Application of DORAR Method to Nonlinear Optimization Problems with Many Variables

Mitsunori Miki*,Tomoyuki Hiroyasu* Shin Oguri**, Taiju Ikeda**

(Received July 16, 1999)

A new distributed optimization method called DORAR has been proposed with the concept that each element in a discrete system have being autonomous behavior only with its local information, and the whole system approaches to the best condition as a result. The distributed optimization algorithm consists of two processes, the resource reduction process and the resource addition process. The DORAR method has been successively applied for optimizing truss structures. However, this method has not been applied to nonlinear optimization problems with many variables. This paper describes the problem for applying with many variables, and proposed a new rule which is not affected by the increase of the number of design variables, the termination criterion and the amount of the resource used in resource addition processing. Also, this paper deal with constraints that have not been considered before. The proposed algorithm is found to be effective for nonlinear optimization problems.

Key words ： optimum design, distributed algorithm, parallel processing, DORAR method

キーワード ： 最適設計，分散アルゴリズム，並列処理，資源追加削減法

多変数非線形最適化問題への資源追加削減法の適用

三木光範・廣安知之・小栗伸・池田大樹

1. 緒　　言

非線形最適化問題においては，伝統的に微分積分学に基づくアプローチが大きな成果をあげてきた．特に，最小化すべき目的関数を逐次２次近似して反復を行うことにより早い収束性をもたせるNewton法の概念と，目的関数を単調に減少させる降下法の概念をいかにして両立させるかという点に力がそそがれ，その結果，準Newton法，逐次２次計画法，一般化簡約勾配法など，非常に優れた収束性をもつ手法が開発されている．しかしながら近年，問題の複雑化，大規模化に伴い，解を見いだすのに膨大な計算が必要になり，計算負荷が高くなるという問題が生じてきている．現在ではそれらの問題に対してアルゴリズムの改良のみならず，既存の逐次的アルゴリズムの並列化，並列処理に適した新しい最適化アルゴリズムの開発などが課題となっている[1)]．このような観点から，三木により提案された資源追加削減法[2)]（以下DORAR法）は，並列計算機のための分散最適化コードであり，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，その後微少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という，並列処理に適した新しい最適化手法である．現在のところ電気回路最適化問題，トラス

* Department of Knowledge Engineering and Computer Science,Doshisha University, kyoto
Telephone:+81-774-65-6434, Fax:+82-774-65-6796, E-mail:mmiki@mail.doshisha.ac.jp
** Department of Knowledge Engineering and Computer Science,Doshisha University, kyoto
Telephone:+81-774-65-6716, E-mail:shin@mikilab.doshisha.ac.jp

構造物最適化問題 [2,3,5,6] といった資源最小化問題に対してその有効性が確認されているものの，対象問題は比較的小規模な問題に限定されていた．

本研究では，資源追加削減法を多変数の非線形最適化問題に適用し，設計変数の増加に伴う問題点を提示する．従来のアルゴリズムでは資源追加処理において設計変数の数に関わらず一定量を微少追加資源量として付加していたが，設計変数の増加に伴う解の発散が見られた．そこで，設計変数の増加に影響を受けない微少追加資源の決定方法，および終了条件を提案し，資源追加削減法のアルゴリズムに改良を加える．また，従来，資源追加削減法では対象問題を資源最小化問題に限定してきたため，制約条件として下限値を与えるもののみを対象とし，制約として上限値を与えるものは考慮されていなかった．そこで，これらの制約条件についても検討し，新な資源余裕の評価方法を提示する．これらの改良を加えた資源追加削減法を多変数の非線形最適化問題に適用し，数値実験を行った結果，設計変数の増加に影響を受けることなく良好な解を得ることができた．このことからも，改良したアルゴリズムは資源追加削減法を多変数の問題に適用する上で有効であるといえる．

2. 資源追加削減法の概略

2.1 対象問題

資源追加削減法は対象とする問題を離散的な要素の最適化問題である資源最小化問題に限定する．各要素は資源を有し，その関数として種々の機能が実現される．目的はシステム全体の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制約条件である．これらの資源最小化問題は以下のように定式化できる．

$$
\begin{aligned}
&Minimize\ R=\sum_{i=1}^{n} R_i \qquad (1)\\
&Subject \quad g_{ik}\le 0\ (i=1,\cdots,n;\ k=1,\cdots,n_i)\\
&\qquad\quad G_j\le 0\ (j=1,\cdots,m)
\end{aligned}
$$

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．そしてこの問題を分散的に解く．つまり，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，それ自身の資源を変化させる．このプロセスの繰り返しによりシステム全体の最適化を達成する．要素 i が局所的に利用できる情報は式 (2) で表される．

$$
\begin{aligned}
&g_{ik} \quad (k=1,\cdots,n_i)\\
&G_j \quad (j=1,\cdots,m) \qquad (2)\\
&\frac{\partial g_{ik}}{\partial R_i} \quad ,\frac{\partial G_j}{\partial R_i}
\end{aligned}
$$

ここで R_{ik} , G_j , $\partial g_{ik}/\partial R_i$，$\partial G_j/\partial R_i$ はそれぞれの要素の局所制約，システムの全体制約，およびそれらの制約のその資源 R_i に関する感度情報である．ここで感度が高い要素とは，その制約条件に対する影響が大きいことを意味し，システムの中でその制約条件を満たすために必要度が高いことを意味する．このように資源追加削減法では対象問題を式 (1) のような資源最小化問題に変換し，最適化を行う．実際多くの最適化問題はこうした問題に書き換えられる．離散構造物の最小重量問題や最小コスト問題も設計変数を変換すれば、ここに示した資源最小化問題に変換できる場合が多い．

