

メッシュパーティショニングの 遺伝的アルゴリズムによる最適化の検討と評価

増田太郎* 吉岡顕†

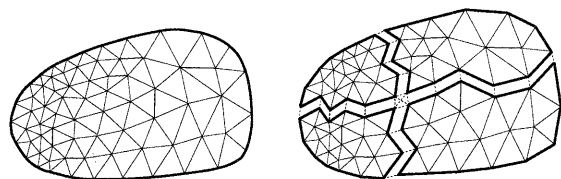
東京大学工学系研究科電子情報工学専攻* 東京大学情報基盤センター†

1 はじめに

近年の計算機性能の向上により、さまざまな分野において有限要素法などによる数値解析は開発、設計、評価のための重要な手段として使われている。より良い精度を求めるために解析対象の大規模化が進むにつれて、解析時間の増大は大きな問題となっている。

その解決法として並列処理が挙げられる。有限要素法を並列化する手法の一つとして、解析領域を分割して並列に解析するものがある。この手法は粒度が大きく高い並列効率を得ることができる。

1000 を越えるプロセッサをもつ超並列計算機では仕事を均等に割り当て、プロセッサ間の通信を少なくしなくては高い並列効率を得ることができない。有限要素法では、解析領域をメッシュに分けて解析するため、並列化するにはこのメッシュを図 1(b) のようにパーティショニングする必要がある。パーティショニングの結果により並列効率に大きな影響を及ぼす。しかし、この問題は NP 完全問題として知られており、迅速かつ高精度な解法が求められている。



(a)Finite element mesh (b)Partitioned mesh

図 1: mesh partitioning

本稿では部分領域とは有限要素の集合を表し、カットエッジとは部分領域の境界上の辺を表す。並列計算を行う時には各部分領域をプロセッサにマッピングする。カットエッジの数は近似的にプロセッサ間での通信量を示す。

*Taro Masuda, Information and Communication Engineering, Engineering Department, Univ. of Tokyo

†Akira Yoshioka, Information Technology Center, Univ. of Tokyo

2 遺伝的アルゴリズム (GA) の適用方法

2.1 GA のスケジューリング

1. 集団を初期化する。
2. 個体を交叉させて新しい個体を生成する。また、ある確率で突然変異を起こさせる。
3. 新たな個体の入る場所を空けるために集団の一部を削除する。(淘汰)
4. 新たな個体の適応度を求め、集団に挿入する。
5. 終了条件が満たされた場合は終了、そうでなければ 2 へ戻る。

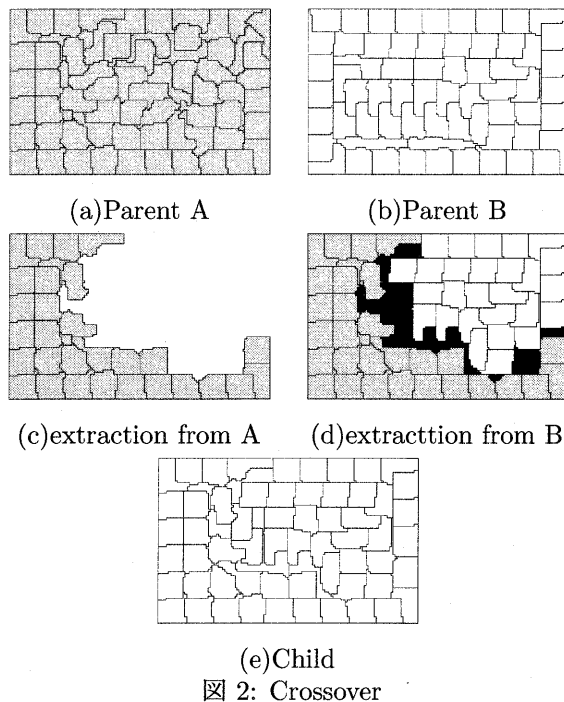
以上のような処理を行うことにより解の精度を向上させる。ここでは、負荷分散は実現されているものとしてカットエッジの数のみを適応度とし、少ない程適応しているとする。

