社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2006－MPS－62 (3)
2006－BIO－7 (3)
2006／12／21

ペナルティ法による目的関数生成における重み付け自動化

村田幸弘 安藤竜弥 阿部充志

株式会社 日立製作所 電力電機開発研究所

多制約条件を課した場合の，ペナルティ法を用いた汎用的な目的関数作成手法を提案する．制約条件が複数存在する場合，得られる解の精度はその重みの良し悪しに依存する．本研究では，各ペナルティ関数と最小化する評価関数の代表値の比を各重みとした．代表値は探索回数毎に求められる関数値点列の近似曲線から求め，重みは規定回数毎に自動的に更新する．これにより，各関数が偏り無く評価され，局所解に陥ることを回避できる．提案手法をテスト関数に適用した結果，これまでに提案された手法と同等以上に高精度な解を得られることがわかった．

キーワード: ペナルティ法，多目的関数，自動重み付け，スプライン関数，平滑化

The Use of Automated Weighting to Generate a Multiobjective Function in the Process of Penalty Method

Yukihiro MURATA, Ryuuya ANDO, and Mitsushi ABE

Power & Industrial Systems R&D Lab., Hitachi, LTD.

This paper proposes a method to generate a multiobjective function with automated weighting, and how it can be applied to many well-known optimization algorithms, such as penalty method. In cases where a lot of constraints exist, the weighting is critical. In this paper, the ratio of the evaluated function value over each penalty function value is used to determine each weight. Each function value is smoothened through spline fitting. These weights should be updated automatically at predefined times. This proposed method could diminish the values of each function impartially, thereby avoiding obtaining local optimum solutions. In addition, this paper shows the results of benchmark tests which showed better performance at solving some test problems than by already-proposed algorithms.

KEYWORDS: penalty method, multiobjective function, automated weighting, spline function, smoothing

1 はじめに

近年，最適化計算の工学機器設計への応用は目覚しく進展している．しかし，それにも関わらず，制約条件を課した最適化には様々な問題が浮上している．これは，制約条件をどのような形で最適化問題に導入すれば最適解が得られるかが未解明であるためである．また，対象となる目的関数の形状を知ることが困難である場合が多く，更に工学機器設計での制約条件も一層厳しくなっている．

このような背景の下，複数の制約条件を関数化し，それをペナルティ関数として目的関数に取り込み，無制約条件下での最適化問題に置き換える手法 (ペナルティ法) で，工学機器を設計することに成功した (I.Parmee, 1998)．しかしながら，ペナルティ関数に乗じる重み係数によって，その結果の良し悪しが左右されるため，この手法を他の工学設計に応用することは困難であり，設計者が重み係数の設定，変更を繰り返して解を得るしかなかった．

一般的に，制約条件を満足する大域的最適解を得るためには，実現可能領域と実現不可能領域の双方を探索する必要がある．このため，目的関数を適切に作成しないと，どちらかの領域に偏った探索が進行し最終的に局所解に陥ってしまう．

これまでもこのような問題を解決するための研究が行われてきた．近年では，遺伝的アルゴリズム (Genetic Algorithm: GA) による多制約条件付き最適化問題に関連する研究が盛んに行われている (COMOGA, VEGA, MOGA, NPGA)．GA は，初期値依存性の低い最適化アルゴリズムであるが，計算量が多い点が問題である．更に，工学機器設計では，経験に基づいた良解が存在する場合が多く，軽微な変更の際には変更前のパラメータを初期値として利用できるため，スピードを重視した最適化アルゴリズムが望まれる．

そこで，比較的短時間に最適解や局所解を探索できる焼きなまし法 (Simulated Annealing: SA) に適用可能な，ペナルティ法による汎用性に富む目的関数を作成した．各ペナルティ関数値と最小化する評価関数値とは相関関係はないため，これらの線形和である目的関数の最小化は，ある特定の関数のみの最小化となりうる．本研究では，この対策として各ペナルティ関数に可変重みを乗じることで偏りを排除した．ところで，重みの変動量が大きい場合は目的関数が固定されない問題がある．そこで，本研究では探索毎に得られる各関数値から成る点列に振動の小さい近似曲線 (平滑化自然スプライン関数) を当てはめ，曲線上の点を用いて重み付けを行うことで変動を抑える工夫を施した．本手法をよく知られたテスト関数に適用し，これまでに提案された手法と比較して，その有効性を検証する．

