

多重バス通信を用いた動的計画法の計算

渋 沢 進

茨城大学工学部情報工学科

多重バス結合は、超立方体結合より比較的少数のデータを高速に転送することができ、しかも分かりやすい並列性を有している。結合モデル上で少数のデータを再帰的かつ並列的に転送することによって解くことのできる問題のひとつに動的計画法がある。本報告では、多角形の最適3角形分割を多重バス結合と超立方体結合モデル上で動的計画法を用いて解く方法を導き、その計算時間を定量的に評価比較し、実行例を示している。 $N = \sqrt{N} \times \sqrt{N}$ ノードをもつ2次元大域バス結合と超立方体結合モデル上で、 $n (\leq \sqrt{N})$ 頂点多角形の最適3角形分割は、それぞれ動的計画法を用いてオーダー $O(n)$, $O(n \log N)$ 時間で解くことができる。これは、多角形の最適3角形分割を動的計画法で求めるとき、多重バス結合の方が超立方体結合よりも効率的であることを示している。

Dynamic Programming Computation Using Multiple Bus Broadcast

Susumu Shibusawa

Department of Computer and Information Science,
Faculty of Engineering, Ibaraki University
4-12-1 Nakanarusawa-machi, Hitachi, Ibaraki 316, Japan

E-mail: sibusawa@ibaraki.ac.jp

Data at a node can be transmitted to other nodes in $O(1)$ steps on a global bus model, while it takes $O(\log N)$ steps on an N -node hypercube. Data transmission between two nodes through global connections is more efficient than transmission using hypercube connections for small quantities of data. Dynamic programming is an operation involving recurrence of parallel data transmission for small quantities of data on interconnection models. This paper presents a method and an example to triangulate a polygon using dynamic programming on a multiple bus connection model and a hypercube. For $n \leq \sqrt{N}$, a polygon with n vertices can be triangulated in $O(n)$ and $O(n \log N)$ time, respectively, on a multiple bus connection model and the hypercube with $N = \sqrt{N} \times \sqrt{N}$ nodes.

1 はじめに

並列的で高速な処理を実現する方式が多く分野で強く求められている。多数の処理要素を相互に関連付け、並列的に動作させることによって高速な処理を実現しようとする方式は、高速で並列的な処理に対する重要な方式のひとつである。

処理要素の相互関係を処理要素間の通信のための相互結合とみなすとき、処理要素とこれらを結合する配線より成る構成はそれぞれノード(頂点)と結合線(辺)より成る相互結合モデルで表現することができる。ノード間の結合パターンに応じて、これまで局所結合、大域結合モデルなどが提案されてきた。局所結合モデルには格子、木、超立方体結合モデルなどが含まれる¹⁾。大域結合モデルとしては1次元、2次元単一、2次元直交多重大域バス結合モデルなどが提案されてきた^{2),3)}。2次元直交多重大域バス結合は、2次元的に配置されたノードを各行及び列方向に直交する複数の大域バスで結合する構成である。大域バス結合は、一般に局所結合よりも一度に多くのノードにデータを放送できるという特徴をもつ。

これまで、与えられた問題を高速に解くために各種の結合モデルが用いられてきた。そのうち、超立方体結合モデルは強力な計算・通信能力をもつ結合モデルであるが、いくつかの操作に対してはそれほど高速でない場合もある。超立方体結合モデルと2次元直交多重大域バス結合を比較すると、超立方体結合の方が大域バス結合よりも速い場合、その逆の場合、計算時間のオーダーが等しい場合がある。バイトニックソートや高速フーリエ変換などは超立方体結合モデル向きの問題である。これに対して、大域バス結合の方が超立方体結合モデルよりも高速に解くことのできる問題としてデータ転送操作や動的計画法などが報告されている⁴⁾⁻⁶⁾。

本報告では、2次元直交多重大域バス結合モデル上で動的計画法を用いて多角形を最適3角形分割する方法を導き、超立方体結合モデルを含む他の結合モデルの場合と比較している。多角形の基本図形への分割は、画像処理、コンピュータグラフィックス、VLSI設計などに現れる基本的な幾何学的問題のひとつである⁷⁾⁻⁹⁾。動的計画法は比較的少数のデータをノード間で再帰的かつ並列的に転送することによって解くことのできる問題である。その結果、 $N = \sqrt{N} \times \sqrt{N}$ ノードをもつ2次元直交多重大域バス結合と超立方体結合モデル上で、 $n (\leq \sqrt{N})$ 頂点多角形の最適3角形分割は動的計画法を用いてそれぞれオーダー $O(n)$, $O(n \log N)$ 時間で解くことができる。また、本報告では、結合モデル上でのアルゴリズムの動作の理解のために具体的な例も示している。

2 結合線の通信能力

本報告で用いる相互結合モデルの各ノードは1ビットの基本演算を単位時間で計算する能力をもつとする。また、各ノードは結合モデルのノード数 N または解くべき問題のサイズ n に依存したあまり大きくない記憶能力をもつとする。

1本の結合線は単位時間に基本単位量のデータを送ることができる通信能力をもつとする。また、ノードの結合パターンに応じて、結合モデルの各ノードが一度に送信できるノード数に関する通信能力と、その結合モデル内で一度に送信できるデータ数に関する通信能力を定義することができる。前者の通信能力を次のように定義する。

