資源追加削減法の並列性の検証

The parallel optimization of DORAR method

森　俊明（知的システムデザイン研究室）
Toshiaki MORI (Intelligent Systems Design Laboratory)

Abstract The DORAR method is a new parallel and distributed algorithm for optimum design of discrete systems, and has been found to be effective for the optimization of electrical circuits and discrete structures so far. This optimization algorithm consists of two processes, namely the resource reduction process and the resource addition process. In this paper, The DORAR method is verified by mountining to PC cluster.

1　はじめに

離散システムの最適設計のために提案された資源追加削減法は，システムを構成する離散的な各要素が，要素に関する情報を頼りに，要素の持つ知識のみで，自律的に挙動し，その結果としてシステム全体がより最適な方向へ近づくという考え方に基づいている．対象問題は，連続変数かつ離散的な要素から構成されるシステムの最適資源配分問題に限定する．資源追加削減法は，任意の初期点から収束するロバスト性，基本的にパラメータフリーであるといった特徴を持ち，アルゴリズムそのものが並列処理に適しているという特徴を有する．

資源追加削減法に関しては，収束性の改善，制約条件の扱いといったアルゴリズム的な側面からの研究は行われていたが，システムの大規模化による変数の増加の伴う並列性の検証は本格的に行われていない．

本論文では，資源追加削減法を，近年安価で容易に構築が可能となった PC クラスタに実装し，トラス構造物最適化問題に適用することで，アルゴリズムの並列性の検証を行う．また，各プロセス間の情報交換を効率的に行うための通信手法に関する検討も行う．

2　資源追加削減法の概略

資源追加削減法は，離散的な各要素が自律分散的に挙動した結果，最適化が達成される手法である．設計変数を資源と考え、各要素の資源の総和，つまり総資源量の最小化を複数の局所制約条件と複数の全体制約条件の下で行う．

DORAR 法のアルゴリズムを以下に示す．

(1) 局所制約条件に関する資源余裕を評価する．
(2) 全体制約条件に関する資源余裕を評価する．
(3) 上の資源余裕の最小値を各要素の臨界資源余裕とし，これを削減する．(資源削減処理)
(4) 各要素に一定の微少な資源を追加する．(資源追加処理)
(5) (1) から (4) を繰り返すことにより最適解を得る．

3　DORAR 法の並列処理の手法

最適化の対象とするシステムを等価のサブシステムに分割し，サブシステムにプロセッサを割り当てることで，並列処理を実現する．

各サブシステムはステップ毎に，資源余裕の見積り，資源削減処理，資源追加処理を並列に行っている．そのため，各サブシステムはシステム全体の状態を保持している必要があるので，自己の資源余裕を他の要素全てにブロードキャストしなければならない．通信後各サブシステムは受信して他の資源余裕からシステム全体の状態推移を求めることが可能となる．その後再び資源余裕を見積り，最適化を繰返すのである．

並列化の仕組を図 3 のフローチャートに示す．

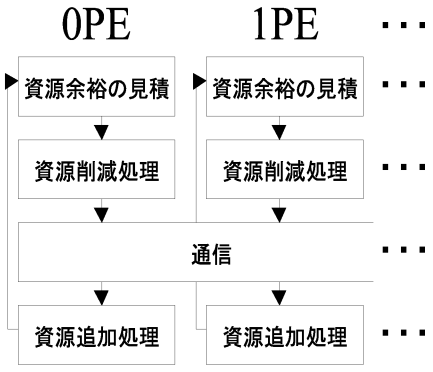

図 1: 並列化の仕組

4 並列の検証

4.1 対象問題と実装した PC クラスタ

本研究では離散的問題としてトラス構造物最適化問題を解く．トラス構造物とは，直接部材を摩擦のないピンで接合した構造物である．さらにピンの空間的な広がりは無視して点であると考え，節点と呼ぶ．トラス構造物は節点と部材からなる．

ある節点に負荷を加え複数の制約条件を与えたとき，最小体積のトラス構造物を求めることが目的である．制約条件は，局所制約条件と全体制約条件に分けられる．局所制約条件として各部材の引張，圧縮応力及び座屈強度を考え，全体制約条件として一つの節点の変位を考える．設計変数は部材の体積である．