2　目的関数作成手法概要

いま，最小化したい評価コスト関数を $f(\boldsymbol{x})$ とし，m 個の制約条件を $g_k(\boldsymbol{x}) \geq 0 (k = 1, 2, \cdots, m)$ とする．

Minimize: $f(\boldsymbol{x})$

subject to:

$$g_k(\boldsymbol{x}) \geq 0, \quad k = 1, 2, \cdots, m.$$

提案手法では，制約条件をコスト化した関数 (ペナルティ関数とよぶ)$p_k(\boldsymbol{x}) (k = 1, 2, \cdots, m)$ を評価コスト関数 $f(\boldsymbol{x})$ に加算して目的関数 $o(\boldsymbol{x})$ を作成する．

$$o(\boldsymbol{x}) = f(\boldsymbol{x}) + \sum_{k=1}^{m} w_k p_k(\boldsymbol{x}). \tag{1}$$

ここで，ペナルティ関数 $p_k(\boldsymbol{x})$ に乗じた $w_k (\geq 0)$ は重みである．また，$p_k(\boldsymbol{x})$ は制約対象の計算値と制約値を用いて，

$$p_k(\boldsymbol{x}) = \frac{|\,\text{計算値} - \text{制約値}\,|}{\text{制約値}}, \tag{2}$$

と定義する．以下のテスト関数では，

$$p_k(\boldsymbol{x}) = g_k(\boldsymbol{x}), \tag{3}$$

とした．これらの各コスト関数値は最適化による探索が実施される毎に求まる．

ところで，評価コスト関数 $f(\boldsymbol{x})$ と得られたペナルティ関数 $p_k(\boldsymbol{x})$ 間で数値に大きな差がある場合，それらの線形和である目的関数 $o(\boldsymbol{x})$ の最小化問題は，最も大きな関数のみの最小化問題に置き換わる．このまま探索が進行すれば，全ての制約条件を満足する解は得られないことは容易に想像できる．そこで，ペナルティ関数が評価コスト関数と凡そ同値となるような重み w_k をペナルティ関数に乗じることにより，関数間で偏り無く目的関数を最小化する．具体的には，重み w_k は各ペナルティ関数値で評価コスト関数値を規格化した値に相当する．

$$w_k \sim \left| \frac{f(\boldsymbol{x})}{p_k(\boldsymbol{x})} \right|. \tag{4}$$

ここで，重みを等式ではなく近似式で示したことに注意する．重み更新に際し，劇的に重みが変動した場合，つまり各関数値が大きく変動した場合，目的関数 $o(\boldsymbol{x})$ 自体が変動し，最小化対象が不明瞭となる．そこで，変動する各関数値そのものの比を重みとして算出するのではなく，変動する各関数それぞれにおいて探索回数をパラメータとした各関数値の近似曲線を描き，その曲線上の点を各関数値の代表値として選定し，その比を算出する．近似曲線には，ベジエ曲線などがよく用いられるが，変動を吸収した滑らかな近似曲線とはならない．そこで，各関数値に対して，それらの値を反映しながらも滑らかさに重点を置いた近似曲線 (平滑化自然スプライン関数) を採用した．

平滑化自然スプラインとは，以下に示す二つの量の線形結合を最小にすることによって座標上のデータ点列 $(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n) (a < x_1 < x_2 < \cdots < x_n < b)$ を平滑化する手法である [1]．その量とは，

1. データとそれに対応する平滑化曲線上の値との差を二乗し，それに重みを乗じた値の総和，
2. データの対応する平滑化曲線上の値の l 階差分の二乗和 (l は与えられた整数)