[定義1] 結合モデルの各ノードが一度に送信できるノード数に関する通信能力をノード数通信能力とよぶ。□ N ノードの格子結合(トラス)と超立方体結合モデルのノード数通信能力はそれぞれ4と $\log_2 N$ である。これに対して、 N ノードの1次元、2次元単一、2次元直交多重大域バス結合モデルの各ノードは、それぞれ1ステップで N , N , \sqrt{N} ノードに送信できるので、これらの値がノード数通信能力となる。また、データ数に関する通信能力を次のように定義する。

[定義2] 結合モデル内で一度に並列的に送信できるデータ数に関する通信能力を並列データ数通信能力とよぶ。□

N ノードより成る1次元および2次元単一大域バス結合では、一度に1データを N ノードに送信できるが、複数データを送るにはその回数だけ送信操作を繰り返さなくてはならない。これに対して分割された複数の大域バス結合をもつ構成では、1度に全ノード数 N より少ないノードにしかデータを送ることができないが、各大域バスに結合されたノードに選択的かつ並列的にデータを送ることができる。 N ノード2次元直交多重大域バス結合モデルでは行方向および列方向にそれぞれ \sqrt{N} 本の大域バス結合をもち、各方向にそれぞれ異なってもよい \sqrt{N} データを1ステップで並列的に送信することができる。

本報告で取り扱う2次元直交多重大域バス結合モデルを簡単に2次元大域バス結合または多重バス結合モデルとよぶ。この結合モデルを図1に示す。この図において□はノードを表し、点線は大域バス結合を表す。また、比較のために本報告では格子結合モデルと超立方体結合モデルを用いる。図2はノードを反射2進順に配置した超立方体結合モデルを示している。図中の実線は超立方体結合を表す。

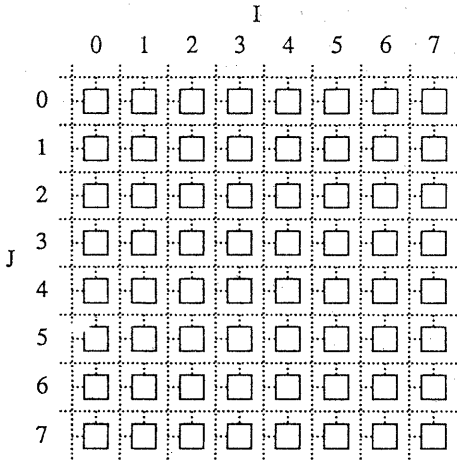


図1 多重パス結合モデル ($N = 8 \times 8$)

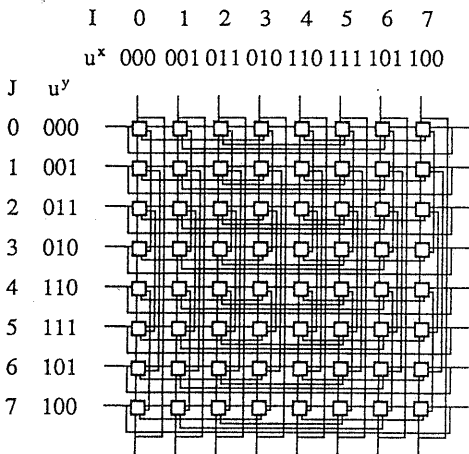


図2 反射2進順にレイアウトした超立方体結合モデル ($N = 2^3 \times 2^3$)

3 多角形の最適3角形分割

3.1 動的計画法を用いた多角形の3角形分割

2次元的に配置されたノードより成る結合モデル上で各種問題を解くとき、動的計画法は比較的少数のデータをノード間で再帰的かつ並列的に転送することによって解くことのできる問題である。

多角形を互いに交差しない弦の集合によって3角形に分割するとき、3角形のある規範に対して最適となるように3角形分割を求める⁷⁾⁻⁹⁾。よく知られている規範としては弦の長さの和を最小にするような3角形分割があり、ここでもその規範を用いる。問題を簡単にするために取り扱う多角形は凸多角形とする。

多角形の3角形分割問題を動的計画法を用いて解く。 $n \geq 3$ の正整数 n に対して、右回りに番号付

けた n 頂点 v_0, v_1, \dots, v_{n-1} より成る多角形を $P = (v_0, v_1, \dots, v_{n-1})$ で表し、この多角形 P に対する動的計画法を用いた最適3角形分割問題を S とおく。 $0 \leq i \leq n-3, 3 \leq s \leq n-i$ に対して、ある頂点 v_i から始まって右回りに s 個の頂点で作られる部分多角形を $P_{i,s} = (v_i, v_{i+1}, \dots, v_{i+s-1})$ とし、この部分多角形に対する3角形分割の部分問題を $S_{i,s}$ とおく。多角形 P の部分多角形 $P_{i,s}$ を図3に示す。動的計画法を用いて部分問題 $S_{i,s}$ を解くときの場合分けは次の3通りとなる⁷⁾。

[解法の方針]

1. 頂点 v_{i+1} に対して弦 $(v_i, v_{i+s-1}), (v_{i+1}, v_{i+s-1})$ と辺 (v_i, v_{i+1}) で3角形を作り、残りの部分多角形 $P_{i+1,s-1}$ に対して部分問題 $S_{i+1,s-1}$ を解く。
2. 頂点 v_{i+s-2} に対して弦 $(v_i, v_{i+s-1}), (v_i, v_{i+s-2})$ と辺 (v_{i+s-2}, v_{i+s-1}) で3角形を作り、残りの部分多角形 $P_{i,s-1}$ に対して部分問題 $S_{i,s-1}$ を解く。
3. $2 \leq k \leq s-3$ であるような k に対して頂点 v_{i+k} を選び、弦 $(v_i, v_{i+k}), (v_{i+k}, v_{i+s-1}), (v_i, v_{i+s-1})$ で3角形を作り、残りの2つの部分多角形 $P_{i,k+1}, P_{i+k,s-k}$ に対して部分問題 $S_{i,k+1}, S_{i+k,s-k}$ を解く。 □

