数理解析研究所講究録
1185 巻 2001 年 140-149

グラフの構造的特徴と効率の良い並列アルゴリズムについて

Some structural features of graphs which make efficient parallel algorithms possible

増山 繁 [1]　　中山 慎一 [2]
Shigeru MASUYAMA　　Shin-ichi NAKAYAMA

[1] 豊橋技術科学大学知識情報工学系
〒 441-8580 豊橋市天伯町雲雀ヶ丘 1-1
masuyama@tutkie.tut.ac.jp

[2] 徳島大学総合科学部自然システム学科数理科学
〒７７０ 徳島市南常三島町一丁目一番地
shin@ias.tokushima-u.ac.jp

摘要: 本稿では，グラフがどのような構造的特徴を持てば，グラフ理論のどのような問題に対し効率の良い並列アルゴリズムが存在するのかを，台形グラフ，イントーナメントグラフ，外平面グラフ，2 連結グラフを例にとって考察する．

キーワード (Keywords): 並列グラフアルゴリズム (parallel graph algorithms)，構造と計算量 (structural features and complexity)，台形グラフ (trapezoid graph)，イントーナメントグラフ (in-tournament)，外平面グラフ (outerplanar graph)，2 連結グラフ (bi-connected graph)，最小彩色 (minimum coloring)，ハミルトン路 (Hamiltonian path)，ハミルトン閉路 (Hamiltonian cycle)，最短路 (shortest path)，最大流 (maximum flow)，中心化問題 (centering)

1　はじめに

並列アルゴリズム (parallel algorithm) は, 複数個のプロセッサを持つ並列計算機上で実行されるアルゴリズムである. 現在, 分割統治法 (divide and conquer) の適用が可能である場合, 即ち, 問題が, それぞれ局所的に解くことのできる部分問題に分解でき, かつ, それぞれを解いた結果を独立に統合していくことで元の問題の解が構成できる場合に効率の良い並列アルゴリズムが構成できることが知られている. しかしながら, 並列処理を用いて個々の現実の問題を効率的に解決するのに十分精密な知見が得られているとはいえない. 従って, 現実の問題が固有に持つ並列性を見究めることにより, 並列処理に適した問題の構造的特徴を理論的に厳密に解明していく必要がある. 特にグラフ上の問題では, 逐次アルゴリズムの場合と同様に, グラフがある構造を持っている時, 一般の場合より効率の良い並列アルゴリズムがある場合が少なからずある.

そこで本稿では, グラフがどのような構造的特徴を持てば, どのような問題に対して効率の良い並列アルゴリズムを作ることができるのかの考察を試みる. その一つの手がかりとして, 我々がこれまでに行なってきた並列グラフアルゴリズム (parallel graph algorithm) の研究の中から, 台形グラフ上の最小彩色問題, 及び, イントーナメントグラフ上のハミルトン路, ハミルトン閉路問題, 外平面グラフ上の最大流問題, 2 連結グラフ上で与えられた節点が中心となるように全域木を構成する問題 (centering) を主に取り上げ, 並列計算量とグラフの構造的特徴との関係の観点から振り返る.

並列計算モデルとしては $PRAM$(parallel random access machine) を取り上げる. これは, 共有メモリを持つ $SIMD$ 型の並列計算モデルであり, 共有メモリへのアクセス制御に関して $CRCW$, $CREW$, $ERCW$, $EREW$ の 4 種に分類される. C は同時にアクセス可, E は高々一つのプロセッサのみアクセス可, R は読み込み, W は書き込みを表す. $PRAM$ は並列アーキテクチャ, 即ち, プロセッサ間の結合形態によらない問題固有の並列性を浮き彫りにするのに有用なので, 理論的研究におい

表 1: 区間グラフ，置換グラフ，台形グラフ，円弧グラフ上の並列アルゴリズム (その 1)

	interval	permutation	trapezoid
(unweighted) minimum dominating set	interval $O(\log\log n)$ $O(n/\log\log n)$ Common CRCW [51] circular-arc $O(\log n)$ $O(n/\log n)$ EREW [48]		
independent dominating set			$O(\log^2 n)$ $O(\max\{n^3/\log^2 n, n^{2.376}\})$ EREW [31]
minimum total dominating set minimum connected dominating set	$O(\log\log n)$ $O(n/\log\log n)$ Common CRCW [51]		$O(\log^2 n)$ $O(\max\{nm^2/\log^2 n, m^{2.376}\})$ EREW [31]
coloring	$O(\frac{n}{p}+\log n)$ p EREW [45]	$O(\log^2 n)$ $O(n^3/\log n)$ CREW [53]	$O(\log^2 n)$ $O(n^3/\log n)$ CREW [42]
shortest paths	a pair (1)$O(\log n)$ $O(n/\log n)$ CREW [13] (2)$O(n/p+\log n)$ p EREW [46]	all-to-all $O(\log n)$ $O(n^2/\log n)$ CREW [14]	