であり，これら二つの量を結合する割合は，データに対して忠実であるか滑らかであるかのどちらに重点を置くかで決まる．具体的には，正の重み ω_i および g を用いて σ を，

$$\sigma = \sum_{i=1}^{n} \omega_i \{f(x_i) - y_i\}^2 + g \int_a^b \{f^{(l)}(x)\}^2 dx, \tag{5}$$

と定義すれば，σ を可能な限り小さくする l 階連続微分可能な関数 $f(x)$ が求める近似曲線 (平滑化自然スプライン関数) となる．正の重み g によって，滑らかさの度合いを調整できる．

探索回数と評価コスト関数値およびペナルティ関数値をそれぞれ横軸と縦軸とした n 個のデータ列 $(1, f_1(\boldsymbol{x})), \cdots, (n, f_n(\boldsymbol{x}))$ および $(1, p_{k,1}(\boldsymbol{x})), \cdots, (n, p_{k,n}(\boldsymbol{x}))$ に対して，n 個のデータ列を用いて描いた上記近似曲線上の n 個の点列をそれぞれ $(1, sf_1(\boldsymbol{x}))$, $\cdots$, $(n, sf_n(\boldsymbol{x}))$ および $(1, sp_{k,1}(\boldsymbol{x}))$,$\cdots$, $(n, sp_{k,n}(\boldsymbol{x}))$ とする．各近似曲線は各データ列に忠実ではあるものの，滑らかさに重点を置いているため，変動の少ない曲線となっている．尚，ペナルティ関数値がゼロとなるデータ (制約条件を満足するデータ) はデータ列から除外して近似曲線を算出する．ここで得られた近似曲線上の点から，式 (4) と同様に比を算出して重み w_k とする．

$$w_k = \left| \frac{sf_n(\boldsymbol{x})}{sp_{k,n}(\boldsymbol{x})} \right|. \tag{6}$$

また，制約条件を満足させる度合いにより，これらの重みに更に定数を乗じることも有効である．重みは一定回数探索が進行する度毎に更新する．

以上述べた手法により，評価コスト関数内での各項の関係は，適切な重みで均衡を保つことができるから，特定の

コスト関数のみが最小化されるような偏った最適化は起こらない.

3 提案手法の有効性の検証

提案手法で大域的最適解にどの程度近づいているかを,テスト関数を用いて検証した. テスト関数にはMichalewicz等の不等式制約を課した関数 [2] を採用し, このテスト関数を用いてこれまでに提案されたアルゴリズム (COMOGA, VEGA, NPGA, MOGA) を評価した結果については, 文献 [3] を用いた.

尚, 問題ごとに重み係数を調整したのでは比較検証できないので, 各アルゴリズムにおいて使用する定数は全ての関数に対して同等とした. また, パラメータ初期値はパラメータ領域の中間値とした. そのため, 初期値に依存する場合は得られる結果は最良解でないかもしれない. 後述するベンチマーク結果については, 文献 [3] に記載の最良解を各アルゴリズムの結果として併記した. また, 考察では, 上記探索に機械的な操作を加えることで, より大域的最適解に近づけることが可能であることを示す.

最適化アルゴリズムには, パラメータ感度の違いに対応した焼きなまし法 (Simulated Annealing: SA) である, Adaptive Simulated Annealing(ASA) [4] を採用した.

3.1 提案手法による重みと関数値の推移

本項では, ペナルティ関数に乗じた重みおよび評価コスト関数値の推移を確認する. 対象としてテスト関数 $g01$ を選定した.