2.2 資源追加削減法のアルゴリズム

資源追加削減法の最適化の原理は単純で，各要素はその設計変数である資源に余裕があれば削減し，その後, 微少資源を追加するというものである．以下にアルゴリズムを示す．

1) 各要素はその要素に関する制約条件を基にその要素の資源余裕を評価する．要素の資源を R_i とすると，要素 i の k 番目の局所制約条件 g_{ik} に関する資源余裕は式 (3) で与えられる．

$$
R_{mi}^{g_{ik}} = \frac{g_{ik}}{\left(\dfrac{\partial g_{ik}}{\partial R_i}\right)} \qquad (3)
$$

2) 各要素がシステムに与えられた全体制約条件に関する資源余裕を見積もる．資源 R_i の j 番目の全体制約条件 G に関する資源余裕は式 (4) で与えられる．

$$
R_{mi}^{G_j} = \alpha\frac{G_j}{\left(\dfrac{\partial G_j}{\partial R_i}\right)} \qquad (4)
$$

α は責任係数と呼ばれる係数で，全体制約条件を達成するための，要素の責任の重要さ，つまり全体制約に対する要素の重要さを表している．一般的には，要素数の逆数 (1/n) を用いる．

3) 各要素における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 の臨界資源余裕は式 (5) で表すことができる．この処理を資源削減処理と呼ぶ．

$$R_{mi}^{(k)} \quad = \quad Min(R_{mi}^{g_{ik}}, R_{mi}^{G_j}) \qquad (5)$$

4) 各要素に一定の微少資源を追加する．(この処理を資源追加処理と呼ぶ) この微少追加資源量 ΔR は次式により決定される．

$$\Delta R \quad = \quad r_{add} \times \sum_{i=1}^{n} R_i \qquad (6)$$

r_{add}(微少追加資源割合) とは追加する資源量の総資源に対する割合であり，資源追加削減法における唯一のパラメータである．

5) 手順 (1) から (3) を繰り返すことによって最適解を得る．

このように，資源追加削減法では資源削減量は式 (4) より，微少資源追加量は式 (6) より決定される．ここで，総資源量は要素数によって増加することから，資源追加量は要素数に比例するといえる．そのため，要素数が増えると資源削減量は減少し，微少資源追加量が増加するため，解が発散する可能性もある．従来，資源追加削減法は比較的小規模な問題に適用されていたため，このような問題については考慮されていなかった．そこで，本研究では特にこの問題に着目し多変数の問題に資源追加削減法を適用し，これらの問題を検討する．また，終了条件については，三木 [3] による論文では特に考慮されていなかった．ここでは，ある設計点から 1 ステップ後の設計点における目的関数値の差を計算し，その値が一定値以下となった時を終了と考えることとする．

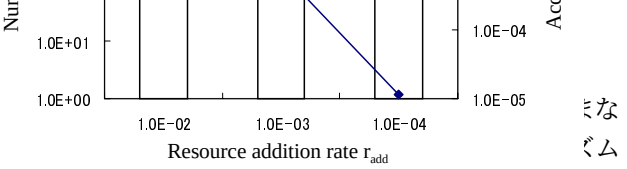
3. 非線形最適化問題への適用

3.1 対象とする問題

本研究では，資源追加削減法を多変数の問題に適用することを目的としている．そこで対象問題として，式（7）に示す非線形最適化問題を考える．

$$\begin{aligned} &Minimize \quad R = R_1 + R_2 + \cdots R_n \\ &Subject \quad \sqrt{R_1} + \sqrt{\frac{R_2}{2}} + \cdots + \sqrt{\frac{R_n}{n}} \geq 0 \qquad (7) \\ &\qquad R_1 \geq 0, \cdots, R_n \geq 0 \end{aligned}$$

この問題における n は設計変数の数であり，容易に設計変数の数を変更できる．この問題に資源追加削減



に変更を加える．Fig. 1 に n=2 における対象問題を示す．

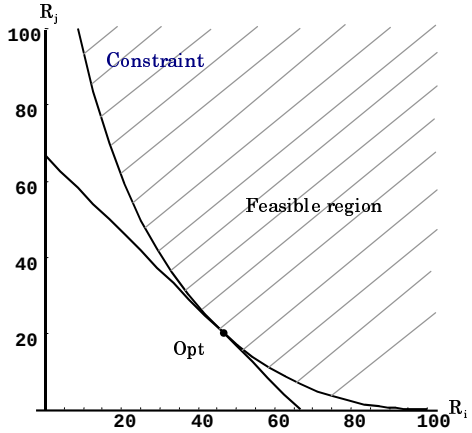


Fig. 1 Nonlinear problem (n=2).

