日本機械学会論文集(A 編)，66-642, (2000) , pp.411-418

資源の追加と削減に基づく並列分散最適化法*

三木　光範*1，廣安　知之*1

Parallel Distributed Optimization Method based on Resource Addition and Reduction

Mitsunori MIKI*2 and Tomoyuki HIROYASU*2

*2 *Doshisha University, Dept. of Knowledge Engineering and Computer Sciences, Kyo-Tanabe, Kyoto, 610-0321, Japan*

A parallel distributed optimization method for the minimization of the total resource of a system with discrete elements is proposed, and a theoretical and experimental investigations are carried out in this paper. The distributed optimization algorithm consists of two processes, namely the resource reduction process and the resource addition process. In the former process, each element discards its critical resource margin with respect to global and local constraints, while in the latter process, a small amount of resources are added to all the elements. The proposed method is successively applied for optimizing truss structures, and the method is found to be very robust and suitable for parallel processing.

Key Words **:** Optimum Design, Distributed Algorithms, Parallel Processing, Structural Optimization

1.　はじめに

近年, 最適設計の分野でいくつかの新しい手法が注目を集めている．それらは遺伝的アルゴリズム(1)，シミュレーテッドアニーリング(2)，ニューラルネットワーク(3), セルラーオートマトン(4), およびオブジェクト指向に基づく最適化(5)などである．これらの手法は大別すると二つのカテゴリーに分類できる. すなわち，進化的戦略と分散問題解決である．

進化戦略については文献(6)が詳しく論じている．この方法はランダムな変動と適者生存の原理を基礎としている. これらの戦略は非線形性の強い問題や悪定義問題に対しても最適あるいは準最適な解を与えることができる．しかしながら，これらの方法では最適化の過程に関する知識，すなわち，最適でない解をより最適な解に変化させる知識を得ることができない. したがって, 進化戦略は個別の目的を有する特定のシステムを設計する際には有益であり, 最適解そのものに関する知見を得ることはできるが, 最適化のための一般的な知識を蓄積することには貢献しない.

一方，分散問題解決(7)はシステムの要素の局所的な相互作用を基礎としており, このアプローチを用いることにより複雑な問題解決の自動化が行えるほか, ルールの生成と検証を繰り返すことによりシステム全体を最適化する局所ルールを見いだすことができる可能性がある．この場合，このアプローチは最適化と最適解に対する多くの知識をもたらすことになる. こうして得られた知識は他のシステムを最適化する際に有用である．

分散問題解決戦略は離散的システムの最適化に対して次の点から非常に有用と考えられる．それらは，1)非線形性の強い複雑なシステムの最適化，2)並列計算機を用いた最適化，そして3)最適状態を得るための局所的ルールの発見，である．

Miki は離散構造物の最適化のための新しい要素指向, あるいはオブジェクト指向に基づく最適化の方法を提案した(5)．その方法では構造解析と最適化が完全に統合化されており, しかもその構造の各要素における設計変数は局所的なルールを用いて自律的に変化する．しかしながら，この方法は分散的ではあるが並列計算モデルではなかった．なぜなら，各要素における設計変数の変化は要素全体として逐次的に行われており，その順序が最適解に影響するおそれがあった．また，要素間における資源の移動プロセスが存在し，こ

*原稿受付　平成 11 年 ** 月 ** 日

*1 正員，同志社大学(610-0321 京田辺市多田羅都谷 1-3)

れは逐次的処理を基礎としていた．

近年, 科学技術計算の分野での数値解析の大規模化にともない, 並列処理に関する数多くの研究がなされている．しかしながら，大半の研究は線形系の解析，固有値問題，N体問題のシミュレーション，偏微分方程式の解法, 領域分割法による流体解析や固体解析などに属しており, 最適化に関する並列化の研究は相対的に少ない．

連続変数に関する非線形最適化問題に関する並列処理においては文献(8)が包括的なサーベイを行っている．そこでは，最適化における並列化を新しい挑戦と捉え, 既存の逐次的アルゴリズムを並列化するのみならず, 並列処理に適した新しい最適化のアルゴリズムが開発されることが述べられている．その中で，ヘッセ行列の並列処理を行う新しい準Newton法(9), 新しい感度解析不要の方法(10), そしてブロックトランケートされたNewton法(11)などは並列処理を指向して生み出されたアルゴリズムである．

本論文はこのような最適化の新しいアルゴリズムの中で研究がほとんど行われていない局所ルールに基づく並列分散最適化アルゴリズムの構築を目的としたものである．進化戦略とは異なり，この方法では最適化のための局所ルールが明示的に提示される．また，この方法は並列処理に最適であり, ここではそのための具体的手順を示す．

