

進化的アプローチによる資源追加削減法の高速化

Speed Up of DORAR Method by Evolving Approach

小栗 伸 (知的システムデザイン研究室)

Shin OGURI (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. 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, improvement of algorithm for speed up have not been considered before. In this paper describes speed up of DORAR method by evolutionary approach.

1 緒言

トラス構造物の体積最小化問題に対して提案された資源追加削減法¹⁾(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。この手法は並列処理に適した新しい最適化手法である。これまでに、トラス構造物問題以外に電気回路問題)に適用され、アルゴリズムの改良などが行われてきたが、収束性の向上を目的とした改良は考えられていない。本研究では、資源追加削減法の高速化を目的とした進化的アプローチによるアルゴリズムの改良を行う。

2 資源追加削減法のアルゴリズム

資源追加削減法においてはトラス構造物の総体積を資源ととらえている。目的はシステム全体の体積の最小化であり、総体積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。それらは局所情報により決定される局所的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り、いくつかの最適化問題はこうした問題に変換することが出来る。DORAR法のアルゴリズムは以下の手順で示される。

- 1) 要素ごとに局所制約条件に関する資源余裕を評価。
 - 2) 要素ごとに全体制約条件に関する資源余裕を評価。
 - 3) 資源余裕の最小値を各要素の臨界資源余裕とし削減する。この処理を資源削減処理と呼ぶ。
 - 4) 各要素に一定の微少な資源(以下 R)を追加する。この処理を資源追加処理と呼ぶ。
- 手順1)から4)を繰り返すことにより最適解を得る。

3 進化的アプローチによる改良

資源追加削減法は、任意の初期値からも最適解に収束するというロバスト性を有するが、ある設計点において体積最小化のベクトルと活性な制約条件の方向ベクトルが直交するような場合には、設計点の推移が極めて微少になり、収束までの繰り返し数を要するという問題点がある。そこで、本研究では解探索課程において、設計点の推移が一定値以下に達した場合、各資源軸方向に新たな探索点(子)を生成し、それらの点の中から最も良好な一点を選択(淘汰)し探索を再開するという、進化戦略に基づく方法(以下EDORARと略記する)を提案する。提案する手法による設計点の推移をFig. 1に示す。初期点は P_0 、制約条件は全体制約条件 G_1 と局所制約条

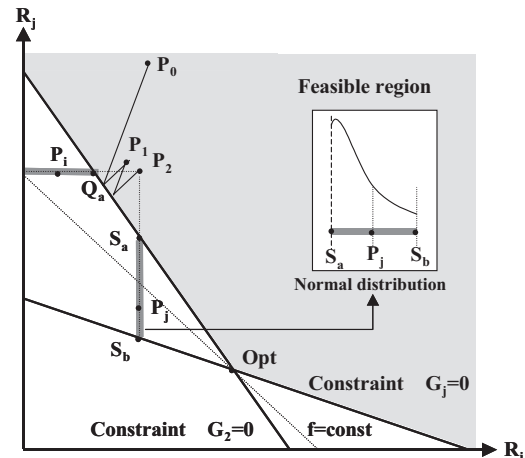


図 1: The way of generating a new point

件 G_2 とする。設計点 P_2 における資源 R_i, R_j 方向における資源余裕はそれぞれ P_2-Q_a, P_2-S_a で表せる。ここ

で、 G_1, P_2 のように設計点の推移が一定値以下になった場合、新たに各資源上の制約条件 G_1, G_2 の間に正規分布を用いて新たな点を発生させ、その後資源余裕の最も大きい点、この場合では P_j を次の探索点として探索を行う。図のように制約条件に張りついた設計点を制約条件の外に出すことにより解探索化の高速化を計る。これは資源追加削減法の解近傍までの早い収束性をいかした改良であるといえる。

4 数値実験

4.1 線形問題

予備実験として提案した手法を線形問題に適用し、計算回数の測定を行う。例題における n は任意の整数であり、 n の値によって設計変数の数を容易に変化させることが出来る。

$$\begin{aligned} \text{Minimize } R &= R_1 + R_2 + \dots + R_n \geq 0 & (1) \\ \text{Subject } R_n + \dots + (n-1)R_2 + nR_1 - 50n \\ R_n + \dots + \frac{R_2}{n-1} + \frac{R_1}{n} - 20n &\geq 0 \end{aligned}$$

設計変数の数を変化させた場合における、従来の資源追加削減法と提案する手法の比較を Fig. 2 に示す。この結果からも線形問題においては、提案する手法を用いることで設計変数の数に影響を受けることなく、従来の手法の約半分の計算回数で収束している。このことから収束性の向上が実現できたといえる。