て好んで用いられる並列計算モデルである．通常，*PRAM* 上の並列アルゴリズムにおいては，入力サイズの多項式個のプロセッサを用いて log(入力サイズ) の多項式時間で計算される並列アルゴリズムを効率の良い並列アルゴリズムとみなし，そのようなアルゴリズムが存在する問題のクラスをクラス *NC* と呼ぶ．並列アルゴリズムの教科書，参考書として [19, 27, 36, 47, 49] などがある．ハンドブック [28] も並列アルゴリズムに関して比較的詳しい解説を載せている．

以下，本稿では，特に有向グラフ (directed graph, digraph と略す) であると断らない限り，グラフ (graph) は，無向グラフ (undirected graph) を指すものとする．グラフ $G=(V,E)$，V は節点 (vertex) の集合，E は辺 (edge) の集合，に対して，任意の節点間に路 (path) があるとき，グラフ G は連結グラフ (connected graph) という．また，連結な極大部分グラフを連結成分 (connected component) という．閉路を持たない連結グラフを木 (tree) という．グラフ $G=(V,E)$ の任意の 2 節点 $u, v \in V_i$ に対し，$(u,v)\in E$ であるか，u と v を通る単純閉路がその生成部分グラフ (V_i と，両端が V_i に属する辺のみからなる G の部分グラフ) 内に存在するような極大な節点集合 V_i を G の 2 連

表 2: 区間グラフ，置換グラフ，台形グラフ，円弧グラフ上の並列アルゴリズム (その 2)

	interval	permutation	trapezoid
depth-first search (DFS)	$O(\log\log n)$ $O(n/\log\log n)$ Common CRCW [51]		
breadth-first search (BFS)	$O(\log\log n)$ $O(n/\log\log n)$ Common CRCW [51]	$O(\log n)$ $O(n/\log n)$ EREW [12]	
(unweighted) maximum independent set	interval $O(\log\log n)$ $O(n/\log\log n)$ Common CRCW [51] circular-arc $O(\log n)$ $O(n/\log n)$ EREW [48]		
weighted independent set	$O(\log^2 n)$ $O(n^3/\log n)$ EREW [1]	$O(\log^2 n)$ $O(n^3/\log n)$ CREW [53][54]	
maximum weight clique	$O(\log n)$ n EREW [50]	$O(\log^2 n)$ $O(n^3/\log n)$ CREW [53]	
minimum cliques cover problem	circular-arc $O(\log n)$ $O(n/\log n)$ EREW [48]	$O(\log^2 n)$ $O(n^3/\log n)$ CREW [53]	
Hamiltonian circuit	$O(\log n)$ $O(n^2/\log n)$ EREW [1]		

結成分 (bi-connected component) という [24, 25]. 一つの 2 連結成分のみからなるグラフを 2 連結グラフ (bi-connected graph) という. グラフ理論の用語の詳細は, [5, 23, 26, 7] 等を参照されたい.

2 台形グラフ上の並列アルゴリズム

台形グラフ (trapezoid graph) は, 平面上で 2 本の平行線上にそれぞれ上辺, 下辺を持つ n 個の台形 (trapezoid) が与えられたとき, それぞれの台形に一つずつ節点を対応させ, 対応する台形が共通部分を持つ場合に節点間を辺で結ぶことによって得ることができるグラフである. 台形グラフに対し, そのような 2 本の平行線 (以下, 基準平行線と呼ぶ) と n 個の台形 (n は節点数) を G に対応する台形図式 (trapezoid diagram) という. 台形グラフは, Dagan[11] によって始めて導入された. たとえば, 台形グラフ上の彩色問題は, VLSI 設計において平面上の 2 本の平行線の間に配線可能領域がある時, 台形状の多重配線領域をお互いに重ならないようにするために必要な層 (ウエファー) の数を最小化する問題に対応する等の応用がある [11].

台形グラフは各台形の上底, 下底のいずれも長さが 0 とすると 2 つの平行直線を結ぶ線分となることから, 置換グラフ (permutation graph) の一般化である. また, 上底と下底の左端と右端の x 座標がそれぞれ一致するとき, 2 つの台形が交わるかどうかが, 2 本の平行線を重ねた時に, それぞれの台形に対応する線分が交わるかどうかに帰着できることから, 台形グラフは区間グラフ (interval graph) の一般化である. これらのグラフに関する効率の良いアルゴリズムのある問題に対して, 台形グラフまで拡張できる可能性がある.

