再配置可能バス付き2次元アレイ上における最近点探索アルゴリズム†

1P－4

中嶋 幸治　若林 真一　小出 哲士　吉田 典可
広島大学 工学部

1 まえがき

グラフィックス，Computer-Aided-Design(CAD) などの普及により，図形に関する処理の高速化を目的として，計算幾何学が注目を集めている．一方，近年の半導体技術の進歩により数千あるいは数万プロセッサをもつ並列計算機が実現できるようになってきた．このような状況の下で，現在までに，各種の計算幾何学の問題に対して多くの並列アルゴリズムが提案されてきた．計算幾何学問題に対して用いられてきた並列計算機のモデルとしてはハイパーキューブ，リニアアレイ，2次元メッシュ[3], 2次元アレイをベースとしたピラミッドコンピュータ，木マシン，網目木結合計算機，メッシュバス [1], そしてPRAM等がある．本稿では全最近点問題やユークリッド最小木問題を再配置可能バス付き2次元プロセッサアレイ上で解く並列アルゴリズムを提案する．

2 並列計算機モデル

本研究で用いる再配置可能バス付メッシュ(以下 RM)(図 1) を以下のように定義する．

- 内部にローカルメモリを有する $n\times n$ の n^2サイズの2次元メッシュをベースとする．
- 各プロセッサは4方向 (上，下，左，右) のスイッチ (ローカルコントロールバススイッチ) を持つ．
- 各プロセッサは単位時間で算術論理演算を行なう．
- 各プロセッサはバスを介してバスに接続されたすべてのプロセッサに情報を単位時間で伝達することができる．
- バスには図1に示したアレイの横方向の行バスと縦方向の列バスがある．各プロセッサのスイッチ操作により行 (列) バスを分割することができ，バスによって接続されたアレイ (サブアレイ) 内で独立にブロードキャストが可能である．

3 全最近点問題 ($ANNP$)

3.1 問題の定式化

平面上に n 個の点が与えられた時，各点について最近点を求めよ．

† "Nearest Neighbor Searching Algorithms for 2-Dimensional Arrays with Reconfigurable Buses" by Koji NAKASHIMA, Shin'ichi WAKABAYASHI, Tetsushi KOIDE, and Noriyoshi YOSHIDA, Faculty of Engineering, Hiroshima University.

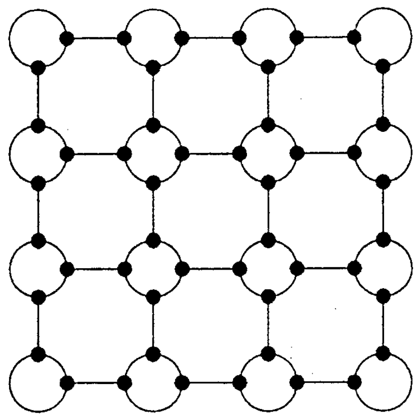

図 1 再配置可能バス付メッシュ

3.2 提案アルゴリズム

全最近点問題に対するアルゴリズムを以下に示す．ただし，アルゴリズムの実行前に，平面上の n 個の点集合を Sとしたとき，Sの各点はプロセッサ $P_{1,j}(1\leqq j\leqq n)$ に格納されているものとする．

[アルゴリズム $ANNA$]

ステップ 1 各プロセッサは各点の座標 (x_j,y_j) を列バスを利用してブロードキャストする．

ステップ 2 プロセッサ $P_{i,i}$は行バスを利用してステップ1でもらった値 (x_i,y_i) をブロードキャストする．

ステップ 3 各プロセッサは受けとった2つの座標間の距離を計算する．

ステップ 4 各列ごとにプロセッサを2分木の葉とみなして同じ親の子同士を比較し，小さい方を左の子とする．これをルートまで繰り返すことにより各点の最近点の座標が求まる．

アルゴリズムの実行例を図2に示す．また，アルゴリズム $ANNA$ の各ステップの計算量はステップ4が2分木の高さの分だけの比較およびブロードキャストを行なうので $O(\log n)$ かかるが，他は $O(1)$ である．したがって次の定理が得られる．

定理 1 サイズ n^2の RM上の最上位行の各プロセッサごとに，高だか1つの点が与えられたとき，サイズ n の全最近点問題はアルゴリズム $ANNA$ を用いて，$O(\log n)$ 計算時間で解ける．□

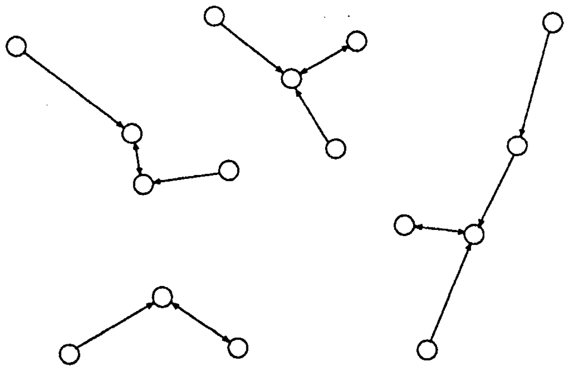

図 2 全最近点の例

4 ユークリッド最小木問題 ($EMSTP$)

