂点 v_t のみである．

帰納段階：今，頂点 $V_{i,30}$ に含まれる頂点のラベルは１に決定されているとする．$V_{i,30}$ のラベルが $correct$ となるのは，$V_{i,30}$ が $4i-3$ 個の近傍を持つことから頂点集合 $V_{i,29}$ に少なくとも一つラベルが０の頂点が存在しなくてはならない．ここで，補題４より，$V_{i,30}$ のラベルが $correct$ となるのは，頂点集合 $V_{i,29}$ の全頂点のラベルが０である場合に限られる．更に，$V_{i,28}$ のラベルが０となることに注意して，$V_{i,29}$ のラベル０が $correct$ であるためには近傍に $4i+1$ 個の色１なる頂点が必要である．言い替えれば，頂点集合 $V_{i+1,30}, V_{i,26}$ に少なくとも $2i+2$ 個の色１なる頂点が，更に言い替えれば，$V_{i+1,30}$, $V_{i,26}$ に含まれる頂点のラベルを全て１とする必要がある．□

[補題 6] $1 \leq i < n$ を満たす全ての i について，$V_{i,26}$, $V_{i,23}$ に含まれる全ての頂点のラベルはそれぞれ 1,0 に決定できる．

(証明)　定義 1-(ii) 及び近傍のラベル付より明らか．□

[補題 7] $V_{i,1}, V_{i,2}$ のラベルが決定されている場合，$V_{i,9}$ のラベルは決定可能である．この時のラベル 0,1 を論理値とみなすと，β_i が $\wedge$ 頂点のとき $l(V_{i,9}) = l(V_{i,1}) \wedge l(V_{i,2})$, $\vee$ 頂点のとき $l(V_{i,9}) = l(V_{i,1}) \vee l(V_{i,2})$ を満たす．ここで $l(V_{i,k})$ は頂点集合 $V_{i,k}$ に含まれる頂点のラベルを表す．

(証明)　今，β_i を $\wedge$ 頂点とする．$V_{i,1}$, $V_{i,2}$ のラベルが決定されているとき，$V_{i,5}$, $V_{i,6}$ のラベルはそれぞれ同じ色に決定されることは容易に分かる．$V_{i,1}$, $V_{i,2}$ の少なくとも一方の全頂点のラベルが０に決定されているとする．このとき，$V_{i,8}$ の全頂点のラベルが０に決定されることに注意して，$V_{i,7}$ の各頂点の $16n+51$ 近傍中 $8n+26$ 頂点が０に決定されたことになり，定義１の (1) より $V_{i,7}$ のラベルは１に決定可能である．また，$V_{i,7}$ のラベルが決まることにより，$V_{i,9}$ のラベルは０に決定できる．$V_{i,1}$, $V_{i,2}$ のラベルが共に１に決定されているときは同様にして，$V_{i,7}$ のラベルを０，更に $V_{i,9}$ のラベルは１に決定される．β_i が $\wedge$ 頂点の場合は省略する．□

[補題 8] $V_{i,9}$ のラベルが $l(V_{i,9})$ に決定されている場合，$V_{i,11}$ のラベルも $l(V_{i,9})$ に決定される．更に $V_{i,12}, V_{h,2}, V_{l,1}$ のラベルも $\overline{l(V_{i,9})}, l(V_{i,9}), l(V_{i,9})$ に決定される．但し，$V_{h,2}$, $V_{l,1}$ $(h,l > i)$ は，それぞれ $V_{i,12}$ と完全2部グラフで結ばれた頂点集合とする．