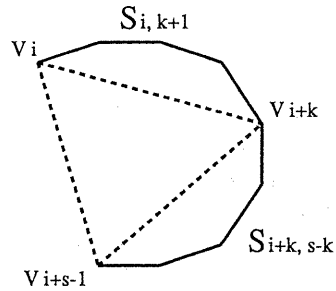


図3 多角形 P の部分多角形 $P_{i,s}$

多角形 P の任意の2頂点を v_i, v_j ($i \neq j$) とするとき、 v_i, v_j が隣り合わない場合に弦 (v_i, v_j) は多角形 P を分割し、多角形の3角形分割における弦の長さの和に弦 (v_i, v_j) の長さが加算される。 v_i と v_j が隣り合う場合は辺 (v_i, v_j) が多角形 P を分割せず、多角形の3角形分割問題に寄与しない。

$0 \leq i < j \leq n-1$ に対して、多角形 P の任意の2頂点 v_i, v_j の距離を d_{ij} で表し、 P の3角形分割における d_{ij} と関係する距離係数 D_{ij} を次のように定義する。

$$D_{ij} = \begin{cases} d_{ij} & (v_i \text{ と } v_j \text{ が隣接しない場合}) \\ 0 & (\text{その他の場合}) \end{cases} \quad (1)$$

ここでは d_{ij} として弦 (v_i, v_j) のユークリッド距離を用いる。

部分多角形 $P_{i,s}$ の3角形分割において、分割弦の長さの和の最小値を部分問題 $S_{i,s}$ のコストとよび、記号 $C_{i,s}$ で表す。コスト $C_{i,s}$ は上記の解法の方針と式(1)を用いて次のように表される。

$$C_{i,s} = \begin{cases} 0 & (s \leq 3) \\ \min_{1 \leq k \leq s-2} \{C_{i,s}^{(k)}\} & (s \geq 4) \end{cases} \quad (2)$$

ここに

$$C_{i,s}^{(k)} = C_{i,k+1} + C_{i+k,s-k} + D_{i,s}^{(k)} \quad (3)$$

$$D_{i,s}^{(k)} = D_{i,i+k} + D_{i+k,i+s-1} \quad (4)$$

式(2)のコスト $C_{i,s}$ は部分多角形 $P_{i,s}$ の辺 (v_i, v_{i+s-1}) の長さを含まないものとする。式(3)は図3に対応している。

3.2 問題と2次元配列の対応づけ

$N = \sqrt{N} \times \sqrt{N}$ ノードより成る2次元配列を (I, J) ($0 \leq I, J \leq \sqrt{N} - 1$) とし、この配列上で動的計画法を用いた n 多角形の3角形分割問題を解く^{10),11)}。ここでは簡単のために $n \leq \sqrt{N}$ とする。問題とノード配列の対応を次のように設定する。

[問題とノード配列の対応]

n 多角形 P の部分多角形 $P_{i,s}$ ($0 \leq i \leq n-3, 3 \leq s \leq n-i$) の3角形分割において、分割弦の長さの和の最小値 $C_{i,s}$ をノード $(I, J) = (i, s+i-1)$ ($0 \leq I \leq J-2, 2 \leq J \leq n-1$) で求める。

$$i = I, \quad s = J - I + 1 \quad (5)$$

最終的な結果をノード $(0, n-1)$ のコスト $C_{0,n}$ で与える。□

上記の対応づけはノード (I, J) でコスト $C_{I,J-I+1}$ を求めることを意味する。ノード配列 I, J をそれぞれ水平右方向と垂直下方向にとれば、係数 $C_{i,s}$ は2次元配列の対角線より左下のノードを用いて求められる。係数 $C_{i,s}$ と2次元配列のノードの対応例を図4に示す。各ノードは高々 n 個のデータを保持できるものと仮定する。

	I							
	0	1	2	3	4	5	6	7
0								
1	C_{02}							
2	C_{03}	C_{12}						
J 3	C_{04}	C_{13}	C_{22}					
4	C_{05}	C_{14}	C_{23}	C_{32}				
5	C_{06}	C_{15}	C_{24}	C_{33}	C_{42}			
6	C_{07}	C_{16}	C_{25}	C_{34}	C_{43}	C_{52}		
7	C_{08}	C_{17}	C_{26}	C_{35}	C_{44}	C_{53}	C_{62}	

図4 係数 $C_{i,s}$ と2次元配列の対応 ($N = 8 \times 8$)

3.3 2次元結合モデルを用いた多角形の3角形分割のコスト

以下では、各ノードで必要な距離係数 $D_{i,s}$ が既に与えられていると仮定する。多角形を3角形分割する弦の長さの和の最小値 $C_{0,n}$ は、式(2)~(4)に従って $C_{i,s}$

を s の小さい方から順に求めていけば良い。そのアルゴリズムの概要は次のようになる。

[アルゴリズムの概要1] (分割コスト)