2.　資源の並列分散最適化

2.1　従来の方法　　非線形数理計画法において従来提案されている方法では各設計変数の変化は集中的に制御されている．たとえばSUMT法(12)では，目的関数と制約条件から構成される擬似的な目的関数とその勾配を用いて探索ベクトルを作成し, その方向に１次元探索を行って次の設計点を得る．こうした方法では集中的に管理される探索ベクトルを用いている限り, 各要素が自律的にその要素の設計変数を変化させることはできない．言い換えれば，１次元探索は並列分散的には実施できないということである．

１次元探索を用いない方法として最適性規準法(13),(14)がある．制約条件が単一の場合の設計変数の一般的な変化ルールはKuhn-Tuckerの最適性の条件から得られる次式で表される(15)．

$$x_i^{\mathrm{new}} = x_i^{\mathrm{old}}(\lambda e_i)^{1/\eta},\ e_i = \left(\frac{\partial g}{\partial x_i}\bigg/\frac{\partial f}{\partial x_i}\right) \quad \cdots\cdots (1)$$

ここで x は設計変数，λはラグランジュ乗数，f は目的関数，g は制約条件，そしてηは収束に関するパラメータである．

この式で解が収束するときは設計変数の有効性がラグランジュ乗数の逆数に等しくなったときであり, そのことはすべての設計変数の有効性が等しくなることを要求している．残念ながら，その値は事前には不明であり, このためラグランジュ乗数を推定する方法がいくつか提案されている．しかし，それらはいずれもすべての設計変数の有効性を基に評価する方法であり，並列分散最適化には利用できない．また，制約条件が複数の場合には有効な変化ルールを構成することは難しい．

一方, ヘムステッチング法(16)では１次元探索も行わず, しかも制約条件に対する他の要素の資源の感度も利用せずに設計変数の変更が行える．この方法では解が許容領域に存在するときは目的関数の最急降下方向に移動し, 解が非許容領域に存在するときには制約条件の最急降下方向に移動する．ステップ幅は適当に決定する．

この方法は並列分散最適化法として利用する場合, 二つの問題点がある．一つはステップ幅の決定が難しいことである．目的関数および制約条件の最急降下方向への移動量はユーザが適切な値を設定する必要があり, しかもその値を適切なシーケンスで小さくしなければならない． これに失敗すると解は発散する．

もう一つの問題点は最適性規準法と同様, 複数の制約条件に対応する場合の方法である．この場合，複数の制約条件に対しての最急降下方向をどのように決めるかは難しい問題であり, 提案されているいくつかの方法では複数の制約条件に対するすべての設計変数の感度を基に非許容領域からの脱出方向を決めており, これは並列分散最適化には利用できない．

こうして, 従来提案されている方法はすべて並列分散最適化という観点からは利用が難しい方法と考えられる．

2.2　対象とする問題と提案するアプローチ　　ここでは離散的な要素を有するシステムの最適化問題を考える．各要素は資源を有し，その関数として種々の機能が発現される．目的はシステム全体として必要な資源の最小化であり, それは各要素の資源の和で表される．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．それらは複数の局所的制約条件と複数の全体的制約条件である．設計変数は各要素の資源とする．すなわち，次式で表される問題を考える．

$$
\text{Minimize} \quad R = \sum_{i=1}^{N} R_i
$$
$$
\text{Subject to}
$$
$$
g_{ik} \le 0 \quad (i = 1, \cdots, N;\ k = 1, \cdots, n_i) \quad \cdots\cdots (2)
$$
$$
G_j \le 0 \quad (j = 1, \cdots, m)
$$

ここで，R はシステムの全資源，R_i は要素 i の資源，N は要素数，g_{ik} は要素 i の k 番目の局所制約，G_j は j 番目の全体制約である．

この問題はオブジェクト指向最適化法[5]においても採用された設定条件である．そこでも示されたように，多くの最適化問題はこうした問題に書き換えられる．離散構造の最小重量（体積）問題や最小コスト問題は設計変数をうまく変換すればここに示した問題に変換できる場合が多い．

この問題を分散的に解く．すなわち，システムの各要素が，それ自身に関係する情報と局所的なルールを基にして，それ自身の設計変数，すなわち資源を変化させる．このプロセスの繰返しによりシステム全体の最適化を達成する．

各要素が利用できる情報は，その要素の状態，その要素の局所制約，システムの全体制約，およびそれらの制約のその要素の資源 R_i に関する感度情報とする．すなわち，以下の量が局所的に利用できると仮定する．

$$
\boldsymbol{S}_i
$$
$$
g_{ik} \quad (k = 1, \cdots, n_i),\ G_j \quad (j = 1, \cdots, m) \quad \cdots\cdots (3)
$$
$$
\frac{\partial g_{ik}}{\partial R_i},\ \frac{\partial G_j}{\partial R_i}
$$

ここで，S_i は要素 i の状態を表す．

ここで重要な点は次の二つである．第一に，局所制約と全体制約を分離したことである．ここでは，各要素における制約条件を局所制約条件とし，システム全体としての制約条件を全体制約条件とした．通常，制約条件を局所的と全体的に分離することは行われない．その理由は，システムにおける任意の制約条件はすべての設計変数の関数であり，単独の設計変数のみで決定される制約条件は側面制約として扱うためである．