台形グラフの顕著な構造的特徴は, co-comparability[11] である. グラフ G が co-comparable とは, G の補グラフ (complementary graph) 上において, 反対称的で, かつ, 遷移的な向き付け (orientation)F を持つことである [11]. この構造的特徴により, 台形グラフはその補グラフ上で対応する問題が路に関する問題となる場合に対し効率の良い並列アルゴリズムを持つ可能性がある. たとえば, 文献 [43] では, 台形グラフに対して, *CREW PRAM* 上で $O(n^3)$ 個のプロセッサを用いて $O(log^2\ n)$ 時間で最小重み連結支配集合 (minimum connected dominationg set) 問題を解く並列アルゴリズムを提案した. そこでは, ソースペアである辺からシンクペアである辺への単純な路がいずれも連結支配集合であるという事実を用いて, その中の最短路を求めることで最小連結支配集合を求めている. 但し, 辺 (u, v) が節点 1 から $min\{u, v\}$ ($max\{u, v\}$) までの節点をすべて支配する時 (u, v) をソースペア (シンクペア) という.

文献 [42] では, 台形グラフに対して, *CREW PRAM* 上で $O(n^3/log\ n)$ 個のプロセッサを用いて $O(log^2\ n)$ 時間でグラフの最小彩色 (minimum coloring) 問題を解く並列アルゴリズムを提案した. これは, グラフの最小彩色問題が, 補グラフ上では完全部分グラフ (clique) への分割問題になることを用いている. 特に台形グラフの補グラフでは完全部分グラフが 1 本の路に対応する. そこで, 最小彩色問題が, 補グラフ上では最小パスカバー (minimum path cover) 問題に帰着される. 以下, 一般性を失うことなく, 2 本の基準平行線は水平, すなわち, x 軸に並行であるとし, 更に, 台形の端点の x 座標は全て異なるとする. 補グラフで最小本数の路によるすべての節点のカバーを求める最小パスカバー問題を解くため, 台形図式で左上-左下の辺に対応する点を赤点, 右上-右下の辺に対応する点を白点として平面上に描く. なお, それぞれの点の座標を, x 座標はハイフンの左の点の台形図式上の基準平行線上での位置 (元の平面上での x 座標), y 座標はハイフンの右の点の台形図式上の基準平行線上での位置 (元の平面上での x 座標) とする. すると,T_i が T_j と交わらないための必要十分条件は, 赤点 $r_j = (x_j, y_j)$ が白点 $w_i = (x_i, y_i)$ を支配する (dominate) こと, すなわち, $x_i < x_j$, かつ, $y_i < y_j$ が成り立つことであることが容易に示せる. そこで,r が w を支配している時 $r \in R$ と $w \in W$ がマッチできるとして最大マッチング (maximum matching) を求める. それにより, 最大パスカバーを求めれば良い.

この節を閉じるに当たり, 区間グラフ, 置換グラフ, 台形グラフ計算量の観点から見た時の相違点について考察する.

先に述べたように, 台形グラフは区間グラフ, 置換グラフを共に含む上位クラスである. 区間グラフ, 置換グラフがダイアグラム上で線の交差により対応するグラフの節点間に辺を接続するのに対し,

台形グラフの場合は，ダイアグラム上で面の交差に関して対応する節点間の辺が決まる．つまり，複数の線の交差関係を調べるのと複数の面の交差関係を調べるのを比較した場合，面の交差関係を調べる方が組み合わせ数が増え，計算量が増えるが，それと同様に同じ問題であっても区間グラフ，置換グラフに比べ台形グラフ上では組み合わせの数が増えるため問題を解くのが難しくなることがある．従って，区間グラフ，置換グラフの方が台形グラフの場合よりも計算量が下げられる場合がある．表1，2に，区間グラフ，置換グラフ，台形グラフ，及び，区間グラフの直線の代わりに円周，区間の代わりに円弧として得られる円弧グラフ上でのこれまでに知られている並列アルゴリズムを示す．

3 イントーナメントグラフ上の並列アルゴリズム

トーナメントグラフ (tournament) は, 完全グラフ (complete graph) の各辺に任意に向きを与えることで得られる有向グラフ (digraph) で, その上ではハミルトン路 (Hamiltonian path) 問題等, 多くの路に関する問題が扱いやすい．

例えば，路に関する問題などでは分割統治法的なアイデアで各局所部分について並列に路を構成し，それら局所的な路同士を繋げて全体の解を構成出来ると並列化可能となる．トーナメントグラフや，本節で紹介する，その一般化であるイントーナメントグラフでは，辺が密に存在するという特徴を利用し，一定の規則に従って局所的な路同士を繋ぐ事ができる場合がある．この事からトーナメントグラフ，イントーナメントグラフ上で，路に関する問題に対し，効率の良い並列アルゴリズムが得られる可能性がある．

文献 [44] では，トーナメントグラフの拡張であるイントーナメントグラフ (in-tournament) 上でハミルトン路, 及び, ハミルトン閉路 (Hamiltonian cycle) の存在判定と構成を行なう NC アルゴリズムを得た．イントーナメントグラフとは、辺 (x,z), (y,z) が存在すれば, 必ず辺 (x,y), (y,x) のいずれか一方, かつ一方のみが存在する有向グラフである．次の補題に基づいてハミルトン路を構成する並列アルゴリズムが構築される．このことにより，イントーナメントグラフは，上記のトーナメントグラフの，効率の良い解を与える特徴を保存しつつ一般化しているといえる．

