103 ゾーン分割を用いた大域的最適化 (第2報, 離散値問題への適用)

Global Optimization by Zone Division (2nd Report, Application for Discrete Problem)

○正 吉崎尚史 (日立インダストリイズ) 正 坂田信二 正 広島実

Naoshi YOSHIZAKI, Hitachi Industries
Shinji SAKATA, Hitachi Industries
Minoru HIROSHIMA, Hitachi Industries

The optimization of the various processes from the development to production of the product is carried out. However, there are many cases that are the approximation solution by using RSM(response surface modeling) and also request a local optimum solution by utilizing the mathematical programming. And generally most of the design variables in optimum object include the continuous numbers and discrete numbers. Thereupon, In addition to the former method of zone division for the continuous numbers, we propose the new method that deal with the continuous and discrete design variables. The effectiveness of proposed method has been confirmed by several numerical examples and a real design problem.

Key Words : Optimization. Global Optimization. Discrete Problem. Zone. Random Number

1. 結 論

メーカでは、低コストでかつ信頼性の高い製品を短時間で市場に投入することが要求され、製品の開発から製造までのさまざまなプロセスの最適化が必要となっている。しかし、実製品を最適化の対象とした場合、実用面から考え、応答曲面法を利用した近似解であったり、数理的手法を利用して局所最適解を求める場合が多い。

最適化対象となる設計変数は、連続値だけでなく離散値を含むものも多い。そこで本研究では、前報告⁽¹⁾の連続値に対するゾーン分割法に加え、離散値を含む最適化問題への適用が可能な様に拡張した手法を提案するものである。前報告同様、大域的最適解を求めることを念頭におきながら、最適化における探索方法を無限個の「点」ではなく、解空間を有限個の「ゾーン」として捉える。分割したゾーン内から選んだ任意の点をそのゾーンの点として代表させ、見かけ上、解空間を縮小して、大規模・多峰性問題でも実用的範囲内で最適化を行なう方式である。本報告では手法と計算例による有効性も示す。

2. 提案手法の概要

2.1 最適化問題の定式化

提案する離散変数と連続変数を対象とした最適化問題は以下のように定式化⁽²⁾することができる。

設計変数: $\mathbf{X}$	(1)
目的関数: $f(\mathbf{X})$	(2)
制約条件: $g_s(\mathbf{X}) \leq 0$	(3)
$XM \leq \mathbf{X} \leq XMAX$	(4)

ここで $\mathbf{X}$ は、設計変数ベクトル(離散値, 連続値)であり、 $XMAX$, $XM \leq \mathbf{X}$ は設計変数ベクトルの上下限值である。また、 $f(\mathbf{X})$ は目的関数である。 $g_s(\mathbf{X})$ は不等式制約条件を表し、添字 s は不等式制約条件の数である。

2.2 アルゴリズム

本研究で提案する手法は、式(3)で表される不等式制約条件を含む、一般的な離散変数及び連続変数を含む有制約最適化問題を対象とする。提案する手法の基本的な考え方は、数理計画法、ヒューリスティックな手法⁽³⁾及び解空間をゾーンで捉える考え方を利用することにより、大域的最適解を求めることを念頭におきながら離散変数及び連続変数を含む最適解を求めるものである。

提案する手法は、以下に示す(3)及び(4)の2つのステップから構成され、それぞれゾーン決定ステップ、及び、最小化ステップと呼ぶ。入力データ定義及び初期計算は(1)、(2)に示す。また、Fig.2.1にアルゴリズムを示す。

(1) 入力データ定義

制約条件を満たす各設計変数の初期点: $\mathbf{X}_0$

各設計変数のゾーン分割数(点): ZDN

各設計変数の上下限值: $XMAX$, $XM \leq \mathbf{X}$

ここで、離散変数におけるゾーン分割点の値は、離散変数の値より任意に選択する。

(2) 初期計算

- (a) 制約条件を満たす各設計変数の初期点を $\mathbf{X}_0$ として、その点における目的関数値 $f(\mathbf{X}_0)$ を求める。このステップは最適化の最初に一回のみ行なわれる。

(3) ゾーン決定ステップ

Fig.2.1 Algorithm

Fig.2.2 Zone decision step

- (b) 乱数を発生させ、分割された解空間上の任意のゾーンを選ぶ。(Fig.2.2ではゾーン番号「06」を選択)
- (c) 選ばれたゾーンが、過去に選ばれたゾーンであれば、(b)に戻り再度乱数を発生させる。
- (d) 選ばれたゾーン内の複数点 X^1, X^2, X^3, X^4 の、目的関数値 $f(X^i)$ 及び制約条件を満たすか否かを計算する。
- (e) (d)で計算した点の中で、制約条件を満たし、一番小さな(大きな)目的関数値でかつ初期計算(又は前の最小化ステップ)で計算された(選ばれた)点より、小さな(大きな)目的関数値を持つ点 X^* を選ぶ。条件を満たす点がなければ (b)に戻る。
- (4) 最小化ステップ