しかしながら，ここでは次の観点からこうした分離を考えた．すなわち，その要素の内部状態が変化しないと仮定したときに，その要素の資源量の変化によりその制約が満足化できるものをその要素の局所制約と考えた．構造物では部材の応力制約や座屈制約などが局所制約と考えられる．それらは内部状態である部材力が変化しないと仮定したとき，部材の局所的な断面積などを増加させることにより制約は満足化できるからである．

一方，これらの局所制約を除いた制約条件は全体制約とした．構造物の場合には変位や固有振動数などが全体制約と考えられる．全体制約は，ある単独の要素の資源量だけを調節しても容易に満足化できない．

次に重要な点は式(3)で示された量はすべてシステムの全体解析を行って得られるものであり，その解析を行うためには各要素がシステム全体のモデルを保有しているという点である．したがって，各要素は他の要素の資源も観測可能である．ただし，他の要素の資源量を直接利用することはなく，また各制約に対する他の要素資源に関する感度情報を評価することはしないとする．

2.3　提案手法　ここで提案する並列分散最適化の方法は次の手順で示される．

1) 各要素はその局所制約条件を基にその資源余裕を評価する．要素 i の局所制約 g_{ik} に関する局所資源余裕は次式で評価できる．

$$
M_{gik} = \left(g_{ik} \middle/ \frac{\partial g_{ik}}{\partial R_i} \right) \quad \cdots\cdots (4)
$$

2) 各要素は全体制約条件を基にその要素の資源余裕を評価し，それらに責任係数を乗じたものをその要素の全体資源余裕とする．要素 i の全体制約 G_j に関する全体資源余裕は次式で評価できる．

$$
M_{iGj} = \alpha_{ij} \left(G_j \middle/ \frac{\partial G_j}{\partial R_i} \right) \quad \cdots\cdots (5)
$$

ここで責任係数 α_{ij} は式(5)の右辺の分母で表される感度情報を交換しない場合には一定（$1/N$）とするのが妥当である．

3) その要素の局所資源余裕と全体資源余裕の最小値をその要素の臨界資源余裕とし，各要素は臨界資源余裕を削減する．要素 i の臨界資源余裕は次式で表される．

$$
M_i = \mathrm{Min}\,(M_{gi1}, M_{gi2}, \cdots, M_{gin1}, M_{iG1}, M_{iG2}, \cdots, M_{iGm}) \quad \cdots\cdots (6)
$$

4) 微小量の一定資源を各要素に付加する．

5) ステップ 1 から 4 を繰り返す．

この方法は Miki が提案したオブジェクト指向最適化法[5]の資源移動プロセスをなくし，資源削減プロセ

スに重要な変更，すなわちステップ4の資源付加過程を加えたものである．この変更によりこれまで必要であった資源移動プロセスをなくすことができ，並列分散最適化法となった．資源移動プロセスは計算負荷が高いだけでなく，各資源要素にプロセッサを割り当てた場合には並列処理ができない欠点を有している．

全体制約に対する臨界資源余裕を求める際の責任係数はオブジェクト指向に基づく最適化法では一定（1/*N*）としていた．これは，全体制約に対する各要素の責任は平等としたことになる．情報が得られない場合にはこれ以外の考え方はなく，妥当な値と思われる．

2.4　設計解の移動と収束　提案手法を図１に示す資源平面を用いて説明する．システムの要素数を２とし，横軸は要素１の資源を，縦軸は要素２の資源を表す．全体制約を G とし，要素１の局所制約を g_{11} および g_{12}，要素２の局所制約を g_{21} および g_{22} とする．まず最初に，提案手法では局所制約と全体制約を分離し，他の要素の局所制約は評価しない．このことは資源平面上で，要素１から局所制約 g_{21} や g_{22} などは観測されず，要素２から局所制約 g_{11} や g_{12} などは観測されないことになる．

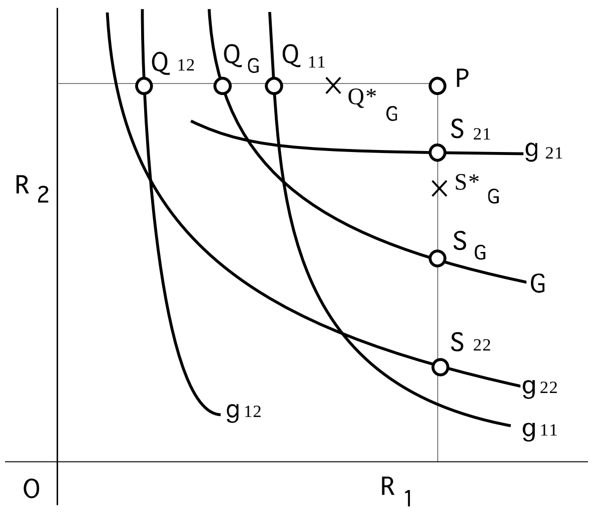

Fig. 1 Critical resource margin

設計点Ｐに関する資源１の局所制約に対する資源余裕は線分 P-Q_{11} および P-Q_{12} で表される．一方，この場合の資源１の全体資源余裕は P-Q_G となる．責任係数を1/2とした場合の全体責任資源余裕は P-Q^*_G で表され，この場合要素１の臨界資源余裕は P-Q^*_G となり，設計点Ｐからは全体制約Ｇのみが観測されていることになる．