(証明)　今，$l(V_{i,9}) = 1$ とする．このとき定義 1-(1) より $V_{i,13}, V_{i,19}, V_{i,22}$ に含まれる全頂点のラベルをそれぞれ 0, 1, 0 と決定できる．$V_{i,23}$, $V_{i,24}$, $V_{i,26}$ の頂点のラベルがそれぞれ 0, 1, 1 となることに注意して，定義 1-(2) より $V_{i,23}$ の決定されたラベル 0 を *correct* にするためには $V_{i,21}$ のラベルを 1 にする以外ない．同様に $V_{i,17}, V_{i,14}, V_{i,10}, V_{i,11}$ の値もそれぞれ 0, 1, 0, 1 と決定できる．更に，定義 1-(1) より，$V_{i,12}$, $V_{h,2}$, $V_{l,1}$ のラベルは 0, 1, 1 に決定できる．$l(V_{i,9}) = 0$ のときも同様である．□

以上の補題から，G' は決定性グラフであることは明らかである．また，G の関数頂点 β_i の値が *true* であるのは，G' の頂点集合 $V_{i,11}$ に含まれる全頂点のラベルが 1 となる時に限ること，及び *false* であるのは，0 となる時に限ることが分かる．このことから，本定理が証明された．□

3　次数が小さいグラフに対する DTML 問題

一般のグラフに対する DTML 問題は P 完全であることが予想されている [7]．本節では次数が 3 以下のグラフに対しては，NC アルゴリズムが存在することを証明する．次数が 2 以下のグラフに対しては問題は自明となるが，3 以下の場合は易しくはない．本アルゴリズムは，二つの有名な NC アルゴリズム（全域グラフ，式の評価）を巧妙に利用している．

[定理 4] 次数が 3 以下に限定されたグラフの DTML を求める NC アルゴリズムが存在する．

(証明)　次数が 3 以下に限定された入力グラフを $G = (V, E)$ とする．簡単の為，G は一つの成分から構成されているとする．G の DTML を求めるアルゴリズムは，次のような流れである．(i) G の全域木で根の次数が 2 以下の木 T を構成する．(ii) T の葉に色 0,1 を割り当てる．(iii) T の葉以外の頂点に関数ラベル NAND, NOR を割り当て，それらの関数ラベルに従い，各頂点の色を計算する．((ii) で割り当てた色 0,1 を論理値とみなす．) (iv) 一部のラベルを反転させることにより，全てのラベルを *correct* にする．

(i) 根つき全域木を求めるアルゴリズムは，CREW-PRAM で $O(\frac{n^2}{\log^2 n})$ プロセッサ $O(\log^2 n)$ 時間のものが知られている [2]．このアルゴリズムで求まる全域木の根頂点は三つの子頂点を持つ場合もあり，以下の議論に不都合である．そこで，この木の葉で最小番号の頂点 v_l から根頂点へのパスとなる有向枝の方向を反転させて v_l を新たな根頂点とし，得られた全域木を $T = (V, E_t)$ とおく．

(ii) T の葉頂点の集合を V_l とし，V_l に含まれる 2 頂点間に存在する枝集合を E_l とする．更に，V_l と E_l で構成されるグラフを $G_l = (V_l, E_l)$ とする．G_l の次数は 2 以下であるので，G_l の成分は単純閉路，直線状のいずれかである．G_l に対し (i) と同様にして根の次数が 1 の全域森 $F = (V_l, E_f)$ を構成する．このとき，F は高々一つの親頂点，高々一つの子頂点をもつ頂点からなる木の集合であり，各木は根と葉を丁度一つずつ持つ．F の各頂点に対し根からの距離を $O(\log n)$ 時間で計算し，その値が偶数の頂点に 0，奇数の頂点に 1 を割り当てる．

(iii) 全域木 T に含まれる葉以外の全頂点に対し根からの距離を計算し，奇数なら関数ラベル NAND，偶数なら NOR をラベル付する．これは，$O(\log n)$ 時間で可能である．全頂点の色を，関数ラベルと二つの子頂点の色（論理値とみなす）により計算する．このような木を用いた式の評価に対して $O(n)$ プロセッサ，$O(\log n)$ 時間のアルゴリズムが知られている [2]．