(f) (e)で選ばれた点 X^* を始点として、数値計画法で局所最適点を求め、最小(最大)の目的関数値を更新する。

(g) 終了条件を満たしていれば探索を終了し、最後に (f) で求めた値を大域的最適解とする。していなければゾーン決定ステップへ戻る。

また、(f)の最小化ステップでは、Kuhn-Tucker 条件を満たして終了するため、得られた解の局所最適性が保証される。

3. 連続値問題の基礎的検討

数値計算例を通して、本報告で提案する手法の有効性を示す。数値計算例では解空間及び探索過程の可視化の為に2変数関数問題を考える。2つの例題はともに、2設計変数1制約条件問題であり、数値計算例1では解空間の比較的簡単な問題を、数値計算例2では局所最適解の多い複雑な多峰性解空間の問題を考える。最小化ステップにおける最適化手法は、逐次二次近似法を用いる。

3.1 数値計算例1

式(6)を不等式制約条件とし、式(5)の目的関数を最小化する2設計変数1制約条件の最適化問題を考える。側面制約条件は式(7)で与えられる。また、解空間は Fig.3.1 のようになる。

Fig.3.1 Design space of sample 1

$$f(X) = \frac{1}{2} \sum_{i=1}^2 (x_i^4 - 16x_i^2 + 5x_i) \quad (5)$$

$$g_1 = x_1^2 + x_2^2 - 9 \leq 0 \quad (6)$$

$$-5 \leq x_i \leq 5 \quad (i=1, 2) \quad (7)$$

本問題においては、設計変数のゾーン分割数を各々10とし、初期点をA ($x_1=1.0, x_2=0.5$)として問題を解く。探索プロセスを Fig.3.2 に示す。まず、制約条件を満たす初期点の目的関数値を求め、-5.719を得た。次に乱数を発生させゾーン22を得た。ここで得たゾーンの4ヶの頂点の目的関数値と制約条件を満たすか否かを計算し、4ヶの頂点の中で、制約条件を満たして、目的関数値が一番小さく、かつ、初期点で計算した目的関数値-5.719より小さい点37を得た。このとき目的関数値-58.0を得た。更にこの点を初期点として最小化を行ない、最終的に大域的最適解 (-2.121, -2.121)を得た。このときの目的関数値は、-62.3566となった。また、Fig.3.3に同問題でケース3の場合の探索過程を示す。Fig.3.2とFig.3.3を比較しても分かるように、探索過程がまったく異なっている。これは、ゾーン決定に乱数を用いているためである。

更に, Fig.3.3 に見られるように, ゾーン決定ステップ後の最小化ステップの探索領域は, 選ばれたゾーン以外の解空間上も対象範囲となり, 選択されたゾーンに縛られることなく, 解

Fig.3.2 Search process of case 9

Fig.3.3 Search process of case 3

空間の状態を正確に捉えることができる様になっている. しかし, どのケースにおいても最終的には大域的最適解を探索し, 本提案手法が有効な最適化手法であることが分かる.

そこで, 同問題を1回だけ行なって, 回数を議論するには無理があると考えられるため, Table3.1 及び Table3.2 に示すように, 初期点をA及びB ($x_1=-1.0$, $x_2=0.5$) として各々10回計算を行ない検討を行なう.

表に各ケースの大域的最適解到達までの関数呼び出し回数(最小化ステップでの関数呼び出し回数+大域的最適解到達までのゾーン決定ステップでの関数呼び出し回数), 最小化ステップ回数, 最小化ステップでの関数呼び出し回数, 及び, 最適化終了時の関数呼び出し回数(最小化ステップでの関数呼び出し回数+ゾーン分割数で決まる探索点の総数)を示す. 両初期点の結果から大域的最適解到達時の関数呼び出し回数に大きなばらつきが見られ, 乱数の特徴が良く現れている. したがって, 探索過程の再現性はなく, 最適化を行なう度に大域的最適解到達時の関数呼び出し回数は異なってくる. しかし, この問題に関しては, 両初期点の全てのケースで大域的最適解に到達し, 回数も平均で約106回で到達できた. また, 最適化終了時の関数呼び出し回数は, ばらつきもかなり小さく, 平均で約170回と一般的な大域的最適化手法である SA⁴⁰や GA と比較して少ない回数で到達できる. また, ゾーン分割数で決まる探索点の総数, 及び, 最小化ステップ回数から計算終了時の回数が予測できる.