1. ノード (I, J) ($0 \leq I < J \leq n-1$) は多角形を3角形分割する弦の長さの和 $C_{i,s}$ ($i = I, s = J - I + 1$) を次のように初期化する。

$$C_{i,s} \leftarrow \begin{cases} 0 & (s \leq 3) \\ L & (s \geq 4) \end{cases} \quad (6)$$

ここに L は十分大きい数とする。

2. 各ノード (I, J) ($0 \leq I \leq J-2, 2 \leq J \leq n-1$) は $D_{i,s}^{(k)}$ ($1 \leq k \leq J - I - 1$) を求める。
3. ノード $(I, J) = (I, I+t-1)$ ($2 \leq t \leq n-1, 0 \leq I \leq n-t$) はもっている弦の和 $C_{I,t}$ を水平方向に送る。
4. ノード $(I, J) = (I, I+s-1)$ ($t < s, 0 \leq I \leq J-s, s \leq J \leq n-1$) はこの $C_{I,t}$ を取り込む。
5. 弦の長さの和 $C_{i,s}$ の部分項 $C_{i,s}^{(k)}$ を計算できるノード $(I, J) = (I, I+s-1)$ はこれを計算し、次の最小値を求める。

$$C_{i,s} \leftarrow \min \{C_{i,s}, C_{i,s}^{(k)}\} \quad (7)$$

6. ノード $(I, J) = (I, I+t-1)$ ($2 \leq t \leq n-1, 0 \leq I \leq n-t$) はもっている弦の和 $C_{I,t}$ を垂直方向に送る。
7. ノード $(I, J) = (I, I+s-1)$ ($t < s, I+t \leq J \leq n-1, 0 \leq I \leq n-t-1$) はこの $C_{I,t}$ を取り込む。
8. 弦の長さの和 $C_{i,s}$ の部分項 $C_{i,s}^{(k)}$ を計算できるノード $(I, J) = (I, I+s-1)$ はこれを計算し、式(7)の最小値を求める。
9. 操作(3)~(8)を $t = 2, 3, \dots, n-1$ に対して繰り返す。□

2次元大域パス結合モデル上で、多角形を3角形分割する弦の長さの最小和を求める操作は次のようになる。

[アルゴリズム1] $N = \sqrt{N} \times \sqrt{N}$ ノードの2次元大域パス結合モデル上で、 n ($\leq \sqrt{N}$) 頂点多角形 P の最適3角形分割コストを動的計画法を用いて求める。

入力：多角形の頂点間の距離係数 $D_{i,j}$ ($0 \leq i < j \leq n-1$)。

出力：多角形 P の頂点 v_i ($0 \leq i \leq n-3$) から始めて右回りに s ($3 \leq s \leq n-i$) 個の頂点で作られる部分多角形 $P_{i,s}$ に対して、3角形分割の弦の長さの最小和を $C_{i,s}$ とする。このとき、もとの多角形 P を3角形に分割したときの弦の長さの和の最小値 $C = C_{0,n}$ をノード $(0, n-1)$ で求める。

方法：図5の手続き $\text{triangulation}(n, D_{i,s})$ による。記号 $\text{for} \dots \text{do begin} \dots \text{end}$ と $\text{for} \dots \text{pardo begin} \dots \text{end}$ はそれぞれは順序的な操作と並列的な操作を表す。図において、第3, 4行でノード (I, J) と係数 i, s を対応づけている。第5行は各ノードにおける弦の長さの和 $C_{i,s}$ の初期化であり、第6, 7行は頂点距離の起こりうる場合の和である。第8~24行が動的計画法の本体で

あり、そのうち第9～15行と第16～22行がそれぞれ水平、垂直方向の操作である。第9, 16行での係数 C_{Ii} の水平、垂直方向への転送には大域結合を用いる。第11, 13, 18, 20行の条件はコスト C_{is} の部分項 $C_{is}^{(k)}$ を計算できるノード (I, J) の場合分けを示す。 $C_{is}^{(k)}$ の計算できたノードは第15, 22行を実行し、最終的にノード $(0, n-1)$ でコスト $C = C_{0n}$ が求まる。 □

```

procedure triangulation( $n, D_{is}$ );
{ initialization }
1. for  $I \leftarrow 0$  to  $n-2$  pardo begin
2.   for  $J \leftarrow I+1$  to  $n-1$  pardo begin
3.      $i \leftarrow I$ ;
4.      $s \leftarrow J - I + 1$ ;
5.      $C_{is} \leftarrow \begin{cases} 0 & (s \leq 3) \\ L & (s \geq 4) \end{cases}$ 
6.     { chordal length  $D_{is}^{(k)}$  }
7.     for  $k \leftarrow 1$  to  $s-2$  do begin
8.        $D_{is}^{(k)} \leftarrow \begin{cases} 0 & (J = I+k) \\ D_{i,i+k} + D_{i+k,i+s-1} & (I+k+1 \leq J \leq n-1) \end{cases}$ 
9.     end; {  $k$  }
10.    { triangulation of polygon }
11.    for  $t \leftarrow 2$  to  $n-1$  do begin
12.      { horizontal operations }
13.      node  $(I, I+t-1)$  transmits  $C_{Ii}$ 
14.      through horizontal connections;
15.      node  $(I, I+s-1)$  ( $s > t$ ) takes in  $C_{Ii}$ ;
16.      if  $3 \leq t \leq \lfloor (n+1)/2 \rfloor$  then
17.         $C_{is}^{(k)} \leftarrow C_{i,k+1} + C_{i+k,s-k} + D_{is}^{(k)}$ 
18.        for a certain  $k$  ( $k = s-t$ )
19.        at node  $(I, I+s-1)$  ( $t < s \leq 2t-2$ );
20.      if  $\lfloor (n+1)/2 \rfloor < t \leq n-1$  then
21.        node  $(I, I+s-1)$  ( $t < s \leq n$ ) computes  $C_{is}^{(k)}$ 
22.        for a certain  $k$  ( $k = s-t$ );
23.       $C_{is} \leftarrow \min\{C_{is}, C_{is}^{(k)}\}$ 
24.      at the node of steps 12 and 14;
25.      { vertical operations }
26.      node  $(I, I+t-1)$  transmits  $C_{Ii}$ 
27.      through vertical connections;
28.      node  $(I, I+s-1)$  ( $s > t$ ) takes in  $C_{Ii}$ ;
29.      if  $3 \leq t \leq \lfloor (n-1)/2 \rfloor$  then
30.        node  $(I, I+s-1)$  ( $t < s \leq 2t-1$ ) computes  $C_{is}^{(k)}$ 
31.        for  $k = t-1$ ;
32.      if  $\lfloor (n-1)/2 \rfloor < t \leq n-1$  then
33.        node  $(I, I+s-1)$  ( $t < s \leq n$ ) computes  $C_{is}^{(k)}$ 
34.        for  $k = t-1$ ;
35.       $C_{is} \leftarrow \min\{C_{is}, C_{is}^{(k)}\}$ 
36.      at the node of steps 19 and 21
37.    end {  $t$  }
38.  end {  $J$  }
39. end. {  $I$  }

