区間グラフにおける全関節節点を求める並列アルゴリズム

01506161 釧路高専　　　　本間 宏利　HONMA Hirotoshi
01603863 豊橋技術科学大学　増山　繁 MASUYAMA Shigeru

1. はじめに

$G=(V,E)$ を単純無向グラフとし，$G-u$ を節点集合 $V-u$ から誘導される G の部分グラフとする．また，$d_G(x,y)$ を G 上の2節点 x,y 間の最短経路の長さとする．Chang ら [1] は $d_{G-u}(x,y)>d_G(x,y)$ を満たすような二つの節点 $x,y\in V-u$ が存在する時，節点 u を関節節点 (*hinge vertex*) と定義した．本研究では区間グラフに対して，その全関節節点を求めるプロセッサ数 $O(n)$，計算時間 $O(log\ n)$ で実行可能な効率的な並列アルゴリズムを開発した．但し，n は G の節点数である．

2. 定義

$G=(V,E)$ を単純無向グラフとし，$G-u$ を節点集合 $V-u$ から誘導される G の部分グラフとする．また，$d_G(x,y)$ を G 上の2節点 x,y 間の最短経路の長さとする．この時，$d_{G-u}(x,y)>d_G(x,y)$ を満たすような二つの節点 $x,y\in V-u$ が存在するならば，節点 u を関節節点 (*hinge vertex*) と定義する [1]．

区間グラフの定義の前に区間ダイアグラム [3] を紹介する．区間ダイアグラムは区間と呼ばれる n 本の水平線分の集合で構成され，区間ダイアグラム中の各区間はその両端にそれぞれ左端点 a，右端点 b を所持する．また，各区間は端点を共有することはなく，全ての端点には最も左の端点から右方向へ昇順に $1\le i\le 2n$ なる端点番号 i が割り付けられる．更に各区間にはその右端点の端点番号の昇順に対応させて，$1\le i\le n$ なる区間番号 i が割り付けられる．以降，区間 i の左端点番号を a_i，右端点番号を b_i と表現する．このような特徴を持つ幾何学的図形を区間ダイアグラムと呼ぶ．

次に区間グラフを定義する．区間ダイアグラムの各区間が各節点に一対一対応し，区間ダイアグラムの区間 i と区間 j とが交差する場合のみに節点 i,j 間に辺が存在するようなグラフを区間グラフ I(*interval graph*) と定義する [3]．すなわち

$I=(V,E)$,
$V=\{i\mid i$ は区間番号 $\}$,
$E=\{(i,j)\mid$ 区間 i,j は区間ダイアグラム上で交差を持つ $\}$.

また，区間グラフの各節点には，区間ダイアグラム上の対応する各区間に割り当てられている区間番号がそのまま節点番号として割り付けられる．区間グラフ $I=(V,E)$ において，節点 $v\in V$ に隣接する全ての節点集合を $N(v)$ と定義する．すなわち，$N(v)=\{w\mid v,w\in V,(v,w)\in E\}$．また，節点 v に隣接する v より大きい節点の中で，最大の節点を $M(v)$，同様に2番目に大きい節点を $SM(v)$ と定義する．ただし，そのような節点が存在しない場合はそれぞれ $M(v)=v, SM(v)=v$ とする．更に，区間ダイアグラム上の区間 v において，$D(v)=\{i\mid b_{SM(v)}<i<b_{M(v)}\}$ なる端点番号 i の集合を判別集合と定義する．この判別集合によって区間グラフ上の全関節節点を効率的に導出することができる．図1と図2に区間数11から構成される区間ダイアグラム D とそれに対応する区間グラフ I の例を示す．また，表1に各節点 v における $M(i),SM(i),D(i)$ を示す．

3. 区間グラフにおける関節節点の性質

Theorem 1　グラフ $G=(V,E)$ において，節点 $u\in V$ が関節節点である必要十分条件は，節点 u は節点 x,y に隣接する唯一の節点であり，かつそれらは隣接しないような節点 x,y が存在することである．□

Lemma 1　節点 u が区間グラフ I の関節節点であれば，$d_{I-u}(x,y)>d_I(x,y)$ を満たすような二つの節点 $x,y(x<y)\in N(u)$ が存在するが，この時，節点 u は節点 x に対して $u=M(x)$ の関係が成立する．□