[補題 5.1] 連結イントーナメントグラフ D がハミルトン路を持つための必要十分条件は D がインブランチング (in-branching) を持つことである．但し, インブランチングとは根 r を除くすべての節点の出次数が 1 で, 閉路を持たず, 根 r 以外の節点から r に到達可能な有向グラフである．

アルゴリズムの概略は以下の通りである．

1. 各強連結成分をそれぞれ節点として非巡回グラフを作る．

2. 出次数 0 の節点を根 r とし, 各枝の向きを逆転して幅優先探索を行なうことでインブランチングを得る．

3. インブランチングをそれぞれ合流する 2 つの路からなる部分グラフ $D_1, D_2, \ldots, D_s$ に分割する．

4. 各 D_i に対し並列にハミルトン路 H_i を求める．

5. $H_1, H_2, \ldots, H_s$ を併合して 1 本の路を構成する．

ハミルトン閉路の構成については, 先に構成したハミルトン路 $x_1, x_2, \cdots, x_n$ をもとに, ハミルトン路に使われていない辺の中で, 後方辺 (x_j, x_i), $x_i < x_j$ という辺に着目している．これら後方辺の組と路 $x_1, x_2, \cdots, x_n$ の部分を並列に置き換える事により, x_n から x_1 に戻る路を構成し, 最終的にハミルトン閉路 $x_1, \cdots, x_n, \cdots, x_1$ を構成している．

4 外平面グラフ上の並列アルゴリズム

外平面グラフ (outerplanar graph) は, 直並列グラフ (series parallel graph) の部分クラスであるが, その特殊な構造を活用して, 効率の良い並列アルゴリズムが構築できる可能性がある．外平面グラフは, 各 2 連結成分の外周 (marginal cycle) を表す閉路 (cycle) を, 対角線 (diagonal) がお互いに交わらないように平面に描くことができるグラフのことである．外平面グラフであるための必要十分条件は，節点をうまく直線上に並べると，平面上で全ての辺をお互いに交差しないように直線の片側の領域に描くことができることである [23]．この性質より，外平面グラフは，たとえば，平面上で直線の

片側が配線領域である場合に，配線間が交わらないように端子間を結ぶ問題とみなせるなど，VLSI 設計等に応用がある [52]．また，この条件は，自然言語処理における係り受け関係の非交差性に対応する [34]．

各 2 連結成分 (bi-connencted component) に対して面 (face, 一つの基本閉路で区切られた平面の閉領域) を節点, 面が辺を共有する時それらの面に対応する節点間を辺で結ぶと木となる．そこで, それぞれの面毎に局所的に解を構成し, 上述した面間の木構造を利用して, まず, 各 2 連結成分に対して解を構成する．更に, 2 連結成分間の関係も木で表されることが知られているので, それを利用することで全体の解を構成していくことができる．つまり, 分割統治法の適用が可能である．

直並列グラフの場合は 2 端子が指定されている．例えば, 最長路問題, 最大流問題等, 始点, 終点をその 2 端子に指定した場合のみ一般のグラフより効率の良いアルゴリズムが知られている．それに対し, 外平面グラフでは, これらの問題に対し自由に始点, 終点をとれるという利点がある．

外平面グラフの各 2 連結成分の外周は, *st* 番号付け (*st* numbering) により付された番号の順に辿っていくことで求まる．*st* 番号付けとは次の条件 (1), (2) を満たす節点集合 V から整数の集合 $\{1, 2, \ldots, n\}$ への 1 対 1 写像のことである [33]．本稿では，n は与えられたグラフの節点数を表す．

(1) $f(s) = 1$, $f(t) = n$.

(2) 各節点 $v \in V - \{s, t\}$ において, v は $f(v_1) < f(v) < f(v_2)$ であるような節点 v_1, v_2 に隣接する．

なお，*st* 番号付けは *ARBITRARY CRCW PRAM* 上で $O(n\alpha(m,n)/log\ n)$ 個のプロセッサを用いて $O(log\ n)$ 時間で実行可能である [18]．但し，本稿では $\alpha(m,n)$ はアッカーマン関数 (Ackermann function) の逆関数を表す．この関数は, m, n に関して非常にゆっくりと増加するので，ほとんど定数であるとみなすことができる．

文献 [38] では, 辺の長さが非負である外平面グラフ G の任意に与えられた相異なる 2 節点 s, t 間の最短路 (shortest path) を *CRCW PRAM* 上で $O(n\ loglog\ n/log\ n)$ 個のプロセッサを用いて $O(log\ n)$ 時間で求める並列アルゴリズムを提案した．