3.2 微少追加資源量の決定方法

資源追加削減法では資源削減処理の後，各要素に一定の微少な資源 (以下 ΔR) を追加する資源追加処理を行う．この微少追加資源量 ΔR は式 (6) より求めることができる．このように微少追加資源量は r_{add} によって決定される．つまり r_{add} を大きくするほど微少追加資源量も大きくなる．ここでは，設計変数が増加した場合微少追加資源量をどのような値に設定するのが望ましいか検討する．Fig. 2 では設計変数の数を 10 に設定し，微少追加資源量の大きさを変化させ，微少追加資源量が解の精度，計算回数にどのように影響するかを調べる．

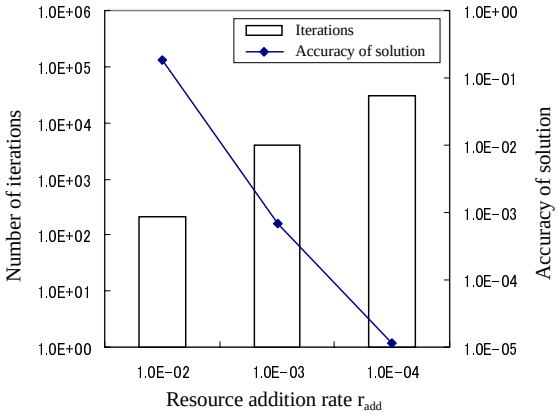
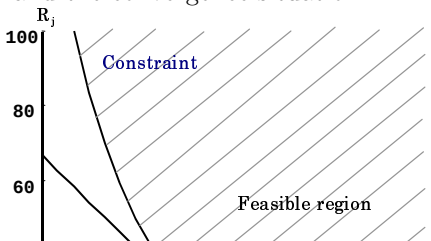


Fig. 2 Relation between little addition resource and the convergence situation.



	Number of Variable	2	5	10	20	30	40
Optimum solution	ΔR=0.1ΣR	67.43	Not converged	Not converged	Not converged	Not converged	Not converged
	ΔR=0.01ΣR	66.67	274.9	1013	Not converged	Not converged	Not converged
	ΔR=0.001ΣR	66.66	273.7	853.4	2817	6631	Not converged
	Solution by DOT	66.66	273.7	853.5	2779	5632	9364

また，設計変数の数と微少追加資源量の大きさ
化させたときの DORAR 法で得られる解と，最
の比較を Table 1 に示す．なお，最適解は非線形
化問題を解く場合，世界的に標準とされている
derplaats Research & Development 社の DOT よ
られた解を用いる．

	Number of Variable	2	5	10	20	30	40
Optimum solution	ΔR=0.1ΣR	67.43	Not converged	Not converged	Not converged	Not converged	Not converged
	ΔR=0.01ΣR	66.67	274.9	1013	Not converged	Not converged	Not converged
	ΔR=0.001ΣR	66.66	273.7	853.4	2817	6631	Not converged
	Solution by DOT	66.66	273.7	853.5	2779	5632	9364

まず，設計変数の数を 10 に設定した場合につ
考察する．r_{add} を 10^{-2} に設定した場合，つまり微少
追加資源量が大きいと，計算回数は少ないが，解の精
度が著しく悪いことが分かる．また，r_{add} を 10^{-4} と
微少追加資源量を小さくした場合は，解の精度は向上
するものの，計算回数が増加してしまう．このことか
らも，設計変数の数を 10 に設定した場合，r_{add} の値
は 10^{-3} が理想的であると考えられる．

従来の資源追加削減法による最適化においても，微
少追加資源量は総資源量の 0.1 %，つまり r_{add} の値は
10^{-3} と設定されていた．しかし，r_{add} を 10^{-3} に設定
した場合，設計変数の数が 30 を越えると著しく解の
精度が劣化し，さらに設計変数の数が増加すると，初
期点によっては Table 1 に示すように発散する可能性
も生じる．そのため，設計変数が 30 を越えると，それに伴い r_{add} の値はさらに低く設定する必要がある．
設計変数 30 においては r_{add} の値は 10^{-4} の方が好ましいと考えられる．このように従来の微少追加資源量の決定は，設計変数の数の増加に応じて，微少追加資源量の値を下げる必要があり，ある程度の試行が必要となってくる．

また Table 1 から，従来の手法では設計変数増加に伴う解の精度の低下，計算回数の増加といった問題点が確認できる．そこで次節ではこれらの設計変数増加に伴う問題点について検証する．

次に解の発散について検討する．この原因としては，資源追加処理における微少追加資源量が考えられる．そこで，設計変数 100 における総資源量の推移を見る．

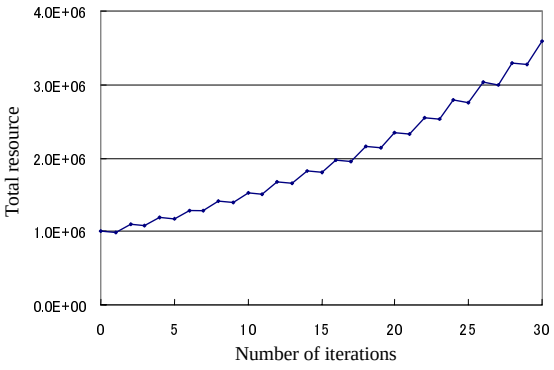


Fig. 3 The change of the total resource when using the conventional addition resource rule.

Fig. 3 から資源削減処理で削減される以上に資源追加処理で追加され，その結果，徐々に総資源量が増加し発散していることが確認できる．このことから，設計変数の増加に伴う発散は，資源削減量が設計変数の数の増加により減少するのに対し，微少追加資源量は設計変数の数に関わらず一律的に決定することが発散の原因であると考えられる．次節では，これら設計変数の増加に伴う問題点を改善するために，新たな微少資源追加量，終了条件の決定方法について検討する．

4. 設計変数の増加に伴う問題点への対策

4.1 設計変数の増加に影響を受けない終了条件の検討

前節で述べた設計変数の増加に伴う解の精度の低下を改善するために，改めて終了条件 (Termination criterion) について検討する．本論文では終了条件として次のような 3 つの方法を設定した．終了条件 1 は式 (9) に示すように目的関数の値の変化率が 10^{-5} 以下とした．終了条件 2,3 はそれぞれ目的関数の変化量が 10^{-5} 以下，各資源の変化量が 10^{-5} 以下とし，式 (10), 式 (11) で表すことができる．なお，$R_{i(j)}$ は j 番ステップの i 番目の資源量である．

$$\frac{\sum_{i=1}^{n}(R_{i(j+1)} - R_{i(j)})}{\sum_{i=1}^{n} R_{i(j+1)}} \leq 10^{-5} \tag{8}$$

$$\sum_{i=1}^{n}(R_{i(j+1)} - R_{i(j)}) \leq 10^{-5} \tag{9}$$

$$R_{i(j+1)} - R_{i(j)} \leq 10^{-5} \tag{10}$$

対象問題は式 (7) とし，設計変数の数を変化させ解の精度，計算回数を比較する. それぞれの終了条件と計算回数の関係を Fig. 4 に, 終了条件と解の精度の関係を Fig. 5 に示す.