3.2 数値計算例2

次に, 式(8)から(10)で与えられる, より複雑な解空間を持つ2設計変数1制約条件の最適化問題を考える. 解空間は, Fig.3.4 のようになる.

$$f(X) = 418.9829n + \sum_{i=1}^2 \left\{ -x_i \sin \sqrt{|x_i|} \right\} \quad (8)$$

$$g_1 = -x_1^2 - x_2^2 + 300^2 \leq 0 \quad (9)$$

$$-500 \leq x_i \leq 500 \quad (i=1, 2) \quad (10)$$

本問題においても設計変数のゾーン分割数を各々10とし, 初期点を ($x_1=200.0$, $x_2=-400.0$) として問題を解く. 数値計

Table.3.1 Optimization results (Initial point A)

Initial point A(1.0,0.5)	Case number										Average
	1	2	3	4	5	6	7	8	9	10	
Number of function calls in global minimum	137	98	132	125	56	134	121	169	29	66	106.7
Number of minimization steps	2	2	3	2	2	2	2	3	1	2	2.1
Total number of function calls in minimization steps	52	51	70	52	47	47	57	70	20	57	52.3
Total number of function calls in optimization	174	173	192	174	169	169	179	192	142	179	174.3

Table.3.2 Optimization results (Initial point B)

Initial point B(-1.0,0.5)	Case number										Average
	1	2	3	4	5	6	7	8	9	10	
Number of function calls in global minimum	164	175	105	81	68	121	110	84	123	35	106.6
Number of minimization steps	2	2	2	2	2	2	2	2	2	1	1.9
Total number of function calls in minimization steps	57	57	57	49	51	47	57	48	47	30	50.0
Total number of function calls in optimization	179	179	179	171	173	169	179	170	169	152	172.0

Table.3.3 Optimization results

Initial point (200,-400)	Case number										Average
	1	2	3	4	5	6	7	8	9	10	
Number of function calls in global minimum	318	213	344	182	166	223	86	322	197	208	225.9
Number of minimization steps	4	3	4	2	2	2	1	3	3	2	2.6
Total number of function calls in minimization steps	244	192	260	120	153	121	81	211	160	121	166.3
Total number of function calls in optimization	366	314	382	242	275	243	203	333	282	243	288.3

Fig.3.4 Design space of sample 2

算例1同様10回行った結果を Table. 3.3 に示す。まず、最小化ステップの平均回数を見ると、数値計算例1と比べて約0.6回増えている。これは解空間がより複雑（山・谷の数が増えた）になったことにより、探索過程で最小化ステップの回数が増加したためである。一般に複雑な解空間を持つ問題ほど、また、ゾーン分割数が多くなるほど、最小化ステップの回数が増加するため、最適化終了時の関数呼び出し回数が多くなる。

4. 離散値問題への拡大適用

次に、離散変数および連続変数を含む問題での本手法の有効性を、数値計算例を通して検討する。前述の数値計算例1で示した解空間の問題で有効性の検討を行なう。今度は、最小化ステップにおける最適化手法は、SAM (Successive Approximation Method) を用いる。

4.1 数値計算例1を使った離散値問題への適用

式(11)~(14)に示す離散値と連続値を含む2設計変数1制約条件の最小化問題を考える。

$$f(X) = \frac{1}{2} \sum_{i=1}^2 (x_i^4 - 16x_i^2 + 5x_i) \quad (11)$$

Fig.4.1 Search process of sample 1

$$g_1 = x_1^2 + x_2^2 - 9 \leq 0 \quad (-5 \leq x_i \leq 5) \quad (12)$$

ここで、 $x_1: -5, -4.5, -2.121, -1, 0$

$$0.25, 1, 2.121, 3, 5 \quad (13)$$

$$x_2: \text{連続値実数} \quad (14)$$

離散値変数 x_1 のゾーン分割数を9とし、本問題は、離散変数 x_1 の取り得る離散値と分割点を同じにする。分割点を(13)に示す。また、連続値設計変数 x_2 のゾーン分割数を10の等分割とする。初期点を▲($x_1=1.0, x_2=0.5$)として問題を解く。探索の過程の一例を Fig.4.1 に示す。初期点の目的関数値-5.719を求め(制約条件は満たしている)、次に乱数を発生させゾーン39を得た。ここで得たゾーンの4ヶの頂点の目的関数値と制約条件を満たすかを計算し、4ヶの頂点の中で一番小さく、かつ、初期点で計算した目的関数値-5.719より小さい点44を得た。更にこの点を初期点として最小化を行ない、局所最適解(-1.0, -2.828)を得た。このようにして、最終的に大域的最適解(-2.121, -2.121)を得た。