一方，設計点Ｐに関する資源２の局所資源余裕は P-S_{21} および P-S_{22} となる．また，資源２の全体資源余裕は P-S_G となり，責任係数を1/2とした場合の全体責任資源余裕は P-S^*_G で表され，この場合要素２の臨界資源余裕は P-S_{21} となり，設計点Ｐからは局所制約 g_{21} のみが観測されていることになる．こうしてある設計点

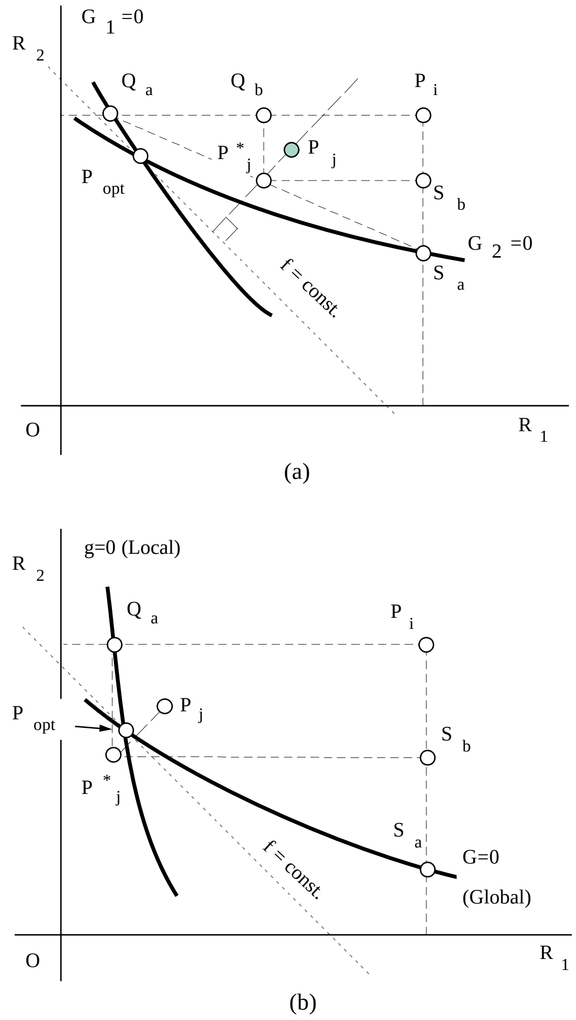

Fig. 2 Movement of design solutions

からは各資源軸方向に観測される単一の制約だけが存在することになる．

図２は２要素の場合の資源平面での設計点の移動過程を表している．ここで現在の設計点は P_i で表されている．観測される制約条件は，図中 (a)では二つの全体制約とし，(b)では全体制約と局所制約とする．(a)の場合，要素１の全体臨界資源余裕は P_i-Q_a で表され，責任係数 1/2 を乗じた臨界資源余裕は P_i-Q_b となる．一方，要素２の全体資源余裕は P_i-S_a で表され，責任係数 1/2 を乗じた臨界資源余裕は P_i-S_b となる．これらの臨界資源余裕を削減すると設計点は P^*_j となる．これは Q_a-S_a の中点でもある．ここで 微小量の一定資源を各要素に付加すると設計点は P_j へ移動する．このとき P^*_j-P_j は▽ R の方向であり，資源等高線に垂直な方向でもある．一方，(b)では，要素１の臨界資源余裕は線分 P_i-Q_a であり，要素２の臨界資源余裕は P_i-S_b となり，設計点 P_i は資源削減により P^*_j に移動し，微小量の一定資源が各要素に付加され，設計点は P_j となる．

設計点の収束状況を図３に示す．解の収束は資源削減ベクトルと資源増加ベクトルが同じ方向で逆向きになったときに生じる．これらの図において全体責任資源余裕から構成される資源削減ベクトル P_i-P_j は▽R(P_j-P_i)の方向と等しい方向となり，設計点は P_i に留まる．