Fig. 4,5 からも，終了条件 1 は設計変数が増加しても計算回数はほぼ一定だが，解の精度は著しく劣化していることが分かる．これは，設計変数が増加すると，良好な最適解に収束する前に終了しているためだと考えられる．また終了条件 2 は，設計変数の増加に伴い計算回数は増加するものの，逆に解の精度は上がっている．そこで，まず終了条件 2 について検討する．終了条件 2 は式 (10) で表される．式からも分かるように設計変数の数 n が増えると，各資源の和である目的関数の値が大きくなるのに対し，終了条件は常に目的関数の値の変化量が一定値以下となっている．そのため設計変数が増加するほど終了条件は厳しくなり，結果的により良好な解に収束している．このように，終了条件 1，2 は設計変数の増加に大きな影響を受けている．一方，終了条件 3 は終了条件 1，2 に比べ計算回数も少なく，解の精度も良い．これは式 (11) からも分かるように終了条件 3 には設計変数の数は影響していないため設計変数が増加しても解の精度，計算回数も他の条件ほど影響を受けてないと考えることができる．これらのことからも，提案した 3 つの終了条件の中で終了条件 3 がもっとも設計変数に影響されにくい終了条件であると言える．

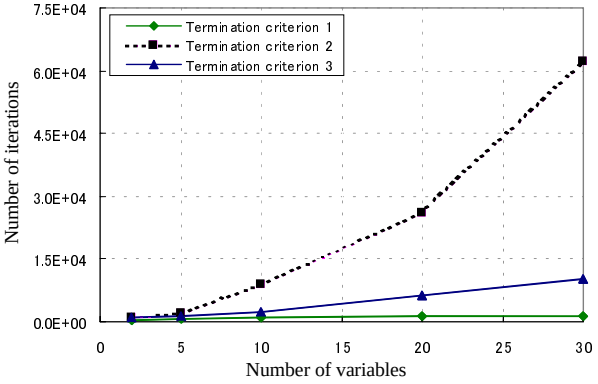


Fig. 4 Relation between number of variable and iteration.

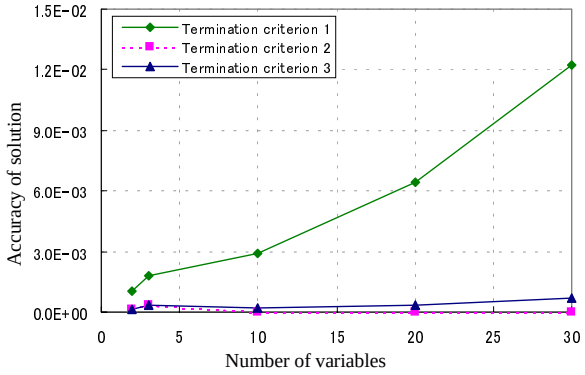


Fig. 5 Relation between number of variable and accuracy of solution.

4.2 設計変数の増加に影響を受けない微少追加資源量の提案

3.2 節で述べた設計変数の増加に伴う発散を改善するため，新たな微少追加資源量の決定方法について検討する．従来の資源追加削減法のアルゴリズムでは，資源削減量は資源余裕・(1/n)，資源追加量は総資源量・r_{add} から決定されていた．この場合，各要素における資源削減量は設計変数 n の増加に伴い減少する．一方，微少追加資源量は設計変数の増加に伴い，それ自身も増加してしまう．これは設計変数が増加すると総資源量も増加するためである．このように従来の方法では設計変数の増加に伴い，資源削減量は減少，資源追加量は増加する．その結果，r_{add} の値によっては常に資源削減量より追加量が大きくなり発散してしまう．このような問題点は，従来の資源追加削減法のアルゴリズムが設計変数の増加に伴って生じる問題点を考慮しなかったためである．そこで新たな r_{add} 設定方法を提案する．

$$Rule1 \quad \Delta R = r_{add} \times \sum_{i=1}^{n} R_i \tag{11}$$

$$Rule2 \quad \Delta R = \frac{1}{n} \times r_{add} \times \sum_{i=1}^{n} R_i \tag{12}$$

$$Rule3 \quad \Delta R = \frac{1}{n^2} \times r_{add} \times \sum_{i=1}^{n} R_i \tag{13}$$

従来の微少追加資源量は Rule 1 によって与えられていた．新たに提案する方法は，Rule 2 と Rule 3 である．Rule 2 は設計変数の影響を受けないように，従来の方法に設計変数の数である n の逆数を掛けている．ここでの $(1/n)\cdot\sum_{i=1}^{n} R_i$ は各資源の平均であり，これはほぼ各要素の資源量である．

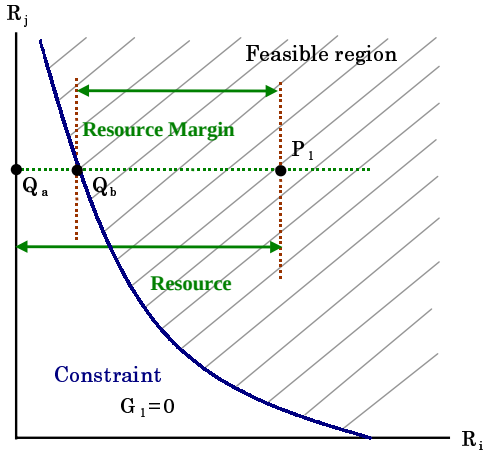


Fig. 6 Estimation way of resource margin.