(i) ～ (iii) で全ての頂点に色が割り当てられたことになる．この彩色を CL と呼ぶ．彩色 CL に対し，次の 2 補題が成り立つ．

[補題 9] 彩色 CL に対し T の葉でない頂点の全ラベルは *correct* である．

(証明)　いま，二つの子頂点の一方のラベルが 1，他方が 0 で，関数ラベルが NAND（または NOR）である頂点 v のラベルは 1(0) となる．ここで，v が T の根ならば（次数が 2 なので）*correct* である．v の親頂点 v_p が存在するなら，v_p の関数ラベルは NOR (NAND) となるので v_p の親の色は 0(1) となることは容易に分かる．このことから v は *correct* となる．

次に，子頂点のラベルが全て 1（または全て 0）の頂点 v は，関数ラベルが NAND, NOR のいずれであっても

v のラベルは 0(1) となり 3 近傍のうち 2 近傍の色が異なるようにラベル付されているので *correct* である． □

[補題 10] 彩色 CL に対し，F に含まれる葉，根以外の全頂点のラベルは *correct* である．

(証明) F に含まれる葉，根以外の全頂点は，F の頂点集合 E_f に近傍として 2 頂点を含んでいること，及び E_f への彩色の方法より明らかである． □

現時点での彩色では F に含まれる根，葉頂点は，F に含まれない 2 頂点を近傍として持つので，*correct* であるとは限らないことに注意がされたい．ラベルが *incorrect* である頂点に関する処理は次の (iv) で説明する．ラベルが *incorrect* である頂点 u の性質として，(a) 補題 1,2 より，u は全域森 F の根，葉頂点に限られる．(b) u は F に含まれる頂点の近傍を一つ持ち，そのラベルは $\overline{l(u)}$ である．

(iv) ここで以下の議論を容易にするため，頂点 w に対し近傍のラベルの状況を表す関数 $f(w)$ を定義する．$f(w)$ は w の近傍でラベルが $\overline{l(w)}$ である頂点数から $\left\lceil \frac{d(w)}{2} \right\rceil$ を減じた値である．w のラベルが *correct* であるの $f(w) \geq 0$ の場合に限られることが分かる．次数が 3 以下であることから $f(w)$ は，$-2 \leq f(w) \leq 1$ を満たす．更に，w を F に含まれる頂点とすると $-1 \leq f(w) \leq 1$ である．

今，F のある木の根頂点 u のラベルが *incorrect* であるとする．このとき，その木に含まれる頂点で u に最も近い $f(w) = 1$ なる頂点 w を探す．いま，そのような頂点 w が見つかったとする．このとき，u と w 間にある頂点，及び u のラベルを反転する．見つからない場合は木に含まれる全頂点のラベルを反転させる．以上の操作の後，その木の葉頂点のラベルが *incorrect* なら，同様の操作をもう一度行なう．$f(v) = 1$ なる頂点 v，反転させる頂点の決定は $O(\log n)$ 時間で可能である．また，ラベルが *correct* であった頂点を *incorrect* にすることもないことは容易に確かめられる． □

4 発見的並列アルゴリズム

本アルゴリズムの基本的考え方は，各頂点のラベルが *correct* であるかを調べ，*correct* でないならある確率で反転させることである．ここで *correct* でない全頂点を確率 1 で常に反転させたのではうまくいかないことは，例えば，直線上のグラフで全ての頂点のラベルが 0 でラベル付されている場合等を考えれば容易に分かる．

Algorithm
while (ラベルが *incorrect* である頂点が存在する) **do**
  call *SETPROB*;
  各頂点 v_i のラベルを確率 $p(v_i)$ で反転;
end (while).

