

# 資源追加削減法の改良アルゴリズムにおける並列性の検証

## Study of Parallelization in the improved DORAR Method

阿南 英文

Hidefumi Anan

**Abstract:** The DORAR method is a parallel distributed optimization method for the minimization of the total resource of a system with discrete elements. Now, the DORAR method is improved, and its effectiveness is reported. However, its Parallelization is not studied. In this paper, the Parallelization in the improved DORAR method is studied.

### 1 はじめに

離散構造物の最適設計に対して提案された局所ルールに基づく並列分散最適化の手法である資源追加削減法<sup>1,2)</sup>(以下 DORAR 法)は, 各要素でその設計変数である資源に余裕があれば削減し, その後微小資源を追加するというプロセスを繰り返すことで最適解を得る. しかし, 従来のアルゴリズムでは, 局所解に陥る場合があり, これを克服するために, アルゴリズムに改良が加えられた. そして, 工学的な問題に適用した結果, その有効性が確認されている<sup>3)</sup>. しかしながら, DORAR 法の特徴の 1 つである並列性については考察されていない. そこで, 本研究では改良された DORAR 法における並列性の検証を行う.

### 2 資源追加削減法 (DORAR 法) 概略

DORAR 法は, 離散的な要素からなるシステムを対象とし, 各要素を資源と考え, システム全体の資源の和の最小化を目的とする. また, システムには要求される機能が制約条件として課せられている. それらは, 各要素の情報により決定される局所制約条件とシステム全体の情報から決定される全体制約条件である. 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) 上記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) 各要素に一定の微小な資源を追加する (資源追加処理).
- (5) (1) ~ (4) を繰り返すことにより最適解を得る.

### 3 制約条件の確率的選択による DORAR 法の概略と研究目的

従来の DORAR 法のアルゴリズムでは, 局所解に陥る場合があった. それを改善するために各要素が基準とする制約条件を確率的に選択することを基本とするアルゴリズムの改良が提案された. これは, 各要素がランダムに基準とする制約条件を選択し, 選択されなかった制約条件の値の変化によって評価を行うというものである. これをトラス構造物最適化問題に適用した結果, 良好な解に収束した.

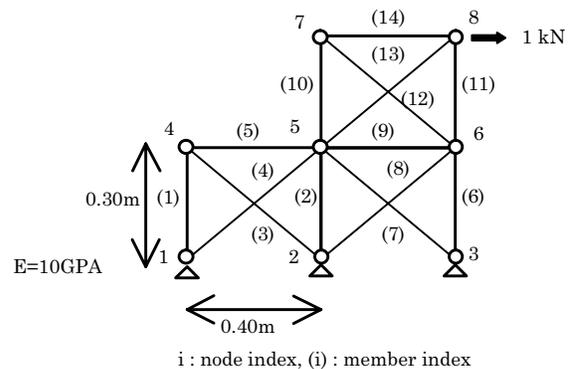


Fig. 1 14 member truss

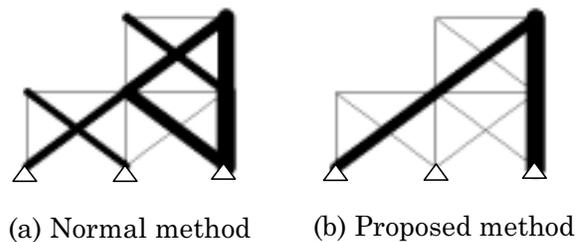


Fig. 2 Converged solutions

しかしながら, 並列性については考察されていない. そこで, 本研究では, この改良アルゴリズムにおける並列性を検証することを目的とする.

Table 1 PC Cluster SPEC

jupiter	
OS	Linux2.2.17
NODE	8+1
CPU	Pentium III(Katmai) 600MHz
Memory	256MB
Network	100Mbps Fast Ethernet
通信ライブラリ	MPICH

Table 2 RS/6000 SP SPEC

RS/6000 SP	
OS	AIX4.2.1
NODE	10
CPU	Power2 Super Chip 160MHz
Memory	1GB
Network	1.2Gbps SP スイッチ
通信ライブラリ	IBM Parallel Environment for AIX-MPI Library

#### 4 DORAR 法における並列化

DORAR 法における並列化とは、分散システムを構成している各資源を分散させることを意味する。すなわち、トラス構造物最適化問題においては、トラス構造物を構成している各部材が資源に相当し、並列化とはその部材を各プロセッサに割り振ることである。通信ライブラリとしては MPI(Message Passing Interface) を用いた。通信手法としては、今回は任意のプロセッサ数で、かつ任意の部材数で並列化できるように Allgather 通信を用いた。Allgather 通信は、Allgather 通信が各送信元プロセスからの受信メッセージの長さが一定なのに対して、受信メッセージの長さを受信バッファ内の位置を、送信元プロセスごとに変えることができる通信である。