Minimize:

$$f(\boldsymbol{x}) = 5\sum_{i=1}^{4} x_i - 5\sum_{i=1}^{4} x_i^2 - \sum_{i=5}^{13} x_i \tag{7}$$

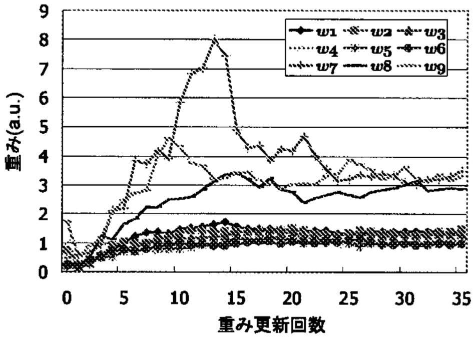
subject to:

$$g_1(\boldsymbol{x}) = 2x_1 + 2x_2 + x_{10} + x_{11} - 10 \le 0 \tag{8}$$
$$g_2(\boldsymbol{x}) = 2x_1 + 2x_3 + x_{10} + x_{12} - 10 \le 0 \tag{9}$$
$$g_3(\boldsymbol{x}) = 2x_2 + 2x_3 + x_{11} + x_{12} - 10 \le 0 \tag{10}$$
$$g_4(\boldsymbol{x}) = -8x_1 + x_{10} \le 0 \tag{11}$$
$$g_5(\boldsymbol{x}) = -8x_2 + x_{11} \le 0 \tag{12}$$
$$g_6(\boldsymbol{x}) = -8x_3 + x_{12} \le 0 \tag{13}$$
$$g_7(\boldsymbol{x}) = -2x_4 - x_5 + x_{10} \le 0 \tag{14}$$
$$g_8(\boldsymbol{x}) = -2x_6 - x_7 + x_{11} \le 0 \tag{15}$$
$$g_9(\boldsymbol{x}) = -2x_8 - x_9 + x_{12} \le 0 \tag{16}$$

パラメータ:

$$0 \le x_i \le 1 (i = 1, \ldots, 9),$$
$$0 \le x_i \le 100 (i = 10, 11, 12), 0 \le x_{13} \le 1. \tag{17}$$

ペナルティ関数は 9 個あり, 重みをそれぞれ $w_1, w_2, \cdots, w_9$ としている. 重みは 200 回探索する毎に更新することとした. つまり, 図 1(上) には 7000 回探索 (35 回重み更新) まで各重みをプロットしている. 示していないが, これ以降は, 小刻みな振動はあるもののほぼ一定値に到達している. この結果から, 局所解を避けながら目的関数が徐々に固定され, 本研究の狙いを達成しているといえる.



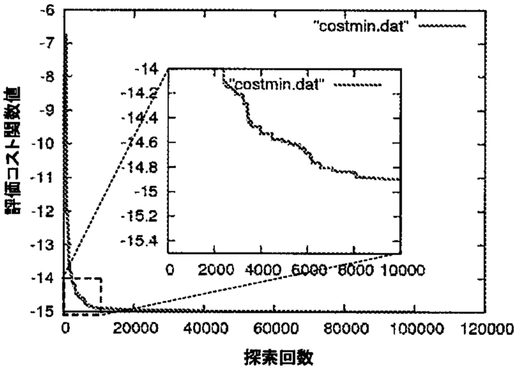


図 1: テスト関数 $g01$ での重み (上) と関数値 (下) 推移

また, 得られた目的関数の最小化によって, 制約条件を満足しながら評価コスト関数値を最小化していく推移を図 1(下) に示す. プロットは各探索回数までの評価コスト関数最小値である. 探索回数が 7000 回近傍までは重みが変化し評価コスト関数値も急激に減少しているが, 重みがほぼ固定された後は緩やかに減少している. 工学に応用するべくパレート最適解を得るためには, 本テスト関数での探索回数は凡そ 20000 回程度と予想される.

表 1: ベンチマーク結果

手法 / 関数#	optimum	proposal	COMOGA	VEGA	NPGA	MOGA
g01	-15.0	-15.000000	-4.806906	-11.136517	-11.007717	-14.504487
g02	0.803619*	0.801036	0.021716	0.000212	0.790404	0.680874
g04	-30665.539	-30665.480371	-30483.474609	-30638.775977	-30659.656250	-30659.845703
g06	-6961.81388	-6928.352727**	-6622.280273	-6941.932129	-6956.971680	-6957.950684
g07	24.306	24.651254	468.216675	28.631790	26.232813	27.512201
g08	0.095825	0.095825	0.095813	0.095826	0.095826	0.095825
g09	680.630	680.739309	723.854919	693.252319	680.872986	681.324036
g10	7049.331	7101.708902	11129.170898	11259.611328	8812.435547	7372.459961