Lemma 2　u が区間グラフ I の関節節点であれば，$d_{I-u}(x,y)>d_I(x,y)$ を満たすような二つの節点 $x,y(x<y)\in N(u)$ が存在するが，この時，$b_{SM(x)}<a_y<b_{M(x)}$ すなわち $a_y\in D(x)$ の関係が成立する．□

Lemma 3　区間グラフ $I=(V,E)$ の節点 u が関節節点である必要十分条件は，$x<y, u=M(x)$ かつ $b_{SM(x)}<a_y<b_{M(x)}$ $(a_y\in D(x))$ の条件を満たす二つの節点 $x,y\in V$ が存在することである．□

Lemma 4　区間グラフ $I=(V,E)$ の二つの節点 $x,y\in V$ に対して，$x<y$ ならば $SM(x)\le SM(y)$ である．□

Lemma 5　グラフ $I=(V,E)$ を区間グラフとする．今，$x,y,z,u\in V$ なる節点において，$x<y<z$，$M(x)=M(y)=u$ の関係が成立していると仮定する．この時，$d_{I-u}(y,z)>d_I(y,z)$ ならば $d_{I-u}(x,z)>d_I(x,z)$ の関係が成立する．□

Lemma 6　グラフ $I=(V,E)$ を区間グラフとし，二つの節点 $x,y\in V(x<y)$ が存在すると仮定する．この時，$M(x)=M(y)$ もしくは $D(x)\cap D(y)=\phi$ のどちらか一方の関係が成立する．□

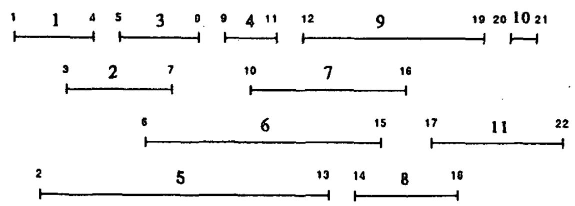

図 1: 区間ダイアグラム D

4. アルゴリズム

区間グラフ $I=(V,E)$ の全関節節点を求めるアルゴリズムを示す．このアルゴリズムは区間グラフ I に対応する区間ダイアグラム D 上における各区間の左端点番号 $a[1:n]$ と右端点番号 $b[1:n]$ を入力とする．

表 1: $M(i), SM(i), D(i)$

v	1	2	3	4	5	6	7	8	9	10	11
a	1	3	5	9	2	6	10	14	12	20	17
b	4	7	8	11	13	15	16	18	19	21	22
$P(i)$	1	2	2	2	2	6	10	12	12	17	17
$SP(i)$	2	3	5	6	6	10	12	14	17	20	
$M(i)$	5	6	6	7	9	9	9	11	11	11	11
$SM(i)$	2	5	5	6	7	8	8	9	9	10	11
$D(i)$	8,9,10,11,12	14	14	ϕ	17,18	ϕ	ϕ	20,21	20,21	ϕ	ϕ

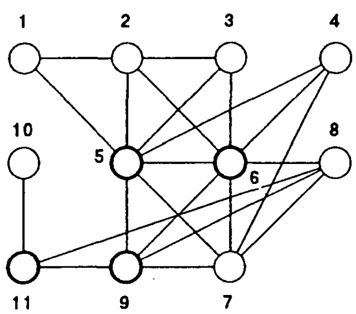

図 2: 区間グラフ I

Algorithm
Step 1 ($M[i]$ の算出)

配列 $P[1:n]$ を作成し，$1 \leq i \leq n$ なる i に対して並列に，$P[i] := min\{a_i, a_{i+1}, ..., a_n\}$ を計算する．

配列 $M[1:n]$ を作成し，$M[n] := n$ とする．

$1 \leq i \leq n-1$ なる i に対して並列に，$b_i < P[j]$ になるような最小の j を求め，$M[i] := j-1$ を計算する．ただし，$b_i > P[n]$ の場合は $M[i] := n$ とする．

Step 2 ($SM[i]$ の算出)