これを Fig. 6 のような要素数 2 の資源平面で考える．要素 i において設計点 P_1 の資源余裕は P_1-Q_b，資源量は P_1-Q_a となる．ここでの P_1-Q_a は総資源量の平均値と考えることが出来る．つまり Rule 2 では，資源削減量は P_1-$Q_b\cdot(1/n)$，資源追加量は P_1-$Q_a\cdot r_{add}$ より決定される．その結果，従来の方法に比べ設計変数の増加に伴う発散の可能性を減少させることが期待される．しかしながら，Rule 2 の方法を用いても $(1/n) < r_{add}$ においては，資源削減量より資源追加量が大きくなり発散する可能性がある．そのため常に r_{add} の値は，設計変数の数の逆数値 $(1/n)$ よりも低く設定する必要がある．そこで Rule 3 では資源削減量と同じく，資源量 (P_1-Qa) に設計変数の数の逆数値 $(1/n)$ を掛け，その値に微少追加資源割合である r_{add} を掛ける．こうすることで設計変数が増加しても，資源削減量より資源追加量が大きくなることはなくなり，設計変数の増加に伴う発散を防ぐことが期待できる．

4.3 最

新たな […] る．終了条件に […] ここでは，設 […] レを適用した． […] 資源量の比較を […] に示す．

設計変 […] も総資源量は緩 […] を越えると Rule1 では発散した．一方，Fig. 7 のように Rule 3 では，500 変数においても発散することなく収束している．これは，Rule 3 が資源削減量，追加量とも設計変数の数に影響を受けないためである．

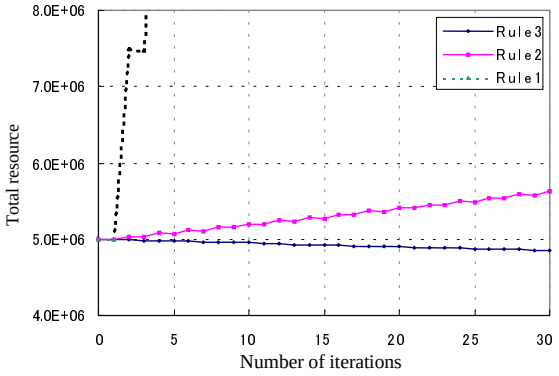


Fig. 7 Change of total resource when using a proposed addition resource rule.

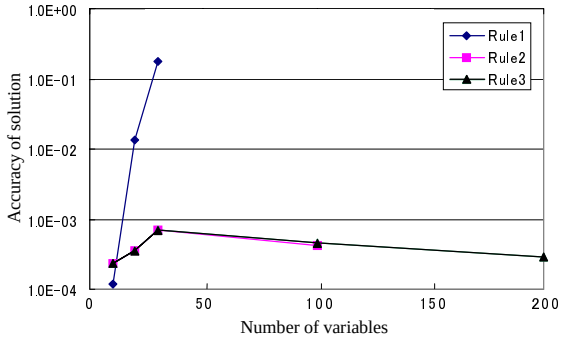
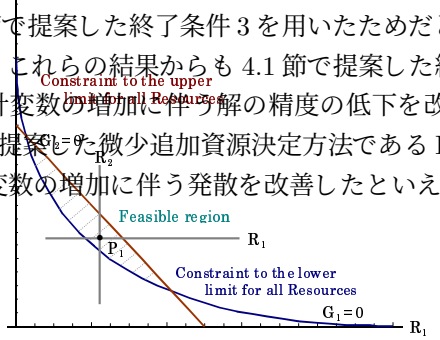


Fig. 8 Accuracy of solution when using a proposed addition resource rule.

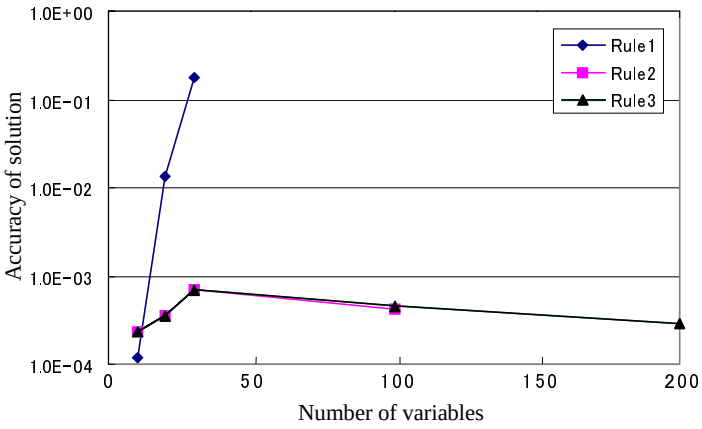
また，Fig. 8 のように Rule 3 では設計変数が増加しても解の精度がほぼ一定である．これは終了条件を 4 章 1 節で提案した終了条件 3 を用いたためだと考えられる．これらの結果からも 4.1 節で提案した終了条件は設計変数の増加に伴う解の精度の低下を改善し，4.2 節で提案した微少追加資源決定方法である Rule 3 は設計変数の増加に伴う発散を改善したといえる．