2.2 交叉

遺伝的アルゴリズムにおいて交叉は、親の形質を適切に子に継承させなくてはならない。メッシュパーティショニングにおいては、1 つの部分領域だけを取り出しても形質とはならず、ある程度の大きさの部分領域の集合が形質となる。そこで、部分領域ベースの交叉を実現しなくてはならない。

2 つの個体から半分ずつ情報を取り出すために、始めに図 2(c) のように親 A から全体の約半分にあたる部分領域を抽出する。このときできるだけ優れた情報を残せるように、(カットエッジの数/要素数) が最も小さいものを起点に隣接する部分領域を取り出す。次に図 2(d) のように親 A から抽出した部分と重ならないように親 A から部分領域を取り出す。この段階ではどの部分領域にも割り当てられていない要素が存在するので、これらを Greedy アルゴリズムによって再

パーティショニングする。この結果、図 2(e) のように新しいパーティショニングデータとなる。



2.3 負荷平衡

全ての部分領域の要素数を均等にすることが厳密な意味での負荷分散の実現であるが、最適化の過程において負荷分散にこだわりすぎるとカットエッジの数を減らすことができない。並列有限要素法の計算をしたときに律速になるのは最も大きな部分領域なので、この部分領域の要素数を制限することだけにとどめる。

許容度 t を定義し、(全要素数/部分領域数 $\times t$) より大きい部分領域 S は S と隣接する部分領域との間で要素数が均一になるように要素を移動させる。この操作を新しくできた個体に対して行い、負荷平衡を保つ。

2.4 パラメータ

GA において重要となるパラメータとして集団の数と交叉、突然変異の起こる確率が必要である。今回は個体の数を 30、交叉の確率を 0.6、突然変異の確率を 0.1 とした。

また、負荷平衡を保つために必要な許容度 t は 1.01 とした。

表 1: Example Run

分割数	時間 (s)	カットエッジ 実行後/前	改善度 (%)	負荷
16	108.66	1837/2047	10.26	1.0048
32	83.53	2829/3010	6.01	1.0080
64	80.43	4135/4429	6.64	1.0048
128	78.29	6041/6412	5.79	1.0112

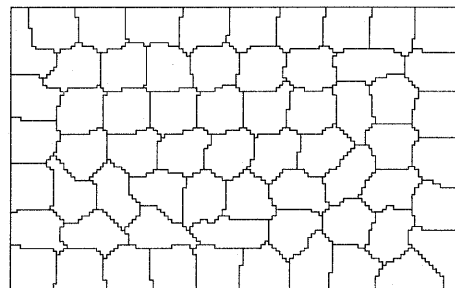


図 3: 10000 要素を 64 分割

3 実行結果

表 1 は 125x80 のメッシュデータを分割数を変えて 400 世代実行したものである。64 分割の場合の結果は図 3 のようになった。6% 程度のカットエッジを除去できている。分割数が少ない場合は交叉のときの結合具合が悪く余計な計算が必要となるため時間が多くなる。

また、負荷分散に関しては 128 分割のものを除いて許容度以内に収まっている。128 分割の場合も理論的に 1.0112 より良くなるので最適化されていることが分かる。

4 今後の課題

本研究の最終目標は並列計算の高速化である。したがって、処理を続けてメッシュパーティショニングの精度が良くなるとしても、その処理時間分以上の並列有限要素法の計算時間が短縮が望めない場合は処理を打ち切るべきである。実際に並列有限要素法の処理を行うことにより処理を打ち切るべきポイントを探っていきたい。

参考文献

- [1] C. Walshaw and M. Cross, "Mesh Partitioning: a Multilevel Balancing and Refinement Algorithm", Tech. Rep. 98/IM/35, Univ. Greenwich, London SE18 6PF, UK, July 1998.