ここで手続き *SETPROB* はラベルが *incorrect* である頂点 v の反転確率 $p(v)$ を並列に決める手続きである．ラベルが *correct* である頂点の反転確率は 0 である．*SETPROB* に関しては，いくつかの方法が考えられるが，ここでは今のところ最も良い結果を示している方法を述べる．

先ず，ラベルが *incorrect* である頂点 v の反転確率 $p(v)$ を初期値 0.5 とする．v の近傍の各ラベルとその反転確率から判断して，v が近傍のラベルの反転に対して *correct* であるためには，v のラベルを反転すべきかどうかを判断する．反転すべきであれば $p(v)$ の値をある値 x だけ大きくし，そうでなければ x だけ小さくする．この操作を繰り返し適用することにより，$p(v)$ の値を同定してゆく．但し，最終的に $p(v)$ がある値に収束するように，x の値は初期値 0.25 から始めて $0.125, 0.0625, \cdots$ と半減させてゆく．アルゴリズムをより詳しく記述すると次のようになる．

procedure *SETPROB*
  $x := 0.25$; (* 確率調整変数 *)
  for $i := 1$ to n **in parallel do**
    $p_{old}(v_i) := p(v_i)$;
    if $l(v_i)$ is *incorrect* **then**
      $p(v_i) := 0.5$
    else
      $p(v_i) := 0$;
  end (for)
  while $(x > \varepsilon)$ **do**
    if $check(v_i) = true$ **then**
      $p(v_i) := p(v_i) + x$
    else

```
        p(v_i) := p(v_i) - x;
    x := x/2;
  end (while)
end (procedure)

function check(v_i)
  sum_i := 0;
  for j := 1 to d(v_i) in parallel do
    Let w_j ∈ N(v);
    if \overline{l(v_i)} ≠ l(w_j) then
      if l(w_j) is correct then
        sum_i := sum_i + 1
      else
        sum_i := sum_i + (1 - p_old(w_j));
    else if l(w_j) is incorrect then
      sum_i := sum_i + p_old(w_j);
    end (if)
  end (for)
  if sum_i ≥ ⌈d(v)/2⌉ then
    return true
  else
    return false;
end (function)
```

上記アルゴリズムの実行時間の評価は以下のようになる．関数 *check* の実行時間は，$O(\log n)$ であることは容易に分かる．また，手続き *SETPROB* は明らかに関数 *check* を定数回呼ぶにすぎないので，手続き *SETPROB* の実行時間は，$O(\log n)$ である．問題は，手続き *SETPROB* の呼び出し回数であるが，今のところ解析的な方法は分かっていない．そこで，ランダムなグラフに対して計算機によるシミュレーションを行ないその結果で評価する．その結果を以下に示す．試行回数はいずれも 500 回である．

表 1．次数が定数（10 程度）のとき

頂点数	50	100	200	400	800	1600
平均実行回数	4.9	5.5	6.5	7.3	8.2	9.3

表 2．枝の存在確率を 0.5 とした場合

頂点数	50	100	200	400	800	1600
平均実行回数	6.5	9.3	14.5	22.1	34.1	53.1

表 1 に示すように，次数を定数に保って頂点数を倍化させた場合，ほぼ線形に実行回数が増加しているようにみえる．このことから，次数が定数のグラフに対しては，本アルゴリズムは確率 NC アルゴリズムになっているのかもしれないと考えている．

5 今後の課題

本稿では，DTML 問題の並列化可能性を探るため，決定性グラフ DTML 問題は P 完全であること，及び，次数が 3 以下のグラフに対して DTML 問題は NC に含まれることを示した．一般のグラフに対する DTML 問題に関しては，未解決の問題として残されたが，その解決へ向けてのヒントを提供した．次数 3 の NC アルゴリズムを定数次数のグラフに対して拡張することは容易ではないが，定数次数の DTML 問題には NC アルゴリズムが存在するのではないかと予想している．

参考文献

[1] 有川，宮野，“オートマトンと計算可能性”, 培風館（昭 61）．

[2] A. Gibbons, W. Rytter, “Efficient parallel algorithms,” *Cambridge University Press*, 1988.

[3] M. Goldberg and T. Spencer, “Constructing a maximal independent set in parallel,” *SIAM J. Disc. Math. Vol.2, No.3*, pp. 322–328, 1989.