5. 制約条件の分類方法の提案

5.1 上限制約と下限制約

資源追加削減法は，対象問題を資源最小化問題に限



こで上限制約は下限制約とは異なる資源の見積もり方をする必要がある．そのためにも，まず制約条件を下限制約，上限制約と分類する．ここでは，制約条件の分類方法について検討する．

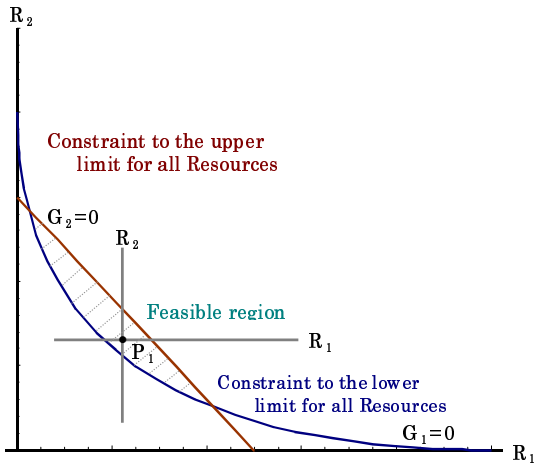


Fig. 9 The way of classifying constraints.

5

Fig. 9 のような要素数 2 の資源平面を考える．現在の設計点は P_1，制約条件は G_1,G_2 の 2 制約とする．要素 R_1 方向において制約条件 G_1 は下限値を与え，制約条件 G_2 は上限値を与える．また，要素 R_2 方向においても制約条件 G_1 は下限値を与え，制約条件 G_2 は上限値を与える．このような場合 G_1 は下限制約，G_2 は上限制約と分類することが可能である．

しかし，Fig. 9 のように制約条件を完全に分類することが出来る場合は限られており，一般的には，制約条件は各要素方向で上限制約にも下限制約制約にもなる．このような場合，制約条件を完全に分類することができない．

そこで，Fig. 10，Fig. 11 に各要素ごとに制約条件を判別する方法を示す．Fig. 10 での設計点は P_1，P_2 で，制約条件は G_1 とする．ここで，資源 R_i 軸方向における制約条件 G_1 の判別を行う．P_1 のように設計点が制約条件内にあり，かつ制約条件 G_1 と P_1 を通る R_i 軸方向の線分との交点 Q_a が P_1 より小さいい場合，制約条件 G_1 は資源 R_i 軸方向において下限制約である．また P_2 のように設計点が制約条件外にあり，かつ制約条件 G_1 と P_2 を通る R_i 軸方向の線分との交点 Q_b が P_2 より大きい場合，制約条件 G_1 は下限制約と判

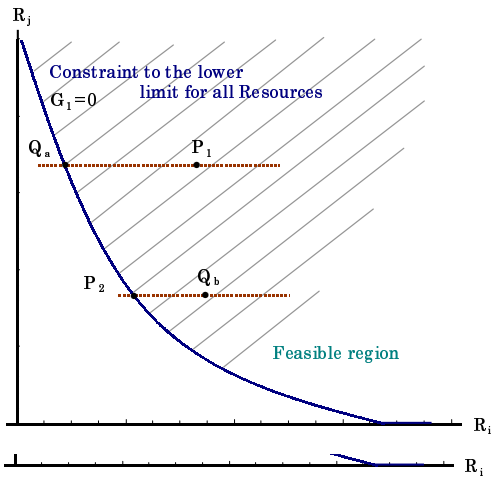


Fig. 10 Distinction of constraint to the lower limit.

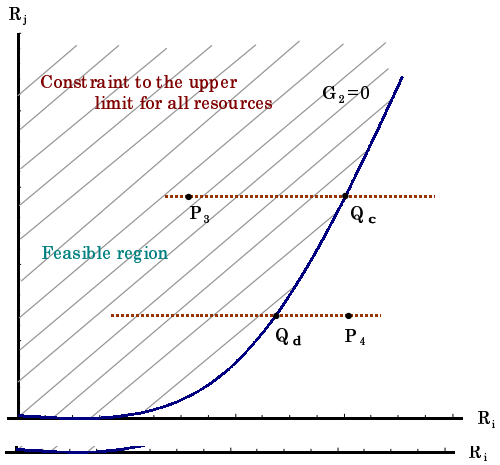


Fig. 11 Distinction of constraint to the upper limit.

6

次に Fig. 11 のような資源平面を考える．ここで，資源 R_i 軸方向における制約条件 G_2 の判別を行う．P_3 のように設計点が制約条件内にあり，かつ制約条件 G_2 と P_3 を通る R_i 軸方向の線分との交点 Q_c が P_3 より

大きい場合，制約条件 G_2 は資源 Ri 軸方向において上限制約である．また P_4 のように設計点が制約条件外にあり，かつ制約条件 G_2 と P_4 を通る R_i 軸方向の線分との交点 Q_d が P_4 より小さい場合，制約条件 G_2 は上限制約と判別できる．この方法を用いることで各要素方向において，制約条件を下限制約，上限制約と分類することができる．ここで重要なことは各要素において制約条件を分類したことで，この分類により従来扱うことのなかった制約条件を扱うことが可能となることもある．