Table.4.1 Optimization results

Initial point (1.0,0.5)	Case number										Average
	1	2	3	4	5	6	7	8	9	10	
Number of function calls in global minimum	83	165	71	229	185	67	189	197	44	136	136.6
Number of minimization steps	2	3	2	4	3	2	3	3	1	2	2.5
Total number of function calls in minimization steps	50	114	56	148	114	53	108	117	31	50	50.0
Total number of function calls in optimization	161	225	167	259	225	164	219	228	142	161	195.1

次に探索回数について検討する。Table.4.1 に初期点を $x_1=1.0$, $x_2=0.5$ として 10 回最適化を行なった時の、大域的最適解到達までの関数呼び出し回数、最小化ステップの回数、最小化ステップでの関数呼び出し回数、及び、最適化終了時の関数呼び出し回数の平均値を示す。前述のようにゾーン決定に乱数を用いているため Table.4.1 に示すように最適化終了時の関数呼び出し回数にバラツキがみられる。また、本問題に関しては約 140 回程度の関数呼び出し回数で大域的最適に到達し、また、終了までの関数呼び出し回数も約 200 回と一般的な広域的探索手法である SA⁽⁴⁾や GA と比較して少ない回数で到達でき、離散値を含む最適化問題においても、本提案手法の有効性を確認できた。

4.2 離散値変数のゾーン分割と大域的最適化

次に離散値変数のゾーン分割方法の相違による探索過程の違いについて検討する。前述 4.1 同様離散値と連続値を含む 2 設計変数 1 制約条件の最小化問題を考える。本問題の離散値の取り得る値を (17) に、また、ゾーン分割における分割点を (18) として最適化を行なう。

$$f(X) = \frac{1}{2} \sum_{i=1}^2 (x_i^4 - 16x_i^2 + 5x_i) \quad (15)$$

$$g_1 = x_1^2 + x_2^2 - 9 \leq 0 \quad (-5 \leq x_i \leq 5) \quad (16)$$

ここで、

$$\begin{aligned} \text{離散値 } x_1: & -5, -4.7, -4.5, -4, -3, -2.5, \\ & -2.121, -1.5, -1, -0.5, 0, 0.25, 0.5, \\ & 1, 2, 2.121, 2.5, 3, 3.5, 4, 5 \end{aligned} \quad (17)$$

$$\begin{aligned} \text{分割点 } x_1: & -5, -4.5, -3, -1.5, 0, \\ & 1, 2.5, 3.5, 5 \end{aligned} \quad (18)$$

$$x_2: \text{連続変数} \quad (19)$$

前述 4.1 の様に離散値の取り得る値とゾーン分割における分割点を同じとした場合は、Table.4.1 に示すように 10 回行なった全てで大域的最適解に到達することができた。しかし、上記 (18) のようにゾーン分割における分割点を、離散値の取り得る値の数より少なく、更に、大域的最適解上に分割点を取らなくすると Table.4.2 に示すように、大域的最適解を探索することができない場合が生じた。大域的最適解に到達した場合の探索過程の一例を Fig.4.2 に、到達できなかった場合の一例を Fig.4.3 に示す。これは、一つのゾーンを大きく取ったため、99ヶの探索点が解空間の山・谷をうまく捉えることができなかったことによる。すなわち、探索半ばで局所最適点 $(-1.5, -2.6)$ に到達し、この後のゾーン決定ステップにおいて、局所最適点より目的関数値の小さな探索点が存在しなくなったからである。これらのことから、本手法を用いて最適化を行なう場合は、各設計変数のゾーン分割数 (点) をいくつにするかが課題となる。一般には、ゾーン分割数 (点) を大きく取れば、解空間をより正確に捉えることができ、大域的最適解に到達する可能性も高くなるが、解空間の探索点が増加し多く

Fig.4.2 Optimization results

Fig.4.3 Optimization results

の計算回数 (時間) を必要とする。課題は残るものの、開発から製造に至るさまざまなプロセスの最適化は時間との勝負であり、適切な計算規模と計算回数の制約を受けるため、計算可能な回数から分割数 (点) を決定でき、かつ、大域的最適解を求めることを念頭においた本手法は、実用的に有効な最適化手法といえる。