観測される制約条件が単一の全体制約である場合を図３(a)に示す．線分 Q_a-S_a は資源等高線と傾きが等し

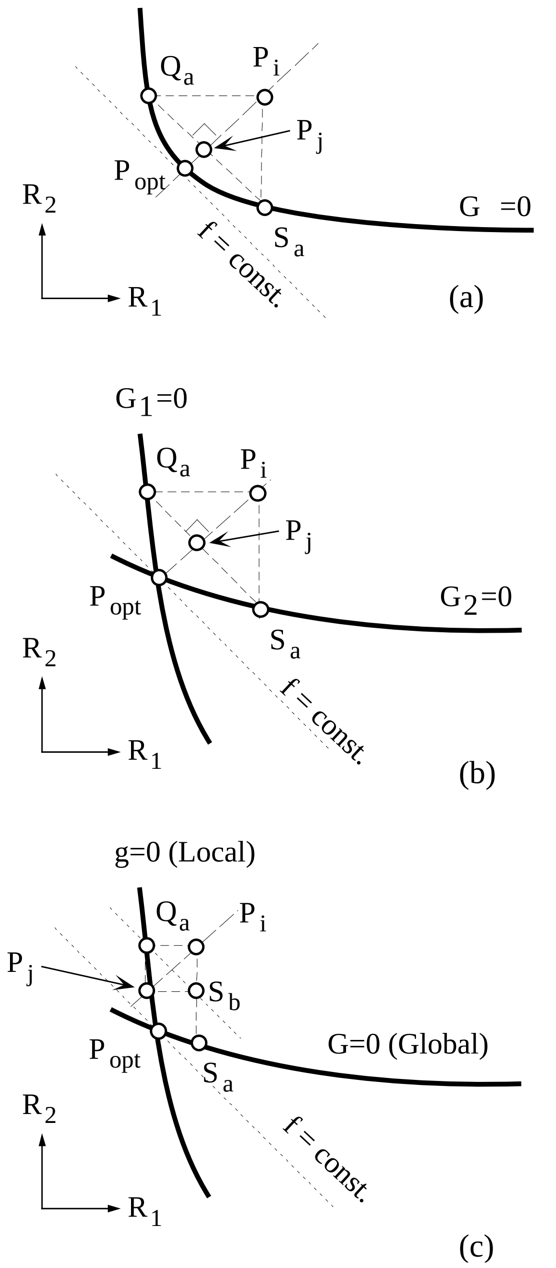

Fig. 3 Convergence of design solutions

くなり, 付加される微小量の一定資源量を減少させると，すなわち P_j-P_i の長さが短くなると，線分 Q_a-S_a は最適解 P_{opt} を通る資源等高線に漸近し，P_i は最適解 P_{opt} に漸近する. 観測される制約条件が複数の全体制約である場合は点 Q_a および S_a が異なる制約条件というだけで，図３(b)に示すように(a)と同様に P_i は最適解 P_{opt} に漸近する.

一方, 観測される制約条件が全体制約および局所制約である場合の設計点の収束状態を図３(c)に示す. この場合 Q_a-S_a が資源等高線に平行になるのではなく，Q_a-S_b が平行になる. ここでも同様で, 微小資源増加ベクトル P_j-P_i を減少させることにより設計点 P_i は P_{opt} に漸近する. 観測される制約条件が複数の局所制約の場合もこれまで示したものと同様に収束解は最適解となる. なお，ここで提案した方法を DORAR（Distributed Optimization by Resource Addition and Reduction）法とよぶ. DORAR 法は基本的に並列化が可能であり，ここではこの方法を並列処理して用いる.

3.　結果と考察

3.1　最適化の対象および使用した計算コード

ここでは離散的構造物の最適化として図４に示す 11 部材のトラス構造の最小体積設計問題を考える. 使用材料は簡単のため縦弾性係数１GPaの線形弾性体とするが，全体の変形は非線形性を考慮する. 目的は体積の最小化であり, したがってこれまで述べてきた資源はここでは体積となる. この問題は同一材料の場合には構造の重量を最小にする問題と等価である. 制約条件は，局所制約条件として部材の引張，圧縮応力および座屈強度を考え, 全体制約条件として一つの節点変位を考える. 部材は中実円形断面とし，設計変数は部材の体積である. 荷重条件は図に示したとおり，節点４と６にそれぞれ５kNづつの水平荷重を負荷した. 応力制約における限界値を引張および圧縮とも 40 MPa とした. また，変位制約として節点６の変位を 0.03m 以下とした. 部材番号１の部材には力が作用せず，無意味な部材であるが, ここではこうした部材も最適化されるかどうかの検証のために付加されている.

使用した並列計算機は分散メモリー型の米国 nCUBE社のnCUBE2Eであり, プロセッサ数は64である. この計算機はプロセッサネットワークがハイパーキューブ結合であり, 用いた言語はプロセッサ間通信のための関数が追加された C++ である.

提案した分散最適化の方法と解析のための方法を組み込んだコードをC++で作成した. このコードではトラスを構成する各部材に一つづつプロセッサを割り当て，各部材が自律分散的にその体積を変化させる. このとき全体制約と関係する局所制約のその部材体積に関する感度の評価はその部材に割り当てられたプロセッサで行う. このため各プロセッサで動作するコードには全体モデルの構造解析コードが含まれ, それに必要な全部材の体積情報はプロセッサ間通信により交換する.