```

図5 多角形を3角形分割する弦の長さの和の最小値を求める手続き

アルゴリズム1は次のように評価される。

[定理1] $N = \sqrt{N} \times \sqrt{N}$ ノードの2次元大域バス結合モデル上で、 n ($\leq \sqrt{N}$) 頂点多角形を3角形分割する弦の長さの和の最小値はアルゴリズム1を用いて $O(n)$ 時間で求められる。

(証明) 第6, 7行の距離係数 $D_{is}^{(k)}$ を求める計算はノード $(0, n-1)$ で最も時間がかかり、 $O(n)$ 時間である。第8～22行は各ノードで並列に $O(n)$ 時間で実行できる。 □

大域結合をもたない $N = \sqrt{N} \times \sqrt{N}$ ノードの格子結合モデル上で、 n ($\leq \sqrt{N}$) 頂点多角形の最適3角形分割のコストはアルゴリズム1と同様に求められる。た

だし、手続き $triangulation(n, D_{is})$ の第9, 16行で係数 C_{Ii} の水平、垂直方向への転送には格子結合を用いる。このとき、多角形を3角形分割する弦の長さの和の最小値は $O(n^2)$ 時間で求められる。

次に、超立方体結合上で多角形の3角形分割のコストを求める。

[アルゴリズム2] $N = \sqrt{N} \times \sqrt{N}$ ノード超立方体結合モデル上で、 n ($\leq \sqrt{N}$) 頂点多角形を3角形分割する弦の長さの和の最小値を動的計画法を用いて求める。入出力と方法はアルゴリズム1と同様である。ただし、手続き $triangulation(n, D_{is})$ で、第9, 16行の係数 C_{Ii} の転送には超立方体結合を用いる。 □

[定理2] $N = \sqrt{N} \times \sqrt{N}$ ノード超立方体結合モデル上で、 n ($\leq \sqrt{N}$) 頂点をもつ多角形を3角形分割する弦の長さの和の最小値はアルゴリズム2を用いて $O(n \log N)$ 時間で求められる。 □

定理1と定理2は、多角形を最適3角形分割する弦の長さの和の最小値を動的計画法で求めるとき、2次元大域バス結合の方が超立方体結合よりも効率的であることを示している。

3.4 多角形を最適3角形分割する弦

アルゴリズム1, 2を実行後、部分多角形 P_{is} の3角形分割コスト $C_{is} = \min_k \{C_{is}^{(k)}\}$ を実現する k の値を各ノードに残しておく。このとき、多角形 P を最適3角形分割する弦は第3.1節の解法の方針に従って求めることができる。2次元ノード配列上で分割弦を求めるアルゴリズムの概要は次のようになる。

[アルゴリズムの概要2] (分割弦)

1. 多角形の最適コスト $C = C_{0,n-1}$ をもっているノード $(0, n-1)$ を開始ノードとする。
2. ノード (I, J) はコスト C_{is} を実現する k の値を用いて多角形の分割弦を求める。これは頂点对 $(v_i, v_{i+k}), (v_{i+k}, v_{i+s-1})$ のうち、隣り合わない頂点の対(弦)である。
3. (a) $k \neq 1, k \neq s-2$ のとき、ノード (I, J) の問題 $S_{i,s}$ は部分問題 $S_{i,k+1}$ と $S_{i+k,s-k}$ に分解できる。
 - i. $k \geq 3$ のとき、多角形 $P_{i,k+1}$ は3角形分割が完了していないので、ノード (I, J) はコスト $C_{i,k+1}$ を求めるノード $(i, i+k) \rightarrow$ 操作の継続を知らせる。
 - ii. $s-k \geq 4$ のとき、多角形 $P_{i+k,s-k}$ は3角形分割が完了していないので、ノード (I, J) はコスト $C_{i+k,s-k}$ を求めるノード $(i+k, s+i-1) = (I+k, J) \rightarrow$ 操作の継続を知らせる。
- (b) $k = 1$ のとき、ノード (I, J) の問題 $S_{i,s}$ は問題 $S_{i+1,s-1}$ を解くことに帰着される。 $s \geq 5$ のとき、多角形 $P_{i+1,s-1}$ は3角形分割が完了していないので、ノード (I, J) はコスト $C_{i+1,s-1}$