[4] M. Luby, “A Simple Parallel algorithm for the maximal independent set problem,” *SIAM J.Comput. Vol.15, No.4*, pp.1036–1053, 1986.

[5] M. Luby, “Removing randomness in parallel computation without a processor penalty,” *Proc. 29th IEEE FOCS*, pp.162–173, 1988.

[6] S. Miyano, S. Shiraishi, and T. Shoudai, “A list of P-complete problems,” *Technical Report RIFIS-TR-CS-17*, 1990.

[7] R. Sarnath, and Xin He, “A P-complete graph partition problem,” *Theoretical Computer Science* 76, pp.343–351, 1990.

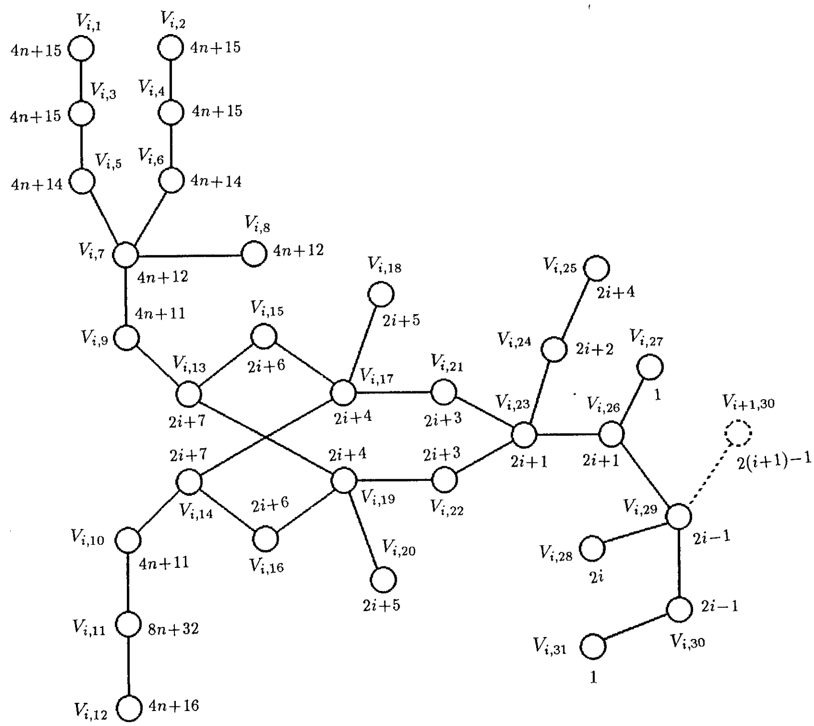
$V_{i,1}$
$4n+15$
$V_{i,2}$
$4n+15$
$V_{i,3}$
$4n+15$
$V_{i,4}$
$4n+15$
$V_{i,5}$
$4n+14$
$V_{i,6}$
$4n+14$
$V_{i,7}$
$4n+12$
$V_{i,8}$
$4n+12$
$V_{i,9}$
$4n+11$
$V_{i,10}$
$4n+11$
$V_{i,11}$
$8n+32$
$V_{i,12}$
$4n+16$
$V_{i,13}$
$2i+7$
$V_{i,14}$
$2i+7$
$V_{i,15}$
$2i+6$
$V_{i,16}$
$2i+6$
$V_{i,17}$
$2i+4$
$V_{i,18}$
$2i+5$
$V_{i,19}$
$2i+4$
$V_{i,20}$
$2i+5$
$V_{i,21}$
$2i+3$
$V_{i,22}$
$2i+3$
$V_{i,23}$
$2i+1$
$V_{i,24}$
$2i+2$
$V_{i,25}$
$2i+4$
$V_{i,26}$
$2i+1$
$V_{i,27}$
1
$V_{i,28}$
$2i$
$V_{i,29}$
$2i-1$
$V_{i,30}$
$2i-1$
$V_{i,31}$
1
$V_{i+1,30}$
$2(i+1)-1$

Fig.1