各プロセッサに含まれる構造解析コードは有限要素

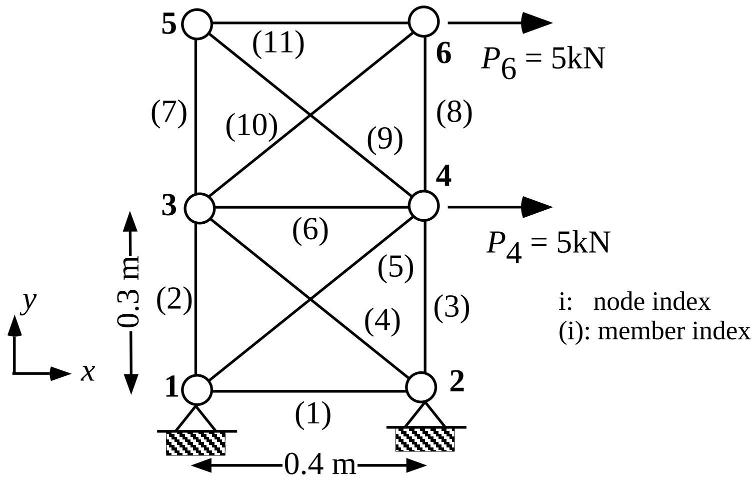

Fig. 4 A 11-member truss structure

法ではなく，Mikiの提案したオブジェクト指向解析[17]に基づくものである．この方法は節点における不平衡力の解消を繰返しながら節点移動を行うもので，任意の非線形挙動を容易に扱うことができる．この構造解析方法は各節点に各プロセッサを割り当てることによって容易に並列化可能であるが，ここでは単一のプロセッサの中で逐次処理によって実施した．その理由は，ここでは各部材ごとにプロセッサを割り当て，並列して異なった構造解析を行う必要があるからである．

分散最適化法における微小資源増加プロセスでは，その時点でのトラス全体積の0.1%を各要素に与えた．

3.2　最適化の結果　乱数を用いて各部材の円形断面の半径を与えた初期値を5種類設定した．これにより部材断面半径は最小1mm，最大50mmのランダムな値となった．これは断面積にして2500倍の差がある設定である．初期値の断面積分布を図5に示す．これを見てもわかるように，初期値は極めて多様であり，このため構造解析において大きな変形が生じることがある．しかし，ここで用いた構造解析手法はオブジェクト指向に基づく分散的方法であるため，こうした大変形解析も問題なく扱える．

提案手法により最適化した場合の全体の体積履歴を図6に示す．全体の体積は初期値では 9.38×10^{-3} から $1.35 \times 10^{-2}\mathrm{m}^3$ の相違があるが，計算の繰返しの初期に急速に減少し，そののち緩慢な変化過程に入ることがわかる．著しい相違を持つ初期値から出発したにも拘わらず良好な収束過程を示したことから，ここで提案した分散最適化手法は極めてロバストな方法であることがわかる．