外平面グラフが 2 連結グラフであるとして最短路を求める並列アルゴリズムの概略を述べる．それを求めることができると，与えられたグラフを 2 連結成分に分解し，次に，2 連結成分間の木構造を用いて，容易に最短路を求めることができる [38]．

2 連結グラフの場合のアルゴリズムの概要は以下の通りである．

1. *st* 番号付けにより G の外周を求める．外周はハミルトン閉路 C をなす．なお，ここで,「st 番号付け」は，ある次数 2 の節点を番号 1 としており，それは必ずしも最短路の始点 s とは一致しないことに注意．おなじく，番号 n の点も最短路の終点 t とは必ずしも一致しない．

2. 節点番号 j の，最短路の始点 s を節点番号 1 とするために，節点番号 $1, 2, \ldots, n$ の節点からなる循環リストを作り，全ての節点番号をリストに沿って $n - j + 1$ だけ移動させる．

3. G の外周上の各節点間の距離を求めるために s から j への距離を表す配列 $LEN(j), j = 1, \ldots, n$ を求める．これは，辺 $\{j-1, j\}$ の長さを格納している配列 $L(j), j = 1, \ldots, n$, からプレフィックス演算 (例えば，[27] 参照) で求めることができる．

4. 節点番号に基づき，以下のようにそれぞれが面に対応する有向閉路 $C_1, \ldots, C_k$ を構成する:

C の辺に i から $i+1$, $i = 1, \ldots, n$, 但し，$n+1 = 1$, となるように向き付け，更に，外周 C の各対角線 $\{i, j\}$ に対し，有向辺 (i, j) と (j, i) を付加することで G から有向グラフ $G' = (V', E')$ を構成する．$C_1, \ldots, C_k$ は G' の辺からなる，C の面と 1 対 1 に対応する有向閉路である．

5. 有向閉路 $C_1, \ldots, C_k$ を節点とする木 T_G を構成する．

6. T_G 上で s を含む閉路に対応する節点と t を含む閉路に対応する節点とを結ぶ主線とそれを除去して得られる部分木である支線 $T^*(i)\ i = 1, \ldots, k$ を求める．

7. $T^*(i)$ に対応する G の部分グラフ $G^*(i)$ と，主線上の節点に対応する G の閉路に共有される節点 $a_i b_i$ 間の最短距離を求める．その最短距離の計算に木の縮約 (例えば，[27]) が用いられる．

8. G^* の外周上の各節点間の距離を求めるために配列 $LEN'(j)\ j = 1, \ldots, n$ を求める．ここで G^* は，G 上で s と t を結ぶ外周上の路 P_1, P_2 と．それらを結ぶ対角線からなる．$LEN'(j)$ の計算法は $LEN(j)$ と同様である．

9. G_i^* に対応する最短路から G^* における最短路を求める．その結果，G の s, t 最短路を得る．

文献[37]では，外平面グラフ G の任意に与えられた相異なる2節点 s, t 間の最長路 (longest path) を $CRCW\ PRAM$ 上で $O(n^3/log^2\ n + M(n))$ 個のプロセッサを用いて $O(log\ n)$ 時間で求める並列アルゴリズムを提案した．本稿では，$M(n)$ は2つの $n \times n$ 行列の積を $O(log\ n)$ 時間で実行するのに必要なプロセッサ数を表す．なお，$M(n) = O(n^{2.396})$ 個で計算できる [28]．簡単のため外平面グラフ G は2連結とする．すると，外周である周辺とその上の節点をお互いに交わらない交差弦が結ぶ形にできる．そこで，周辺上で s と t を結ぶ点素 (vertex disjoint) な路が丁度2つできる．次に，G と st 最長路長が等しくなる非サイクル有向グラフ (acyclic digraph)G' を構成する．更に，非サイクル有向グラフ G' に対する最長路を求める並列アルゴリズム [28] を適用すれば良い．

文献 [39] では，外平面グラフ G における最大流 (maximum flow) 問題を $CREW\ PRAM$ 上で $O(n^2/log\ n)$ 個のプロセッサを用いて $O(log\ n)$ 時間で解く並列アルゴリズムを提案した．

外平面グラフが2連結グラフであるとして最大流を求める並列アルゴリズムの概略を述べる．それが求まると，2連結成分間の木構造を用いて，容易に全体の流量が構成できる [39]．

まず，以下の手順で，s, t 間の流量を求める．

1. st 番号付けにより G の外周を求め，始点 s を節点番号1とし，外周に沿って G の各節点に番号を付ける．対角線がないなら6へ．

2. G に対し外周上では節点番号の小さい方から大きい方へ辺の向き付けをする．また，対角線に対しては2本の逆方向の有向辺を対応させて得られる有向グラフを G' とする．G' は各節点で入次数と出次数が等しいのでオイラーグラフである．節点番号に基づき G' における有向閉路 $C_1, \ldots, C_k$ を構成する．