を求めるノード $(i+1, s+i-1) = (I+1, J)$ へ操作の継続を知らせる。

- (c) $k = s-2$ のとき、ノード (I, J) の問題 $S_{i,s}$ は問題 $S_{i,s-1}$ を解くことに帰着される。 $s \geq 5$ のとき、多角形 $P_{i,s-1}$ は3角形分割が完了していないので、ノード (I, J) はコスト $C_{i,s-1}$ を求めるノード $(i, s+i-2) = (I, J-1)$ へ操作の継続を知らせる。

4. 部分多角形がすべて3角形になるまで、操作2, 3を繰り返す。
5. 分割弦をもっているノード (I, J) は、このノードに操作の継続を連絡したノードへ、水平、垂直方向の結合を用いて分割弦を並列に返送する。
6. すべての分割弦が初期ノード $(0, n-1)$ に送られるまで操作5を同期的に繰り返す。 □

[アルゴリズム3] $N = \sqrt{N} \times \sqrt{N}$ ノードの2次元大域バス結合モデル上で、 $n (\leq \sqrt{N})$ 頂点多角形を最適3角形分割する弦を求める。

入力：手続き $triangulation(n, D_{i,j})$ で求めたコスト $C_{i,s} = \min_k \{C_{i,s}^{(k)}\}$ を実現する k の値が各ノードに残っているとする。

出力：ノード $(0, n-1)$ に多角形を最適3角形分割する弦を集める。

方法：アルゴリズムの概要2を具体化した手続き $chords(k)$ による。詳細は省略する。概要2の3で次ノードへの操作の継続の通知には大域結合を用いる。同様に、概要2の5で分割弦の返送にも大域結合を用いる。 □

[定理3] $N = \sqrt{N} \times \sqrt{N}$ ノードの2次元大域バス結合モデル上で、 $n (\leq \sqrt{N})$ 頂点多角形 P の部分多角形 $P_{i,s}$ の3角形分割コスト $C_{i,s} = \min_k \{C_{i,s}^{(k)}\}$ を実現する k の値が各ノードに与えられているとき、 P を最適3角形分割する弦はアルゴリズム3を用いて $O(n)$ 時間で求められる。 □

大域結合をもたない $N = \sqrt{N} \times \sqrt{N}$ ノードの格子結合モデル上で、 $n (\leq \sqrt{N})$ 頂点多角形 P の部分多角形 $P_{i,s}$ の3角形分割コスト $C_{i,s} = \min_k \{C_{i,s}^{(k)}\}$ を実現する k の値が各ノードに与えられているとき、 P を最適3角形分割する弦はアルゴリズム3と同様に求められる。ただし、操作の継続の通知と分割弦の返送には格子結合を用いる。このとき、分割弦は $O(n)$ 時間で求められる。

[アルゴリズム4] $N = \sqrt{N} \times \sqrt{N}$ ノード超立方体結合モデル上で、 $n (\leq \sqrt{N})$ 頂点多角形 P の部分多角形 $P_{i,s}$ の3角形分割コスト $C_{i,s} = \min_k \{C_{i,s}^{(k)}\}$ を実現する k の値が各ノードに与えられているとき、 P を最適3角形分割する弦を求める。入出力と方法はアルゴリズム3と同様である。ただし、操作の継続の通知と分割弦の返送には超立方体結合を用いる。 □

[系1] 2方向反射2進順の $N = \sqrt{N} \times \sqrt{N}$ ノード超立方体結合モデル上で、 $n (\leq \sqrt{N})$ 頂点多角形 P の

部分多角形 $P_{i,s}$ の3角形分割コスト $C_{i,s} = \min_k \{C_{i,s}^{(k)}\}$ を実現する k の値が各ノードに与えられているとき、 P を3角形分割する弦はアルゴリズム4を用いて $O(n)$ 時間で求められる。 □

定理1, 3より次の結果が成り立つ。

[定理4] $N = \sqrt{N} \times \sqrt{N}$ ノードの2次元大域バス結合モデル上で、 $n (\leq \sqrt{N})$ 頂点多角形の最適3角形分割は $O(n)$ 時間で求められる。 □

大域結合をもたない $N = \sqrt{N} \times \sqrt{N}$ ノードの格子結合モデル上で、 $n (\leq \sqrt{N})$ 頂点多角形の最適3角形分割は $O(n^2)$ 時間で求められる。

定理2と系1より次の結果が成り立つ。

[定理5] 2方向反射2進順の $N = \sqrt{N} \times \sqrt{N}$ ノード超立方体結合モデル上で、 $n (\leq \sqrt{N})$ 頂点多角形の最適3角形分割は $O(n \log N)$ 時間で求められる。 □

4 例

図6のような凸7角形 $P = (v_0, v_1, \dots, v_6)$ の最適3角形分割⁹⁾を2次元大域バス結合モデル上で求める。この7角形の各頂点間の距離 d_{ij} は表1のようである。距離係数 D_{ij} は式(1)を用いて求められる。