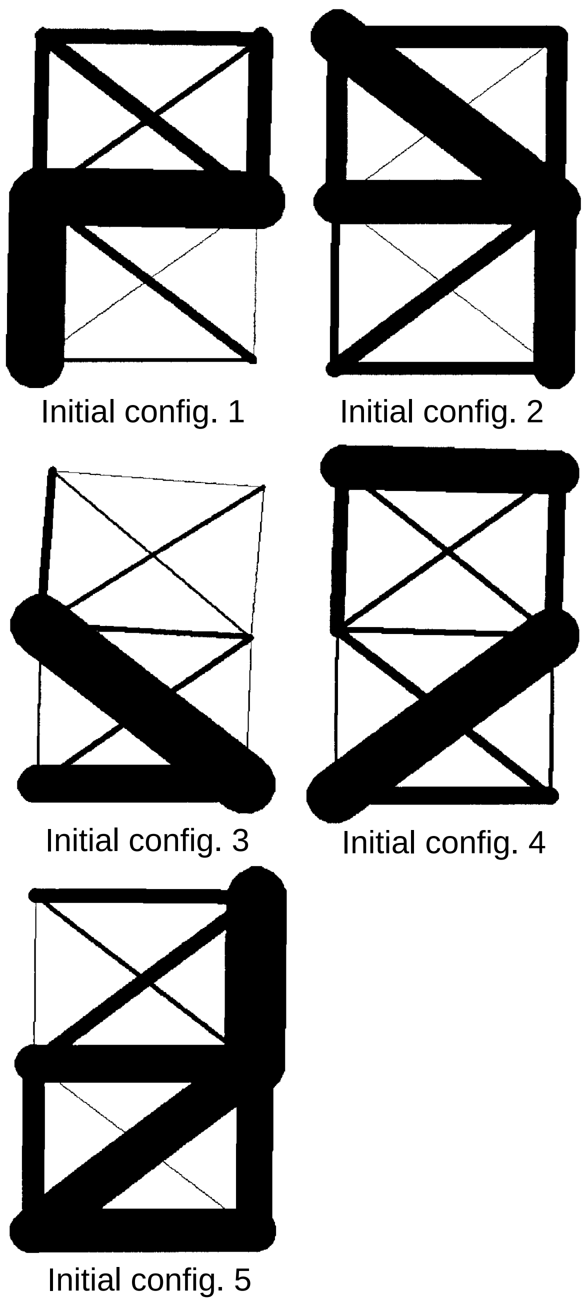

Fig. 5　Initial distributions of sectional areas of the members

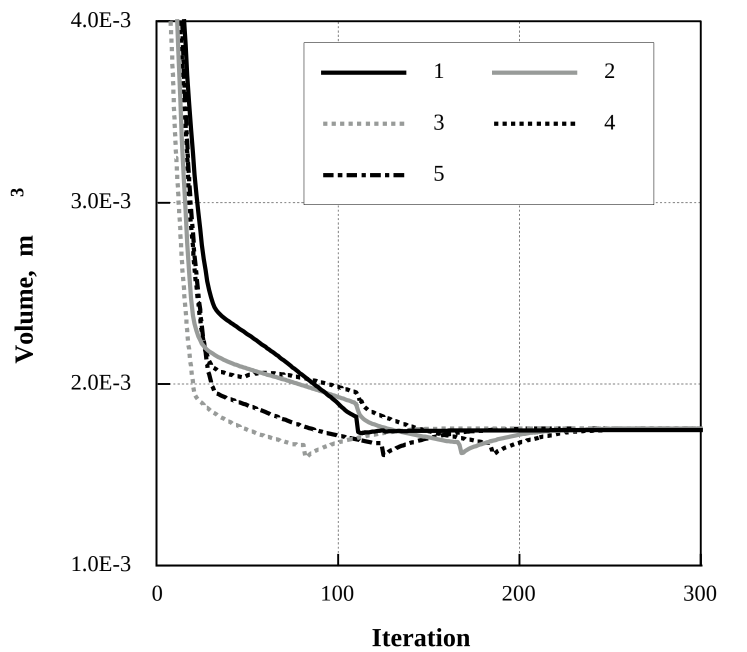

Fig. 6　History of total volume during optimization

5組の初期値を用いて得られた最適解はほぼ同じであった．代表的な解の断面積分布を図7に示す．得られた解の平均的な全体積は $1.69 \times 10^{-3}\mathrm{m}^3$ であった．

この解が有効であることを確認するため遺伝的アルゴリズムを用いて得られた解と比較する．部材の断面積を12ビットのグレイコードで表現し，100個体の単一母集団を用い，交差率0.6の1点交差を行い，突然変異率は遺伝子長の逆数とし，解が安定する500世代

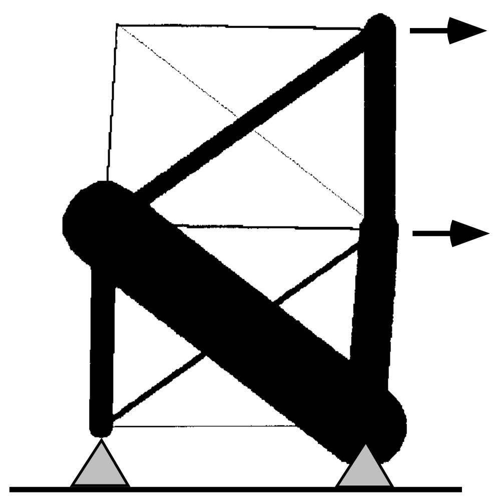

Fig. 7　Distribution of sectional areas of the converged solution

の計算を行った．また，エリート保存戦略を用いた．こうして求められた最適解の平均も同じく 1.69 × 10^{-3}m^3 となった．ただし，遺伝的アルゴリズムによる結果のばらつきはDORAR法による結果のばらつきよりかなり大きかった．この結果より，ここで提案したDORAR 法は有効であることがわかった．

3.3　考　察　DORAR法と遺伝的アルゴリズムの計算量を比較する．DORAR 法では 11 部材× 300 回の繰返しで3300回の解析を行ったのに対して, 遺伝的アルゴリズムでは 100 個体× 500 世代で 50000 回の解析を行った．こうして DORAR 法は１回の試行で計算量は約15分の１となることがわかる．遺伝的アルゴリズムでは初期の早熟により最適解の探索能力が落ちるため, 少なくとも5ないし10回程度の試行が必要となる．一方，DORAR 法では初期値を３ないし５程度変えて行えば十分である．また，遺伝的アルゴリズムでは良好な解が得られるまでに, 適合度関数における目的関数や制約条件の重みに関してのパラメータチューニングが必要となり，何回も試行を繰り返す必要がある．一方，DORAR 法ではパラメータ・チューニングは単に資源追加プロセスにおける微小資源追加量だけであり，このチューニングはほとんど必要ない．これらの考察から，DORAR 法は遺伝的アルゴリズムと比較して計算量は少なく見積もっても 10 分の 1，多く見積もれば100 分の１となる．これより，DORAR 法は計算量の点から考えても優れた方法といえる．