#### 5 計算環境

今回用いた計算環境は、PC クラスタ (jupiter) と IBM マシン (RS/6000 SP) である。Table 1 および 2 にその仕様を示す。

#### 6 数値実験

今回の適用問題は、トラス構造物最適化問題である。トラス構造物最適化問題とは、ある節点に負荷を加え、複数の制約条件のもとで最小体積のトラス構造物を設計するというものである。そして今回適用したトラス構造物は Table 3 に示した 3 種類である。ここで、全体的制約

Table 3 適用したトラス構造物

	トラス構造物
	8 節点 14 部材トラス構造物
	16 節点 39 部材トラス構造物
	36 節点 105 部材トラス構造物

条件として変位に関する制約を考え (  $\delta$  では 8,  $\delta$  では 16,  $\delta$  では 36 の節点の変位が 0.01m 以下), 局所制約条件として各部材の引張応力, 圧縮屈服を考える。また負荷荷重として図中に示すように 1[kN] の水平荷重を加えた。

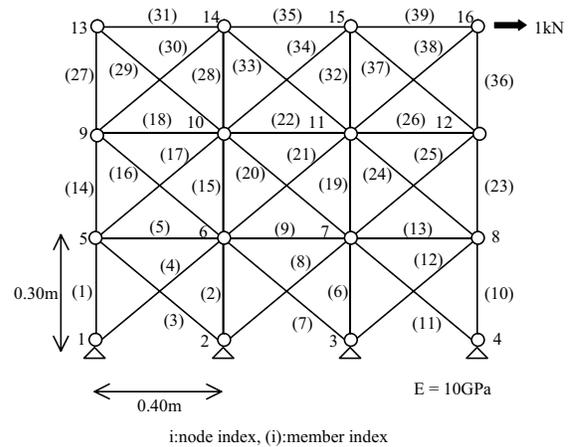


Fig. 3 39 member truss

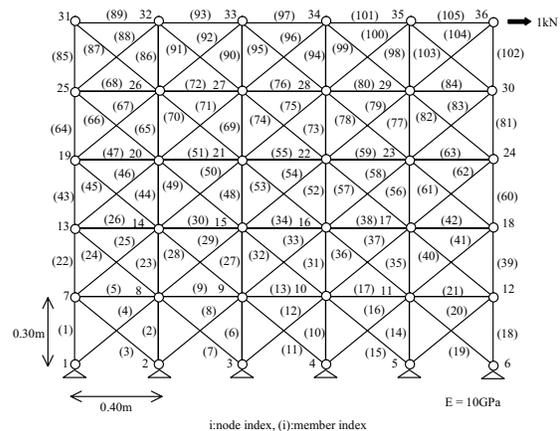


Fig. 4 105 member truss

#### 6.1 実験結果

計測したデータは、各形状に対し、2通りの初期値で、従来の DORAR 法と改良を加えた DORAR 法で、500回の計算回数で行った。計測したデータは通信時間、解

析時間及び総計算時間である。総計算時間には、通信時間と解析時間に加えて、DORAR 法による最適化アルゴリズムの実行時間が含まれている。Jupiter とRS/6000 SP において、105 部材のトラス形状に対して測定した結果を Fig.5 に示す。

使用するプロセッサを増加すると、1 プロセッサの解析負荷が分散するため、解析時間が短縮される。しかし、プロセッサの増加は通信回数を増加させ、通信負荷を大きくする。Jupiter に比べ、RS/6000 SP は高速ネットワークを使用しているため、通信負荷がかからず、プロセッサ数が増えても、総計算時間が減少している。各部材におけるスピードアップのグラフを、従来のアルゴリズムと改良したアルゴリズムにおいて、Jupiter とRS/6000 SP それぞれで求めた。それらを Fig.6, 7 に示す。

まずJupiter とRS/6000 SP どちらにおいても、14 部材のトラス構造物では、並列化を行っても総計算時間に対する通信負荷が高くなり、有意にスピードアップが図られなかった。39 部材のトラス構造物では、高速ネットワークを用いているRS/6000 SP で、そのスピードアップが7 プロセッサまで有意に図られた。一方、105 部材の場合では、Jupiter でも6 プロセッサまでは、有意なスピードアップが図られた。