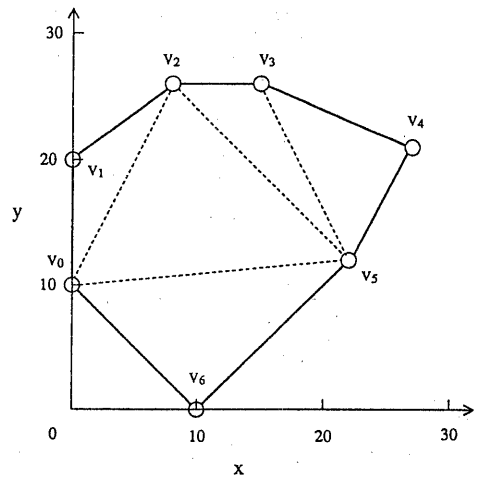


図6 凸7角形の例

表1 凸7角形の頂点間距離 d_{ij} ($i < j$)

i	0	1	2	3	4	5	6
j							
0	-						
1	10.0	-					
2	17.9	10.0	-				
3	21.9	16.2	7.0	-			
4	29.2	27.0	19.6	13.0	-		
5	22.1	23.4	19.8	15.7	10.3	-	
6	14.1	22.4	26.1	26.5	27.0	17.0	-

4.1 初期化

手続き $triangulation(n, D_{i,s})$ の第3～5行より、各ノード (I, J) はパラメータ i, s の値を並列的に決定し、多角形の弦の長さの和 $C_{i,s}$ を初期化する。次に、手続き $triangulation(n, D_{i,s})$ の第6, 7行より、各ノードはもっている係数 $D_{i,s}$ を用いて、ノードに関して並列的で k に関して順序的に $D_{i,s}^{(k)}$ を求める。

4.2 3 三角形分割のコスト

手続き $triangulation(n, D_{i,s})$ の第8行以下を実行する。

(1) $t = 2$ のとき

1. 水平操作

- (a) ノード $(I, I+t-1) = (I, I+1)$ は $C_{I2} = 0$ を水平方向の結合を用いて転送する。
- (b) ノード $(I, I+s-1)$ ($s > 2$) はこの C_{I2} を取り込む。

2. 垂直操作

- (a) ノード $(I, I+t-1) = (I, I+1)$ は $C_{I2} = 0$ を垂直方向の結合を用いて転送する。
- (b) ノード $(I, I+s-1)$ ($s > 2$) はこの C_{I2} を取り込む。

(2) $t = 3$ のとき

1. 水平操作

- (a) ノード $(I, I+t-1) = (I, I+2)$ は $C_{I3} = 0$ を水平方向の結合を用いて転送する。
- (b) ノード $(I, I+s-1)$ ($s > 3$) はこの C_{I3} を取り込む。
- (c) 手続きの第11行目の条件が成り立つので、第12行の $t < s \leq 2t-2$, $k = s-t$ を満たすノードで $C_{i,s}^{(k)}$ を求める。今の場合 $s = 4$, $k = 1$ であり、 $C_{i,s}^{(k)}$ は表2 (a) のようになる。
- (d) $C_{i,s}^{(k)}$ を求めたノードで最小値 $C_{i,s}$ を計算すると表2 (a) のようになる。

2. 垂直操作

- (a) ノード $(I, I+t-1) = (I, I+2)$ は $C_{I3} = 0$ を垂直方向の結合を用いて転送する。
- (b) ノード $(I, I+s-1)$ ($s > 3$) はこの C_{I3} を取り込む。
- (c) 手続きの第18行目の条件が成り立つので、第19行の $t < s \leq 2t-1$, $k = t-1$ を満たすノードで $C_{i,s}^{(k)}$ を求める。今の場合 $s = 4, 5$, $k = 2$ であり、 $C_{i,s}^{(k)}$ は表2 (b) のようになる。
- (d) $C_{i,s}^{(k)}$ を求めたノードで最小値 $C_{i,s}$ を計算すると表2 (b) のようになる。この結果、 $s = 2$ のときの $C_{i,s}$ が完成した。ここまでに求められた $C_{i,s}$ を図示すると図7のようになる。

以下同様にしてコスト $C_{i,s}$ が求められる。最終的に、 $t = 6$ に対する操作終了後の最小コスト C は

$$C = C_{06} = 75.5$$

表2 $t = 3$ のときの $C_{i,s}^{(k)}$ と $C_{i,s}$ の値
(a) 水平操作 (b) 垂直操作

s	k	I	$I+s-1$	$C_{i,s}^{(k)}$	$C_{i,s}$	k satisfied $C_{i,s}^{(k)}$
4	1	0	3	$C_{04}^{(1)} = 16.2$	16.2	1
		1	4	$C_{14}^{(1)} = 19.6$	19.6	1
		2	5	$C_{24}^{(1)} = 15.7$	15.7	1
		3	6	$C_{34}^{(1)} = 27.0$	27.0	1

(a)

s	k	I	$I+s-1$	$C_{i,s}^{(k)}$	$C_{i,s}$	k satisfied $C_{i,s}^{(k)}$
4	2	0	3	$C_{04}^{(2)} = 17.9$	16.2	1
		1	4	$C_{14}^{(2)} = 16.2$	16.2	2
		2	5	$C_{24}^{(2)} = 19.6$	15.7	1
		3	6	$C_{34}^{(2)} = 15.7$	15.7	2
5	2	0	4	$C_{05}^{(2)} = 37.5$	37.5	2
		1	5	$C_{15}^{(2)} = 31.9$	31.9	2
		2	6	$C_{25}^{(2)} = 46.6$	46.6	2