5.2　資源余裕の評価方法の改良

前節で検討した制約条件の分類を行うことにより，各要素で下限制約の臨界資源余裕，上限制約に対する臨界資源余裕を求めることは可能となる．しかし，従来の資源削減処理は下限制約を対象としていたために，上限制約に対しては，新たな資源余裕の評価方法を考える必要がある．そこで Fig. 12 に下限制約と上限制約に対する，設計点 1-6 における資源余裕の評価方法を示す．R_u，R_l はそれぞれ上限制約，下限制約に関する資源余裕である．

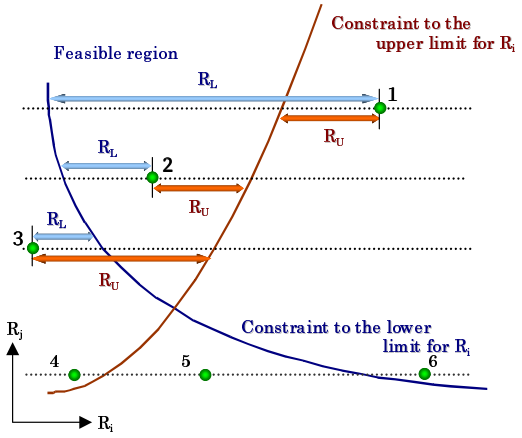


Fig. 12　Estimation way of the resource margin to each constraint.

Fig. 12 の設計点 1-3 に対する資源余裕の評価方法から考える．初期点 1 については，上限制約に対する資源余裕 R_U は過多であり，下限制約に対する資源余裕は R_L となる．この場合，下限制約に対しては余裕があるが，上限制約は満たされておらず，全ての制約条件を満足する方向は資源削減の方向である．よって臨界資源余裕を $(R_u+R_l)/2$ とし，資源削減処理においてそれを削減する．この処理により，設計点 1 を可能領域内に落とすことができる．次に設計点 2 について考える．下限制約条件については R_l が資源余裕となる．このとき，設計点は可能領域内であるため，上限，下限制約ともに満たされている．よって，臨界資源余裕は R_l とし，資源削減処理においてそれを削減する．一方，設計点 3 の場合，上限制約は満たされているが，下限制約に関しては資源不足である．この場合は，設計点 2 と同様，上限制約は満たされているので考慮する必要はなく，下限制約に対しては従来のように資源余裕を評価し，資源削減処理においてそれを削減する．設計点 4-6 については，各資源の軸上に可能領域が存在しないため，資源削減処理は行わない．よって，このような場合，資源余裕は 0 とし，資源追加処理のみを行う．

しか
域が有
のよう
に可能
処理は
なる．
能性が
できな
うな処
して全
を提案
プでは

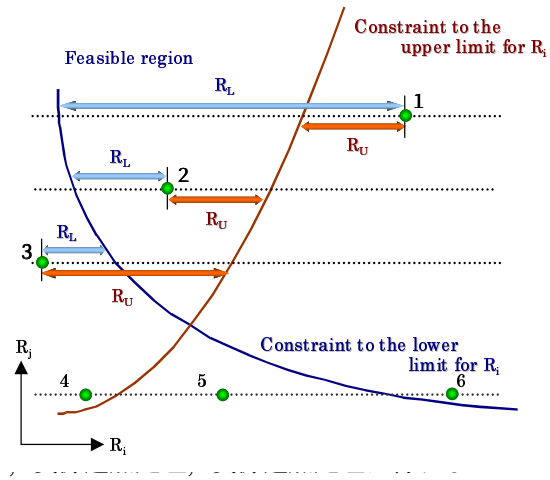


Table 2 Estimation way of the resource margin to

Design point		Resource Margin
1	$R_U>0$, $R_L>0$, $R_U<R_L$	$(R_U + R_L)/2$
2	$R_U<0$, $R_L>0$	R_L
3	$R_U<0$, $R_L<0$, $R_U<R_L$	R_L
4	$R_U>0$, $R_L>0$, $R_U>R_L$	0
5	$R_U>0$, $R_L<0$	0
6	$R_U<0$, $R_L<0$, $R_U>R_L$	0

以上より，各設計点における資源余裕の評価方法は Table 2 の通りである．各資源方向で下限制約，上限制約に対し Table2 に示す資源余裕の評価方法を適用することで，従来考慮されなかった上限制約を扱うことが可能となる．

5.3　上限制約を含む問題への適用

これまで検討してきた制約条件の判別方法，上限制約に対する資源余裕の評価方法の有効性を検証するために，実際に上限制約を含む問題に適用する．対象問題は式 (15) に示される最適化問題とする．目的関数は各要素の資源量の和となり，与えられた制約条件を

7

満足し，総資源量を最小化する資源分布が最適解である．ここで n は設計変数の数となる．

$$
\begin{aligned}
&Minimize\ R_1 + R_2 && (14)\\
&Subject \quad \sqrt{R_1} + \sqrt{\frac{R_2}{2}} \geq 10\\
&\qquad R_2 - R_1^2 \geq 0\\
&\qquad R_1 \geq 0, R_2 \geq 0\\
&\qquad R_1 \leq 100, R_2 \leq 200
\end{aligned}
$$

式 (15) の問題において，制約条件 G_1 は資源 R_1，R_2 に対して下限制約であるが，制約条件 G_2 は資源 R_1 に対しては上限制約，資源 R_2 に対しては下限制約である．この問題における初期点 P からの収束状況を Fig. 13 に示す．

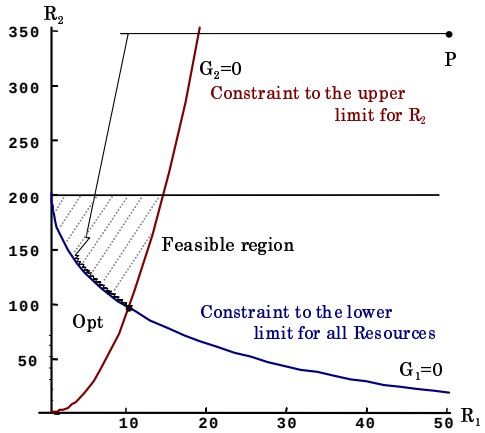


Fig. 13 Change of the design point.