次に DORAR 法のロバスト性について考察する．図５および６からわかるように, この方法は任意の初期値から収束することがわかる．一方，感度解析を用いる通常の非線形数理計画法としてここでは２次計画法を用いてこの問題を解いたところ, 多くの場合に解が発散し，結果が得られなかった．これより，DORAR 法は極めてロバストな方法であり, 任意の初期値を用いて最適解を得ることができ, 手法の利用においては特別の熟練を必要としない優れた方法であることがわかった．

一方，DORAR 法はここで示したように，並列化のために考案されたアルゴリズムであり, 並列処理に適した方法である．もちろん，遺伝的アルゴリズムも並列処理に適しているが, 上で述べたように計算量にかなりの差があるため，並列化してもこの差は残る．なお，ここでは DORAR 法を１プロセッサでは行っていないため，DORAR 法自体の並列化効率は得られていない．これについては今後の課題としたい．

4.　結　論

本論文では離散構造物の最適化問題を分散的に解くための新しい方法として並列分散最適化法を提案した．得られた結論は以下の通りである．

1) 設計点を制約の境界に移動するプロセスと微小資源を付加して制約境界に対して余裕を与えるプロセスの繰返しからなる DORAR（Distributed Optimization by Resource Addition and Reduction）法は有効であり，かつ極めてロバストな方法であることがわかった．また，人為的に与えるパラメータは微小追加資源量のみであり，これは容易に設定可能である．
2) 制約条件を局所制約と全体制約に分離し, 異なった扱いをすることにより分散的な最適化が可能となった．
3) DORAR法では追加する微小資源のため最終的に不要となる資源が消滅せず, 完全な最適解には収束しない．しかしながらその微小資源を十分小さく取ることにより最適解に近づけることができる．
4) DORAR 法は並列処理に適している．

参考文献

(1) Goldberg, D.E., “ Genetic algorithms in Search, Optimization, and Machine Learning, ”Addison-Wesley, Reading, (1989).

(2) Atiqullah, M.M. and Rao, S.S., “ Parallel Processing in Optimal Structural Design Using Simulated Annealing, ” AIAA Jounal, Vol. 33, No. 12 , (1995), pp. 2386-2392.

(3) McMurtry, "Adaptive Optimization Procedures," in Mendel, J.M., editor, A Prelude to Neural Networks: Adaptive and Learning systems, PTR Prentice Hall, New Jersy, (1994), pp. 243-286.

(4) Najim, K. and Pznyak, A.S., "Learning Automata," Elsevier Science, New York, (1994), p.120.

(5) Miki, M., "Object-Oriented Optimization of Discrete Structures," AIAA Journal, Vol. 33, No. 10, (1995), pp.1940-1945.

(6) Schwefel, H.P., "Evolution and Optimum Seeking," John Wiley & Sons, New York, (1995).

(7) Lesser, V.R. and Corkill, D.D., "Distributed Problem Solving," in Shapiro, S. C., editor, Encyclopedia of Artificial Intelligence, John Wiley & Sons, New York, (1987), pp. 245-251.

(8) Schnabel, R.B., “ A View of the Limitations, Opportunities, and Challenges in Parallel Nonlinear

Optimization, ” Parallel Computing, Vol. 21, (1995), pp.875-905.

(9) Laarhoven, P.J., “ Parallel Variable Metric Methods for Unconstrained Optimization, ” Math. Programmimg, (1985), pp. 68-81.

(10) Dennis Jr., J.E. and Torczon, V., “ Direct Search Method Methods on Parallel Computers, ”SIAM J. Optimization, Vol. 1, (1991), pp. 448-474.

(11) Nash, A.G. and Sofer, A., “ Block Truncated-Newton Methods for Parallel Optimization, ” Math. Programming, Vol. 45, (1989), pp. 529-546.

(12) Vanderplassts, G. N., "Numerical Optimization Techniques for Engineering Design," McGraw-Hill, New York, (1984), pp. 121.

(13) Venkayya, V.B., "Optimality Criteria: A Basis for Multidisciplinary Optimization," Computational Mechanics, Vol. 5, (1989), pp.1-21.

(14) Khot, N.S., "Algorithms Based on Optimality Criteria to Design Minimum Weight Structures," Engineering Optimization, Vol. 5, (1981), pp.73-90.

(15) Haftka, R.T. and Gurdal, Z., “ Elements of Structural Optimization, ”Kluwer Academic, Dordrecht, (1992), pp. 371.

(16) Roberts, S.M. and Lyvers, H.I.,“ The Gradient Method in Process Control, ”Ind. Eng. Chem., Vol. 53, (1961) , pp. 877-882.

(17) Miki, M., “ Object-Oriented Approach to Modeling and Analysis of Truss Structures, ”AIAA Journal., Vol. 33, No. 2, (1994), pp. 348-354.