3. $C_1, \ldots, C_k$ を節点とする根付木 T_G を構成する．

4. T_G 上で s を含む閉路に対応する節点と t を含む閉路に対応する節点とを結ぶ主線とそれを除去して得られる部分木である支線を求める．木 T_G の主線に対応する面からなる G の部分グラフを $\tilde{G}$ とする．支線がなければ6へ．

5. 支線にあたる各部分木を $T_1, \ldots, T_k$ とする．

for all $i, 1 \leq i \leq k$ **in parallel do**

T_i に対応する G の部分グラフ G_i と $\tilde{G}$ との共有節点 x_i-y_i 間の最大流量を求める．

6. $\tilde{G}$ の修正双対グラフ $\tilde{G}^*$ を求める．(最大流問題が双対グラフ上では最短路問題になることを用いる．)

7. $\tilde{G}^*$ の各辺 (i^*, j^*) の長さ $\ell(i^*, j^*)$ を，対応する $\tilde{G}$ の辺 (i, j) の容量 $c(i, j)$ とし，$\tilde{G}^*$ における節点 v_{up}^* から v_{down}^* への最短距離を求める (この最短距離が G の最小カットの値であり，G の最大流量に等しい)．

なお，グラフが外平面グラフであるかどうかの判定は，st 番号付けを用いることで $CRCW\ PRAM$ 上で $O(n\alpha(\ell, n)/log\ n)$ 個のプロセッサを用いて $O(log\ n)$ 時間で実行できる [41]．但し，$\alpha(\ell, n)$ はアッカーマン関数の逆関数である．このアルゴリズムはプロセッサ数と時間計算量の積がほぼ線形，すなわち $O(n\alpha(\ell, n))$ であるという意味で，ほぼ最適な並列アルゴリズムである．文献 [15] と文献 [18, 28] のアルゴリズムを組み合わせることでも外平面グラフかどうかを判定できるが，文献 [41] のアルゴリズムの方が簡潔である．

5 2連結グラフ上の並列アルゴリズム

連結グラフ $G = (V, E)$ が2連結であるための必要十分条件が，G の節点が st 番号付けを持つことである [33] ことは，以下に例示するように，2連結グラフ上で効率の良い並列アルゴリズムを作る上で有用である．

文献 [40] では，2 連結グラフ (bi-connencted graph) において与えられた節点 r を中心 (center, r から最も遠い節点が最小となる節点) とする全域木を求める中心化問題 (centering) を解く $O(M(n))$ 個のプロセッサ，$O(log^2 n)$ 時間の並列アルゴリズムを提案した．

アルゴリズムの概略は以下の通りである．

まず，r に隣接する任意の節点 u をとり，それを t として st 番号付けを行なう．次に，G から $\{r, u\}$ を除去したグラフで，各辺を st 番号付けの番号の小さい節点から大きい節点への有向辺で置き換え

て非サイクル有向グラフ $G' = (V, E')$ を作成する．r から各節点への最長路を文献 [28] の方法で並列に求め，r から u への最長路を P とする．また，各節点 $v \in V$ への最長路長を $\ell(r, v)$ で表す．P の両端点 u, r に有向辺 (u, r) を付加して得られる閉路を C とする．C 上で節点 r から $\lceil |C| \rceil /2$ 番目の辺 (a, b) に対し，$C - (a, b)$ 上での節点 r から節点 a への路をそれぞれ $P_a P_b$ とし，有向閉路 C に含まれない G の節点集合を $\bar{V}$ とする．

$\bar{V}$ の各節点 w に対し並列に

$\ell(r, w) \leq \ell(r, a)$ ならば節点 w に隣接する節点の中で $\ell(r, w) > \ell(r, x)$ を満たす x をひとつ選択し，辺 (x, w) を求める木の辺とする．

$\ell(r, w) > \ell(r, a)$ ならば節点 w に隣接する節点の中で $\ell(r, w) < \ell(r, x)$ を満たす x をひとつ選択し，辺 (x, w) を求める木の辺とする．

6 むすび

現在，並列アルゴリズムの設計は技術 (Technology) ではなく，技芸 (Art) の段階である．したがって，並列アルゴリズムをより容易に，組織的に設計できるよう理論体系をより精緻化していくことが，最大の課題である．

参考文献

[1] Bertossi, Alan A.; Bonuccelli, Maurizio A. Some parallel algorithms on interval graphs. Discrete Appl. Math. 16 (1987), no. 2, 101–111.

[2] Balayogan, V. B.; Pandu Rangan, C. Parallel algorithms on interval graphs. RAIRO Inform. Theor. Appl. 29 (1995), no. 6, 451–470.

[3] E. Bampis, Y. Manoussakis, I. Milis, NC algorithms for antidirected Hamiltonian paths and cycles in tournaments, J. Combin. Math. Combin. Comput. 21, pp.223–234, 1996.