このトラス構造物変形解析問題を有限要素法[1]で解く．

本研究で用いたトラス構造は 6 節点 8 部材，18 節点 40 部材，34 節点 80 部材，66 節点 160 部材，130 節点 320 部材である．

使用した計算環境は 16 台の PC(Pentium II 400MH z) からなる PC クラスタである．ネットワークは 100Mbps の Ethernet を用い，OS は Linux2.2.12，通信ライブラリとして MPICH1.1.2 を導入した．

4.2 プロセッサ間における情報交換

並列処理において，計算効率を高めるには通信における送受信の回数を減らすことが重要である．本研究における通信手段はバタフライ通信を用いた．

バタフライ通信とは，使用するプロセッサが 2 のべき乗で表せるとき通信回数が最小となる通信手段のことである．その概要を図 4.2 に示す．また各プロセッサが担当するプロセス数は等価である．

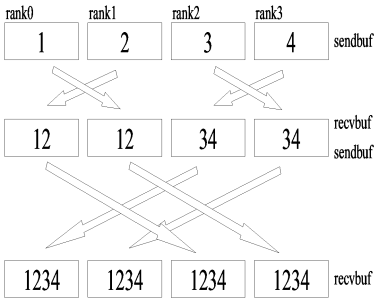

図 2: 4 プロセッサでのバタフライ通信

4.3 計算結果，及び考察

プロセッサ数を多くすると 1 プロセッサにかかる負荷が減少するが．通信回数は増加する．この関係が部材数と計算に使用したプロセッサ数に，どのように影響するかを検証する．

6 節点 8 部材 (図 3 を参照) では並列化することで，単一プロセッサでの最適化と比較して，計算時間が増大する．これは 6 節点 8 部材では粒度が小さすぎるためである．しかし，18 節点 40 部材 (図 4 を参照)，34 節点 80 部材 (図 5 を参照) と部材を増やすにともない並列化効率が高くなっている． これは，1 プロセッサにかかる負荷が大きくなったため，通信の時間の割合が処理全体にかかる計算時間少なくなったためである．1 プロセッサに割当てられた部材数と並列化効率の関係を表 1 にまとめることで粒度と並列化効率の関係がわかる．