5. 実設計問題への適用

次に、3 設計変数 8 制約条件を持つ大小傘歯車の歯数及びモジュールを最適化する実設計問題を考える。本最適化問題は

Table.4.2 Optimization results

Initial point (1.0,0.5)	Case number										Average
	1	2	3	4	5	6	7	8	9	10	
Number of function calls in global minimum	131	x	x	x	127	119	x	106	x	x	120.8
Number of minimization steps	2	1	1	1	2	2	1	2	1	1	1.4
Total number of function calls in minimization steps	59	22	34	22	59	59	34	59	34	34	41.6
Total number of function calls in optimization	159	122	134	122	159	159	134	159	134	134	141.6

以下のように定式化することができる。本来この問題は、2つの傘歯車の歯数及びモジュールを設計変数として求めるものであるが、歯数比は入出回転数でほぼ決まるため、多少の裕度を持った出力回転数を設計変数として最適化を行なった。またモジュール m は、0.1の倍数を取るため最適化の中での処理は、10倍の値(整数値)を使った。

設計変数: $\mathbf{X} (z_1, m, out)$ (20)

z_1 (小歯車歯数: 整数)

m (モジュール: 整数)

out (出力改選数: 実数)

目的関数: $f(\mathbf{X})$ (歯車の大きさを表す係数) (21)

制約条件: $g_i(\mathbf{X}) \leq 0$ (22)

歯幅, 大歯車直径等8制約条件

$XMIN \leq X \leq XMAX$

本問題においては、設計変数のゾーン分割を各々10とし初期点を $\mathbf{X} (13, 160, 196.21)$ 、目的関数値 $f(\mathbf{X})$ を459.91として問題を解く。分割数よりゾーン決定ステップでの探索回数は、 $11^3 = 1331$ 回で、これに最小化ステップでの計算回数を加算した1400回程度がこの問題での計算終了回数と考えられる。

上記条件にて最適化を行なった結果、 $z_1=13$ $m=136$ $out=194$ 目的関数値390.93を得た。また、最適解探索までの計算回数が148回で、計算終了までの計算回数は1343回であった。Fig.5.1に本問題の解空間を示す。左上図中下部の直方体部分は今回の問題で制約条件を満たす点を持つ領域を示し、その領域を拡大した右下の図の○が初期点を、また、●が最適解を示す。本最適解は、設計者が時間をかけて行なった設計計算と同値であり、数十分で最適解を計算できる本手法は実設計問題においても有効であることが分かった。従来はGAを用いた最適化を行っており、5000回程度の計算回数を要していた。

Fig.5.1 Design space and results

6. 結言

従来の解空間を有限個のゾーンに分割し、分割したゾーンから選んだ任意の点をそのゾーンの点として代表させ、少ない探索点を用いて、実用的な時間内で最適化(大規模・多峰性問題を含む)を行なう連続値対応のゾーン分割法を拡張し、離散値も扱うことのできる手法を提案し以下の結論を得た。

- (1) 数値計算例及び実設計計算を通して手法の有効性を確認した。
- (2) 計算可能な回数(時間)から分割数が決定できる。
- (3) 設計空間の全域を探索が可能で、分割数により大域的最適解を求めることができる。
- (4) 乱数を用いているが、複数回の同一ゾーン(探索点)探索回避により、計算時間を大幅に短縮できる。
- (5) 最適解が実空間上にあり、Kuhn-Tucker条件を満たす。

7. 今後の課題

- (1) ゾーン決定ステップ後の、ゾーン内の決められた複数点選択を単点選択による、アルゴリズムの簡素化。
- (2) 構造最適化時の最小化ステップ始点決定において、目的関数が条件を満たさなかった場合、制約関数の計算を省略し、計算時間短縮を計るアルゴリズムの導入。
- (3) 探索精度の向上方法の検討。

謝辞

本研究を行なうに当たり、御助言を頂いた金沢大学工学部山崎光悦教授、北山哲士助手に謝意を表する。

参考文献

- (1) 吉崎尚史・他4名, ゾーン分割を用いた大域的最適化(第1報), Design Symposium 2004 講演論文集, pp. 117-120 (2004)
- (2) 山川宏, 最適化デザイン, 培風館, (1993)
- (3) 北山哲士・山崎光悦, 一般化ランダム・トンネリング・アルゴリズムによる大域的最適化, 日本機械学会論文集, 69-684, A, pp. 1250-1256 (2003)
- (4) 中野良平, シミュレーテッド・アニーリング, 人工知能学会誌, 9-3, pp. 365-372 (1994)