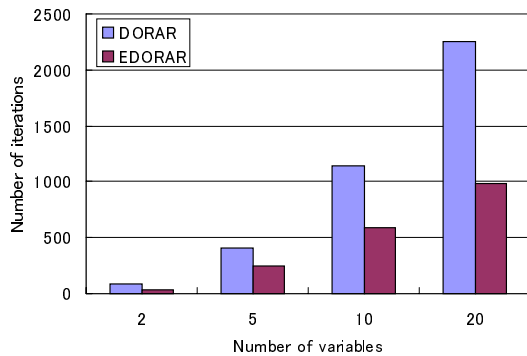


図 2: Relation between number of variable and iteration

4.2 トラス構造最適化問題への適用

次にトラス構造最適化問題に提案した手法を適用する。対象とする問題は設計変数が 10, 15, 20...50 のトラス構造物である。Fig. 3 に設計変数が 25, 50 のトラス構造物と設計変数が 25 のトラス構造物の最適解を示す。ここでは、局所制約条件として各部材の引張力、圧縮座

屈を考え、全体制約として一つの設定変位を考える。目的はトラス構造物の最小となる体積の部材を求めることである。終了条件は総資源量の変化率が一定値以下とする。

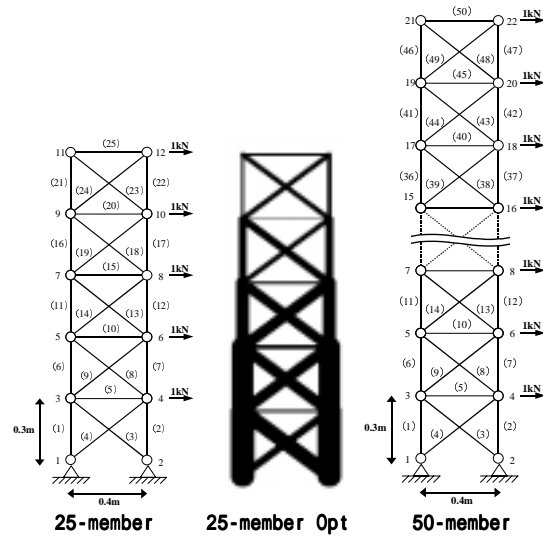


図 3: Truss structure

4.3 最適化の結果

各段数の問題において数通りの初期値を与え、各初期値からの総資源量の変化を測定した。一例として Fig. 3 に 12 接点 25 部材における従来の手法と提案手法の収束状況の比較を示す。Fig. 4 のように計算の初期段階に

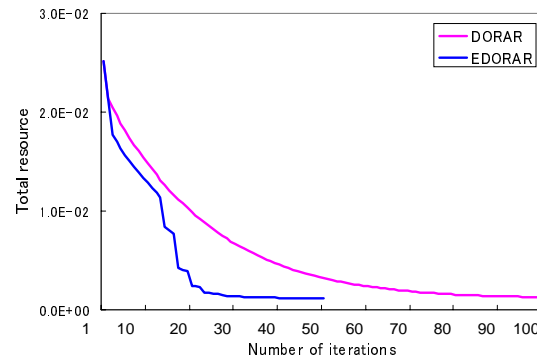


図 4: Change of total resource when using new rule

においては設計点の推移が大きいため、総資源の変化量も大きく従来の手法と提案する手法の差はほとんどない。しかし、提案する手法では総資源量の変化率が一定値以下に達すると新たに点を生成し、選択を行うため従来の手法に比べ、比較的早い段階で収束している。また、提案する手法では新たな点を制約条件外に発生させているものの、得られる解は局所解に陥ることなく収束した。

次に各設計変数のトラス構造物に，提案する手法を適用し，従来の手法と提案手法の計算回数の比較を行う．結果を Fig. 4 に示す．Fig. 4 のようにトラス構造物最適化

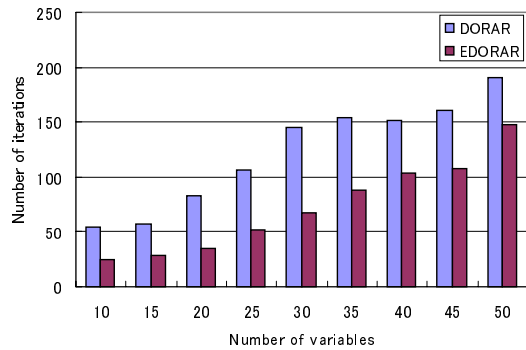


図 5: Relation between number of variable and iteration

問題においても，従来の手法に比べ，提案する手法は，早い段階で収束していることが分かる．このように，提案する手法により資源追加削減法の収束性の向上を実現できた．

5 結 論

本報告では資源追加削減法の高速度を目的とした進化的アプローチによるアルゴリズムの改良を行った．その結果，線形問題においては従来の方法に比べ約半分の計算回数で解を得ることが可能であった．また，トラス構造物問題においても従来の手法に比べ，良好な解を少ない計算回数で得ることが可能となった．このことから提案する手法は資源追加削減法の高速度を実現できたといえる．

参考文献

- [1] 三木光範，廣安知之，”資源の追加と削減に基づく並列分散最適化法”，日本機械学会論文集投稿中
- [2] M.Miki, M.Furuichi, Y.Watanabe, ”Smart Distributed Minimization of the Volume of Discrete Structure”, Proc, AIAA, SDM Conference, pp.2344-2352, 1996
- [3] 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