4.1 問題の定式化

平面上に n 個の点が与えられたとき，その頂点が与えられた点で，枝長の総和が最小であるような木を構成せよ．

4.2 提案アルゴリズム

問題$EMSTP$に対するアルゴリズムを以下に示す．提案アルゴリズム $EMSTA$ は以下に示す手続き $FIND$ を $O(\log n)$ 回繰り返す．ここで用いるラベルとは，コンポーネント (最小木を構成する枝によって接続された点の集合) 内の点の最小インデックスを指すものとする．なお，平面上の n 個の点集合を Sとしたとき，Sの各点はアルゴリズムの実行前にプロセッサ $P_{1,j}(1 \leqq j \leqq n)$ に格納されているものとする．

[手続き $FIND$]

ステップ 1 コンポーネント内の点のラベルが統一されていなければステップ 3へ

ステップ 2 各点に対して，まず最近点を求める．コンポーネントが2つ以上の点からなるときステップ3により同じコンポーネントに属する点が決定され，これらの点の情報はプロセッサ $P_{1,j}$ に隣接しておかれるため，k個の点からなるコンポーネントの場合$k \times k$の RMを独立に使って，他のコンポーネントへの最近点を決定できる [2]．ここで，同じコンポーネント内の点を選ばないようにするために，各プロセッサ $P_{1,j}$ は前回のステップ3で送られてきた点のインデックス lを列バスを利用して $P_{l,j}$ まで送り，さらに $P_{l,j}$ は行バスを利用して同じコンポーネントにブロードキャストする．2つ以上の点からなるコンポーネントの最近点を求めるとき，同じ距離のものがある場合は，各点の< x 座標, y座標 >の辞書式順序の2項組を最小にするものを選択する．

ステップ 3 このステップは以下の操作により実行される．

1. プロセッサ $P_{1,j}$ は自分の点情報を列バスを利用してブロードキャストする．
2. 点 p_i の最近点 p_j との間に枝を加えるとき，行バスを用いてプロセッサ $P_{i,i}$ はプロセッサ $P_{i,j}$ に点 p_i のラベルを送り，同様に今度は，行バスを用いてプロセッサ $P_{i,j}$ はプロセッサ $P_{i,i}$ に点 p_j のラベルを送る．このとき各プロセッサは2つのラベルを比べて，小さい方のラベルを自分のラベルとする．
3. プロセッサ $P_{i,i}$ のラベルおよび，インデックスを用いてソーティングを行ない，その結果によりプロセッサ $P_{i,i}$ の情報をプロセッサ $P_{1,j}$ にコピーする．
4. 列バスを用いてこのステップ内で点情報を扱ったプロセッサすべてのもつラベルのうち最小のものを選ぶ.(アルゴリズム $ANNA$ のステップ4参照.) これをプロセッサ $P_{1,j}$ から列バスを用いてブロードキャストし，各プロセッサのラベルを小さい方に更新する．
5. すべての点のラベルが1になれば終了．そうでなければステップ 1へ．

ここでステップ3において次の補題が成り立つ．

補題 1 あるコンポーネントから最近点を選び出すときサイクルは生じない． □

アルゴリズム $EMSTA$ の計算量については，手続き $FIND$のステップ2とステップ3の4が $O(\log n)$ である以外は $O(1)$ であり，手続きは $O(\log n)$ 回繰り返されるので，以下の定理が成り立つ．

定理 2 サイズ n^2の RM上の最上位行の各プロセッサごとに，高だか1つの点が与えられたとき，サイズ n のユークリッド最小木問題はアルゴリズム $EMSTA$ を用いて，$O(\log^2 n)$ 計算時間で解ける． □

5 あとがき

今後の課題としては，他の計算幾何学問題を再配置可能バス付き2次元アレイを用いて解くことがあげられる．

参考文献

[1] 堀川, 岩間:“メッシュバス上での最小全域木アルゴリズム,” 情処研報, 94-AL-39, No. 35, pp. 1–8 (1994).

[2] R. Miller, V. K. Prasanna-Kumar, D. I. Reisis and Q. F. Stout: “Parallel computations on reconfigurable meshes,” IEEE Trans. Comput., Vol. C-42, No. 12, pp. 678–692 (1993).

[3] R. Miller and Q. F. Stout: “Mesh computer algorithms for computational geometry,” IEEE Trans. Comput., Vol. C-38, No. 3, pp. 321–340 (1989).