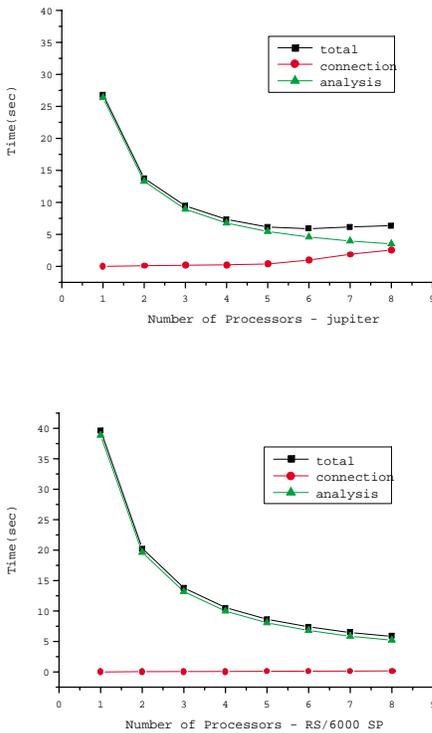


Fig. 5 Relation between time and number of processors at 105 members (Jupiter & RS/6000 SP)

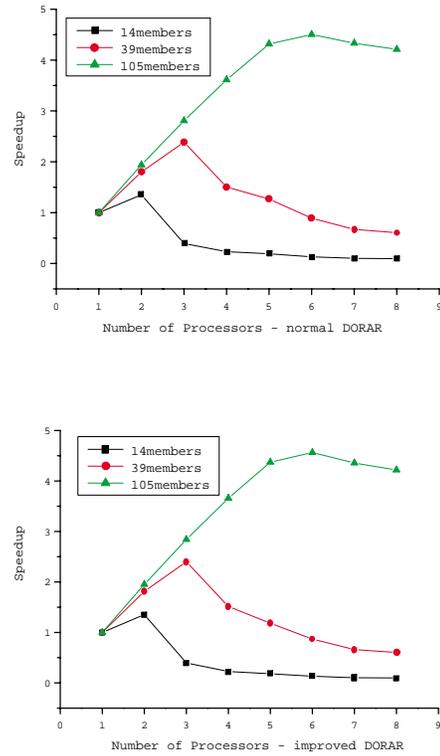


Fig. 6 Speedup by parallel processing (Jupiter)

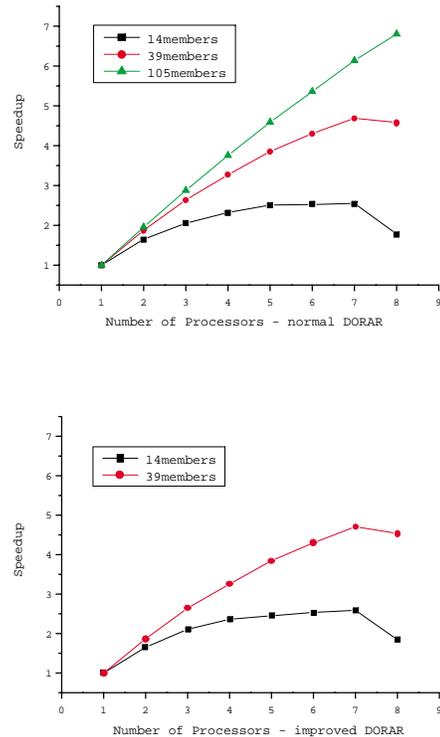


Fig. 7 Speedup by parallel processing (RS/6000 SP)

## 7 結論と今後の課題

本研究では、改良された DORAR 法の並列性の検証を行った。その結果、従来のアルゴリズムと局所解からの脱出を目的とした改良アルゴリズムではほぼ同様な性能を示した。したがって、この改良アルゴリズムでは、並列性が損なわれないことが確認できた。

今後の課題として、さらに大規模なトラス構造物に対してもデータを測定し、考察する。また、通信手法についても考察する必要がある。

### 参考文献

- 1) Mitumori Miki, Tomoyuki Hiroyasu, Taiju Ikeda, 『Parallel Distributed Optimization by Resource Addition and Reduction』, Lecture Notes in Computer Science 1615, Springer, pp.194-205, 1999
- 2) 池田大樹, 『資源追加削減法による離散システムの最適化』, 同志社大学卒業論文, 1998
- 3) 三木光範, 廣安知之, 小林繁, 『制約条件の確率的選択に基づく資源追加削減法の改良』, 日本機会学会第 13 回計算力学講演会講演論文, 2000
- 4) 森俊明, 『資源追加削減法の PC クラスタへの実装と評価』, 同志社大学卒業論文, 1999