[4] J. Bang-Jensen, J. Huang and E. Prisner, In-tournament digraphs, J. Combinatorial Theory, Series B., Vol.59, pp.267-287, 1993.

[5] C. Berge, Graphes et Hypergraphes, Dunod, 1970; 伊理正夫, 伊理由美, 岩坪秀一, 小林欣吾, 佐藤創, 星守訳, グラフの理論 I, II, III, 1976.

[6] F. Bernalt and P. C. Kainen, The book thickness of a graph, J. Combinatorial TheorySer B, Vol.27, 1979, pp.320-331, 1979.

[7] J. A. Bondy and Murty, Graph Theory with Applications, MacMillan, 1976; 立花俊一, 奈良知恵, 田澤新成, グラフ理論への入門, 共立, 1991.

[8] Y. Caspi, E. Dekel, Edge coloring series parallel graphs, J. Algorithms 18, no. 2, pp.296–321, 1995.

[9] F. R. K. Chung, F. T. Leighton and A. L. Rosenberg, Embedding graphs in books: A graph layout problem with applications to VLSI design, SIAM J. Algebraic Discrete Methods, Vol.8, pp.33-58, 1987.

[10] M. Chrobak, D. Eppstein, Planar orientations with low out-degree and compaction of adjacency matrices. Theoret. Comput. Sci. 86, no. 2, 243–266, 1991.

[11] I. Dagan, M. C. Golumbic and R. Y. Pinter, Trapezoid graphs and their coloring, Discrete Applied Mathematics, Vol.21, pp.35-46, 1988.

[12] Chao, H. S.; Hsu, F. R.; Lee, R. C. T. An optimal EREW parallel algorithm forcomputing breadth-first search trees on permutation graphs. Inform. Process. Lett. 61 (1997), no. 6,311–316.

[13] Chen, Danny Z.; Lee, D. T.; Sridhar, R.; Sekharan, Chandra N. Solving the all-pair shortest path query problem on interval and circular-arc graphs. Networks 31 (1998), no. 4, 249–257.

[14] Dahlhaus, Elias Optimal (parallel) algorithms for the all-to-all vertices distance

problem for certain graph classes. Graph-theoretic concepts in computer science (Wiesbaden-Naurod, 1992), 60–69, Lecture Notes in Comput. Sci., 657,

[15] K. Dicks, T. Hagerup, and W. Wrytter, Optimal parallel algorithms for the recognition and coloring outerplanar graphs, LNCS 379, Springer, 1989.

[16] K. S. Easwarakumar, C. Pandu. Rangan and G. A. Cheston, A linear algorithm for centering a spanning tree of a biconnected graph, Information Processing Letters, Vol.51, pp.121-124, 1994.

[17] S. Felsner, R. Muller, L. Wernisch, Trapezoid graphs and generalizations, geometry, and algorithms, Discrete Applied Mathematics, Vol.74, pp.13-32, 1997.

[18] D. Fussel, V. Ramachandran and R. Thurimala, Finding triconnencted components by local replacement, SIAM J. Compt., 22, pp.587-616, 1993.

[19] A. Gibbons, W. Rytter, *Efficient Parallel Algorithms*, Cambridge University Press, 1988.

[20] A. Gibbons, W. Rytter, Optimally edge-colouring outerplanar graphs is in NC. Theoret. Comput. Sci. 71, no. 3, 401–411, 1990.

[21] M. C. Golumbic, Algorithmic Graph Theory and Perfect Graphs, Academic Press, 1980.

[22] D. J. Haglin, M. J. Wolf, On convex subsets in tournaments, SIAM J. Discrete Math. 9, no. 1, pp.63–70, 1996.

[23] F. Harary, Graph Theory, Addison-Wesley, 1968; 池田貞雄訳, グラフ理論, 1971.

[24] 茨木俊秀, アルゴリズムとデータ構造, 昭晃堂, 1989.

[25] 茨木俊秀, C によるアルゴリズムとデータ構造, 昭晃堂, 1999.

[26] 伊理正夫, 藤重悟, 大山達夫, グラフ・ネットワーク・マトロイド, 産業図書, 1986.

[27] J. JáJá, *An Introduction to Parallel Algorithms*, Addison-Wesley, 1992.

[28] Jan van Leewen ed., *Handbook of Theoretical Computer Science Vol.A: Algorithms and Complexity*, Elsevier, 1990; 廣瀬健, 野崎昭弘, 小林孝次郎監訳, 『コンピュータ基礎理論ハンドブック I, アルゴリズムと複雑さ』, 丸善, 1994.

[29] P. N. Klein, Efficient parallel algorithms for chordal graphs, SIAM J. Comput. Vol.25, No.4, pp.797-827, 1996.

[30] C. Levcopoulos, A. Lingas, O. Petersson, W. Rytter, Optimal parallel algorithms for testing isomorphism of trees and outerplanar graphs. Foundations of software technology and theoretical computer science, pp.204–214, Lecture Notes in Comput. Sci., 472, Springer, Berlin, 1990.