1 プロセッサに割当てた部材数	並列化効率
5 部材未満	50%未満
5 部材	50%以上
10 部材	75%以上
20 部材以上	85%以上

表 1: 粒度と並列化効率

さらに 66 節点 160 部材 (図 6 を参照)，130 節点 320 部材 (図 7 を参照) のように部材数を増やすとより粗粒度になる．

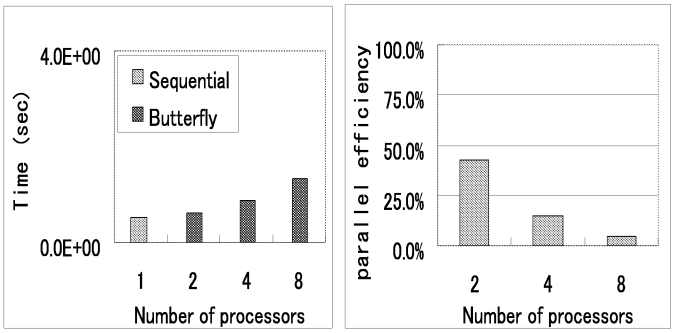

図 3: 6 節点 8 部材 (左) 計算時間 (右) 並列化効率

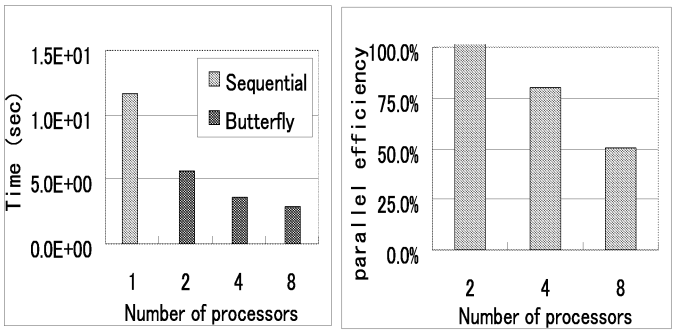

図 4: 18 節点 40 部材 (左) 計算時間 (右) 並列化効率

[1]工学学問に対する近似解を得る為の数値解析技術のこと．

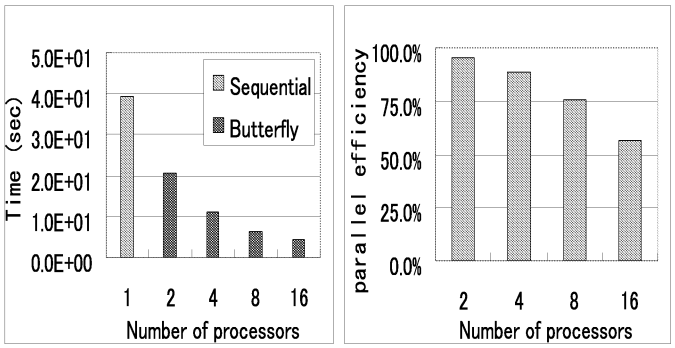

図 5: 34 節点 80 部材 (左) 計算時間 (右) 並列化効率

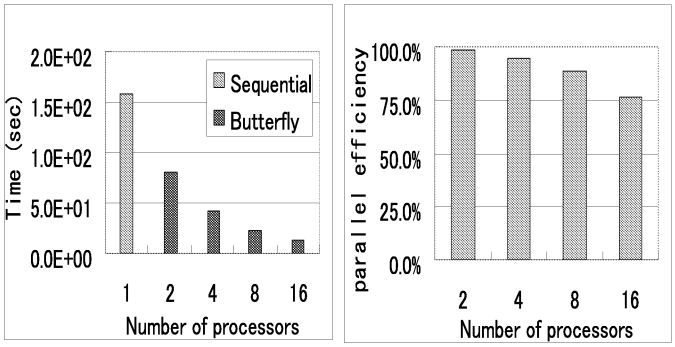

図 6: 66 節点 160 部材 (左) 計算時間 (右) 並列化効率

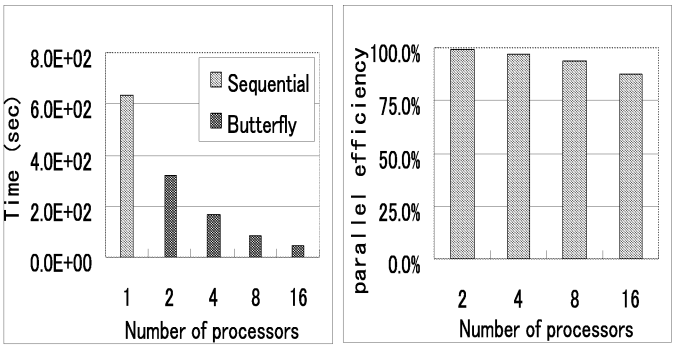

図 7: 130 節点 320 部材 (左) 計算時間 (右) 並列化効率

5 結論

トラス構造物最適化問題に関して，分散最適化手法である DORAR 法は，64 プロセッサの並列計算機 n-cube で並列性の検討が行われ，良好な並列化効率が得られることが報告されている．

本論文では，DORAR 法を 16 台の計算機で構成される PC クラスタに実装することで，並列性の検討を行った．得られた結論は以下の通りである．

- DORAR 法は設計変数が増加しても並列に処理することで総計算速度を十分短縮できる．
- 設計変数が少ない問題に対しては並列化せずとも逐次処理で十分高速なため，並列化する必要性がないと考えられる．

今回の研究では，バタフライ通信が効率よくおこなえるように使用した計算機の台数を 2 のべき乗台に限定して通信をおこなった．しかし，プロセッサ数が 2 のべき乗でないとき，バタフライ通信を実装すると今回同様，通信が効果的におこなわれるとは限らない．

今後の課題として次のようなものが挙げられる．

- プロセッサ数が 2 のべき乗でないとき，もっとも効率の良い通信手段を考察する．プロセッサ数を任意にすることで，利用者が有する最大プロセッサ数を有効に用いることができると考えられる．

参考文献

[1] M.Miki, M.Furuichi, Y.Watanabe, ”Smart Distributed Minimization of the Volume of Discrete Structure”, Proc, AIAA, SDM Conference, pp.2344-2352, 1996

[2] Mitsunori Miki, Tomoyuki Hiroyasu, Taiju Ikeda, ”Parallel Distributed Optimization by Resource Addition and Reduction”, Lecture Notes in Computer Science 1615, Springer, pp.194-205, 1999