Fig. 13 のように上限制約を含む問題においても，設計点を可能領域外から可能領域内に落とすことが可能となった．しかし，可能領域内で一旦は収束するものの最適解近傍で振動してしまう．これは，微少追加資源量の追加処理により可能領域外にでてしまうためだと考えられる．この場合，資源削減処理で再び可能領域内に入るが，1 ステップ前に比べ最適解から離れる．そのため，再び資源追加処理，資源削減処理を行うが，この繰り返しとなり，終了条件を満たすことがないため解の近傍で振動する．

この問題を改善するために，設計点が一度可能領域内に入った場合，資源追加処理では可能領域外にでないように，また，資源追加後の設計点が可能領域外にでた場合，ΔR の低減処理を行い再び微少資源を追加する処理を加える．この処理を繰り返すことで解の近傍での振動を改善できると考えられる．そこで，式 (15) の問題にこの処理を加えた資源追加削減法を適用した．その結果，任意の設計点から発散することなく収束解を得ることができた．

次に設計変数の数を 5,10,15,$\cdots$,30 と設定し，それぞれの問題に改良を加えた資源追加削減法を適用する．Fig. 14 に，資源追加削減法で得られた解の精度と，計算回数を示す．Fig. 14 のように，設計変数が増加しても振動することなく解を得ている．また，設計変数の数が増加しても，著しい解の劣化，計算回数の増加はほとんど見られない．このことからも，制約条件の判別，上限制約に対する資源余裕見積もり方法，一度可能領域内に落ちた設計点が，資源追加処理において可能領域外にでないようにする，ΔR の低減処理は有効であるといえる．これらの方法を用いることで，従来考慮されなかった上限制約も，資源追加削減法で扱うことが可能となる．

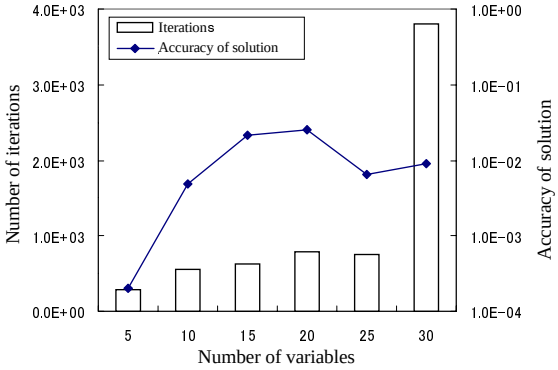


Fig. 14 The adaptation to the problem to contain an upper limit constraint.

6. 結　論

本論文では，資源追加削減法を多変数の非線形最適化問題に適用し，設計変数の増加に伴う問題点を提示した．これらの問題に対しては，資源追加量決定方法の改良を行い，その結果，良好な収束解を得ることができた．また，従来考慮されていなかった上限制約条件についても検討し，新たな資源余裕の評価方法，それに伴う資源削減処理の改良を行うことで，上限制約条件を含む問題についても資源追加削減法の適用が可能となった．得られた結論は以下の通りである．

1) 本論文では多変数の非線形最適化問題に適用し，設計変数の増加に伴う計算回数の増加，解の精度の低下，収束解の発散などの問題点を提示した．

2) 従来，考慮されていなかった終了条件について検討

し，設計変数増加に影響を受けない終了条件を提案した．この終了条件により，より良好な解への収束が可能となった．
3) 従来の資源追加削減法では，発散する可能性がある．これは，設計変数の増加に伴い，資源追加処理における微少追加資源量が増加するのに対し，資源削減量は減少するためである．
4) 新たに，設計変数の増加に影響を受けない資源追加量の設定方法を提案した．この処理を行った結果，多変数の問題においても発散することなく良好な解へ収束した．
5) 資源追加削減法では上限制約は特に考慮されていなかったが，制約条件を各資源で分類し，それぞれ資源余裕を評価することで上限制約の扱いを可能とした．
6) 上限制約のある問題において，最適解近傍で振動してしまう場合があった．そこで，一度可能領域内に落ちた設計点が再び可能領域外にでないよう，新たなΔRの低減処理を行った．その結果，良好な最適解への収束を実現できた．

参 考 文 献

1) 茨木俊秀，福島雅夫「最適化アルゴリズムの最近の動き」システム制御情報学会論文誌，Vol.37(1993).

2) 三木光範「並列分散最適化の最近の研究動向」日本機械学会第 3 回最適化シンポジウム 基調講演 1998.

3) 三木光範，池田大樹「資源追加削減法における追加微少資源パラメータの決定」日本機械学会第 11 回計算力学講演会 投稿論文 1998.

4) 三木光範「並列分散最適化のためのアルゴリズム」計算工学講演会論文集 Vol.1(1996).

5) 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.

6) H.P. ウィリアムス「数理計画モデルの作成法」産業図書 1995.

7) 三宮信夫「最適化」岩波書店 1985.

8) 柏村孝義，白鳥正樹「実験計画法による非線形問題の最適化」朝倉書店 1998.

9) D. デイヴィス「非線形最適化の技法」培風館 1972.

三木 光範・小栗 伸・廣安 知之