配列 $SP[1:n-1]$ を作成し，$1 \leq i \leq n-1$ なる i に対して並列に，$SP[i] := smin\{a_i, a_{i+1}, ..., a_n\}$ を計算する．ここで $smin\{\}$ は2番目に小さな要素を取り出す関数とする．

配列 $SM[1:n]$ を作成し，$SM[n] := n, SM[n-1] := n-1$ とする．

$1 \leq i \leq n-2$ なる i に対して並列に，$b_i < SP[j]$ になるような最小の j を求め，$SM[i] := j-1$ を計算する．ただし，$b_i > SP[n-1]$ の場合は $SM[i] := n-1$ とする．

Step 3 ($D[i]$ の算出)

異なる $M[i]$ の内容（複数の同じ $M[i]$ 値が存在する場合は最小の i を代表させる）を持つ i に対して並列に，$D[i] := \{k \mid b[SM[i]] < k < b[M[i]], k \in \mathbf{N}\}$ を計算する．

Step 4 (関節節点の算出)

Step 3と同様に，異なる $M[i]$ の内容（複数の同じ $M[i]$ 値が存在する場合は最小の i を代表させる）を持つ i に対して並列に，$a_z \in D[i]$ なる a_z が存在するならば $M[i]$ を関節節点とする．

End of Algorithm

アルゴリズムの解説と計算量の解析を行う．このアルゴリズムは補題3で述べている関節節点であるための必要十分条件を基に，区間グラフの全関節節点を並列に求めている．

Step 1は各区間 i において，交差を持つ最も区間番号の大きな区間 $M[i]$ を算出している．$M[i]$ の算出に配列 $P[1:n]$ を用いているが，これは並列プレフィックス演算により $O(n/log\ n)$ 個のプロセッサを用いて $O(log\ n)$ 時間で計算可能である．$M[i]$ の計算では $b_i < P[j]$ になるような最小の j を必要としているが，配列 $P[1:n]$ が昇順にソートされている特性を利用すると二分探索を用いることで $O(n)$ 個のプロセッサを用いて $O(log\ n)$ 時間で $M[1:n]$ は計算可能である．

Step 2は各区間 i において，交差を持つ2番目に大きな区間 $SM[i]$ を算出している．配列 $SP[1:n]$ は $O(n/log\ n)$ 個のプロセッサを用いて $O(log\ n)$ 時間，$M[1:n]$ は $O(n)$ 個のプロセッサを用いて $O(log\ n)$ 時間で計算可能である．

Step 3では判別集合 $D[i]$ を算出している．補題6より判別集合 $D[i]$ の要素は重複することはない．よって，$O(n)$ 個のプロセッサを用いて定数時間で計算可能であるが，Brentの原理を適用すると $O(n/log\ n)$ 個のプロセッサを用いて $O(log\ n)$ 時間で計算可能となる．

Step 4では判別集合 $D[i]$ の要素と配列 $a[1:n]$ の要素間に共通要素が存在するか否かで，$M[i]$ が関節節点であるかの判別を行っている．配列 $a[1:n]$ の要素をソートすることによって判別集合 $D[i]$ の各要素に対して並列に2分探索が適用可能である．最適並列ソーティングは $O(n)$ 個のプロセッサを用いて $O(log\ n)$ 時間で計算可能である．

Theorem 2　区間グラフ $I = (V, E)$ において，全関節節点を求める $O(n)$ 個のプロセッサを用いた $O(log\ n)$ 時間の効率的並列アルゴリズムを *CREW PRAM* 計算機上で構築することが可能である．□

参考文献

[1] J.M. Chang, C.C. Hsu, Y.L. Wang and T.Y. Ho, An efficient algorithm for finding the set of all hinge vertices in strongly chordal graphs, *Proc, Internat. Computer Symp.* (1994) 277-282.

[2] Ting-Yem Ho, Yue-Li Wang and Ming-Tsan Juan, A linear time algorithm for finding all hinge vertices of a permutation graph, *Inform. Process. Lett.* **59** (1996) 103-107.

[3] M. C. Golumbic, *Algorithmic Gruph Theory and Perfect Graphs*, Academic Press, New York (1988).