[31] Y. D. Liang, Parallel algorithms for the domination problems in trapezoid graphs. Discrete Appl. Math. 74, no. 3, 241–249, 1997.

[32] A. Lingas, A. Proskurowski, On parallel complexity of the subgraph homeomorphism and the subgraph isomorphism problem for classes of planar graphs. Theoret. Comput. Sci. 68, no. 2, pp.155–173, 1989.

[33] Y. Maon, B. Schieber and U. Vishkin, Parallel ear decomposition search (EDS) and st-numbering ib graphs, Theoretical Computer Science, Vol.47, pp.277-298, 1986.

[34] S. Masuyama, S. Naito, Deciding whether graph G has page number one is in *NC*, Information Processing Letters, vol.44, pp.7-10, 1992.

[35] S. L. Mitchell, Linear algorithms to recognize outerplanar and maximal outerplanar graphs, Inf. Processing Lett., vol.9, pp.229-232, 1979.

[36] 宮野悟,『並列アルゴリズム』, 近代科学社, 1993.

[37] 中山慎一, 増山繁, 外平面グラフ上の最長路問題を解く並列アルゴリズム, 電子情報通信学会論文誌 DI,Vol. J78-D1, pp.563-568, 1995.

[38] 中山慎一, 増山繁, 外平面グラフ上の s-t 最短経路を求める 並列アルゴリズム, 電子情報通信学会論文誌 Vol. J78-DI, pp.867-877, 1995.

[39] 中山慎一, 増山繁, 外平面グラフ上の最大流を求める並列アルゴリズム, 電子情報通信学会論文誌 DI, No.5, pp.226-236 DI, No.5, 1996.

[40] 中山慎一, 増山繁, 2 連結グラフ上の与えられた節点を中心とする全域木を求める並列アルゴリズム, 電子情報通信学会論文誌 DI, No. 5, pp.299-302, 1996.

[41] Shin-ichi Nakayama, Shigeru Masuyama, A simple near optimal parallel algorithm for recognizing outerplanar graphs, J. Math. Tokushima University, 30, pp.71-80, 1996.

[42] Shin-ichi Nakayama, Shigeru Masuyama, A parallel algorithm for solving the coloring problem on trapezoid graphs, Information Processing Letters 62, pp.323-427, 1997.

[43] Shin-ichi Nakayama, Shigeru Masuyama, A parallel algorithm for finding a minimum-weight connected dominating set on trapezoid graphs, Mathematica Japonica, Vol. 45, No.1, pp.165-171, 1997.

[44] Shin-ichi Nakayama, Shigeru Masuyama, Parallel algorithms for finding a Hamiltonian path and a Hamiltonian cycle in an intounament graph IEICE Trans. Fundamentals, Vol.E81-A, No.5, pp.757-767, 1998.

[45] Pal, Madhumangal; Bhattacharjee, G. P. An optimal parallel algorithm to color an interval graph. Parallel Process. Lett. 6 (1996), no. 4, 439–449.

[46] Pal, Madhumangal; Bhattacharjee, G. P. An optimal parallel algorithm for all-pairs shortest paths on unweighted interval graphs. Nordic J. Comput. 4 (1997), no. 4, 342–356.

[47] E. P. Preparata 著, 上林弥彦, 岡部寿男, 浜口清治, 武永康彦編・訳, 『プレパラータ先生の超並列計算講義』, 共立出版, 1996.

[48] Rao, A. Srinivasa; Pandu Rangan, C. Optimal parallel algorithms on circular-arc graphs. Inform. Process. Lett. 33 (1989), no. 3, 147–156.

[49] J. H. Reif ed., Synthesis of Parallel algorithms, Morgan Kaufmann, 1993.

[50] Shyu, Chii Huah A parallel algorithm for finding a maximum weight clique of an interval graph. Parallel Comput. 13 (1990), no. 2, 253–256.

[51] Sridhar, R.; Chandrasekharan, N. Highly parallelizable problems on sorted intervals. Parallel Comput. 21 (1995), no. 3, 433–446.

[52] M. Yanakakis, Embedding planar graphs in four pages, J. Comput. System Sci., Vol.38, pp.36-67, 1989.

[53] Yu, Chang Wu; Chen, Gen Huey Parallel algorithms for permutation graphs. BIT 33 (1993), no. 3, 413–419.

[54] Yu, Ming Shing; Tseng, Lin Yu; Chang, Shoe Jane Sequential and parallel algorithms for the maximum-weight independent set problem on permutation graphs. Inform. Process. Lett. 46 (1993), no. 1, 7–11.

[55] X. Zhou, S. Nakano, T. Nishizeki, Edge-coloring Partial k-trees, J. of Algorithms, Vol.21, pp.598-617, 1996.