(b)

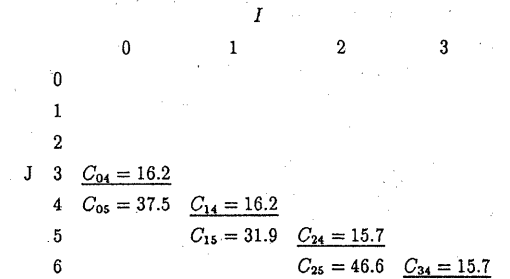


図7 $t = 3$ のときのコスト $C_{i,s}$. コスト C_{I4} は完成した

4.3 分割弦

アルゴリズム3を用いて、多角形を最適3三角形分割する弦をノード $(0, n-1)$ に集める。

(1) 前方操作

1. ノード $(I, J) = (0, 6)$

- (a) 分割弦と部分問題を求める。 $s = J - I + 1 = 7$ であり、最小値 $C_{07} = \min_k \{C_{07}^{(k)}\}$ を満たす k は $k = 5$ であることより、 $i = 0$ に対して分割弦は

$$(v_i, v_{i+k}) = (v_0, v_5)$$

これによって多角形を3三角形分割する問題 S は $S_{i,s-1} = S_{06}$ を解くことに帰着される。

- (b) 部分問題 S_{06} のコスト C_{06} を求めるノードは $(i, s+i-1) = (0, 5)$ より、ノード $(0, 6)$ はノード $(0, 5)$ に操作の継続を知らせる。

以下同様に、分割弦を求める操作はノード(0,5)を經由してノード(2,5)で終了する。

(2) 分割弦の返送

初期ノード(0,6)以外で分割弦を所有しているノード(0,5), (2,5)は、水平、垂直方向の結合を用いて同期的に分割弦を初期ノードに返送する。その結果、ノード(0,6)に次の分割弦が集められる。

$$(v_0, v_2), (v_0, v_5), (v_2, v_5), (v_3, v_5)$$

これによって、凸7角形の3角形分割問題が解け、図6の点線のようにになる。

5 おわりに

多角形の最適3角形分割を求める問題を、多重バス結合と超立方体結合モデル上で動的計画法を用いて解く方法を導き、その効率を評価した。 $N = \sqrt{N} \times \sqrt{N}$ ノードをもつ多重バス結合と超立方体結合モデル上で、 $n (\leq \sqrt{N})$ 頂点多角形の最適3角形分割は、それぞれオーダー $O(n)$, $O(n \log N)$ 時間で解くことができる。また、並列アルゴリズムの動作のより良い理解のために、多角形の最適3角形分割を求める具体的な例を示した。

大域バス結合と超立方体結合モデルを用いて一般の問題を解くとき、大域バス結合の方が超立方体結合よりも速い場合、その逆の場合、計算時間のオーダーが等しい場合の3通りがある。2ノード間のデータ系列の転送に対しては、データ数がオーダー $O(\log N)$ より少ない場合には大域バス結合の方が超立方体結合より速い⁴⁾⁻⁶⁾。本報告で示した動的計画法を用いた多角形の最適3角形分割はこの場合に相当する。今後、さらに多くの問題に対して高速化手法を定量的に評価していくことは意義ある課題のひとつであると考えられる。

参考文献

- 1) Ranka, S. and Sahni, S.: *Hypercube Algorithms*, Springer-Verlag, New York (1990).
- 2) Stout, Q. F.: Mesh-connected computers with broadcasting, *IEEE Trans. Comput.*, Vol.C-32, No.9, pp.826-830 (1983).
- 3) 梅尾博司, グローバルキャスト機能を備えた網目結合並列計算機のためのアルゴリズムに関する最近の研究, *情報処理*, Vol.31, No.7, pp.906-912 (1990).
- 4) Shibusawa, S.: The Cubemat: a parallel interconnection model consisting of hypercube and global bus connections, *Proc. Int. Conf. on Information Technology*, Vol.1, pp.201-208 (1990).
- 5) 渋谷進, 超立方体結合と大域結合より成る一結合モデルについて, *情報処理学会論文誌*, Vol.34, No.4, pp.533-543 (1991).
- 6) Shibusawa, S.: Operations on an interconnection model consisting of hypercube and global bus connections, in *Parallel Computing 91* edited by D. J. Evans, G. R. Joubert and H. Liddell, pp.331-338, North-Holland, Amsterdam (1992).
- 7) Aho, A. V., Hopcroft, J. E. and Ullman, J. D.: *Data Structures and Algorithms*, Addison-Wesley, Massachusetts (1983) (大野義夫訳, データ構造とアルゴリズム, 培風館 (1987)).
- 8) Cormen, T. H., Leiserson, C. E. and Rivest, R. L.: *Introduction to Algorithms*, MIT Press, Massachusetts (1990).
- 9) 浅野哲夫: 計算幾何学, 朝倉書店 (1990).
- 10) 渋谷進, 大域バス結合モデルを用いた動的計画法の計算, 第43回情処全大論文集, 5Q-4 (1991).
- 11) 渋谷進, 大域バス付加超立方体計算機結合モデル, 電子情報通信学会技術研究報告, CPSY91-35 (1991).