提案手法での初期値はパラメータ領域の中間値とした　* 数学的最適解は知られていない (T.P.Runarsson et.al., 2000)
** 複数の初期値を与え初期値依存性を低減することで-6961.484605 を得る

3.2 ベンチマーク結果および考察

テスト関数 [2] を用いて得られる解を，大域的最適解と併せて表 1 に示す．多くの問題に対して，大域的最適解もしくは近傍解を得ているが，以下では，最適解を得るための工夫をテスト関数 $g02$ を例に考察する．

テスト関数 $g02$ は，周期的な非線形関数であり，数学的な大域的最適解は知られていない (Keane, 1994)．この関数は多峰性関数であり，局所解に陥りやすい典型例である．

今回得られた解は表 1 に示す大域的最適解に到達していないが，その原因として以下の二点が考えられる．

第一点は，アルゴリズムに焼きなまし法を用いており，そのパラメータ初期値依存性によって大域的最適解に至っていない点である．焼きなまし法での初期温度を上げることは対策の一つではあるが，上げ過ぎても探索が安定しない．この対策としては，複数の初期値による複数探索を実施すればよい．

第二点はテスト関数が特徴的であり，期待する重み付けが困難である点である．提案する自動重み付け手法では，評価コスト関数 $f(x)$ とペナルティ関数 $g(x)$ の近似曲線上の各代表点の比を重みに代用している．このため，提案手法は，パラメータ変動範囲において各近似曲線が長い周期で変動する場合に限り，変動の少ない重みを決定できる．しかし，各関数が短い周期で変動する場合には，滑らかな近似曲線はその振動に追随できず，近似曲線上の点が代表点とはならないことは自明である．よって，初期パラメータが最適解近傍であれば，単調増加もしくは減少で最適解を見つけることができるが，それ以外では短い周期で振動するため，適切な重みを設定できないと考えられる．

今回，探索によって得られた解を再度初期値とし，更に初期焼きなまし温度を低下させて探索範囲を狭めることで，評価コスト関数の多峰性を極力排除し，大域的最適解近傍解を得た．

どんな問題に対しても大域的最適解を得るアルゴリズムは存在しない．しかし，特に工学分野においては，数学的に最適解でなくとも，現状を改善した解が得られれば目的が達成される場合が多い．また，軽微な変更の場合には経験的に最適解近傍の解はわかっているため，目的関数の特性も既知である．よって，本研究で示したように，局所解に陥りやすい目的関数を扱う場合は，局所解から抜け出す工夫を別途加えることで，より大域的最適解に近い解を得ることができる．

4 まとめ

本研究では，ペナルティ法を用いた目的関数の作成手法を提案し，テスト関数を用いてその性能を評価した．提案する目的関数は簡素であるものの，局所解に陥らないガイドとして機能し，最適解もしくはその近傍解を探索できることを示した．

また，関数によっては局所解に陥りやすい場合もある．その対策として，得られる解の初期値依存性を低減するためには，広範な初期値による複数探索を実施し，制約条件を満足しにくい場合には，ペナルティ関数に更に定数を乗じる操作が有効である．

昨今では，SA を多目的最適化問題に適用するために改良したパレート SA などが登場している．しかし，重み変更の割合を事前にユーザーが設定しなければならないなど，ユーザーにある程度の手法に関する知識が要求される．それに対し，提案手法では，最小化したい関数および制約条件とパラメータ範囲を入力するのみで実現可能な最適解もしくはその近傍解を得ることができるため，よりユーザフレンドリーな設計ツールに適用できると考えられる．

参考文献

[1] I. J. Schoenberg, Spline functions and the problem of graduation, Proceedings of the National Academy of Sciences of the U.S.A., 52, pp.947–950 (1964).

[2] Zbigniew Michalewicz, et.al., Evolutionary Algolithms for Constrained Parameter Optimization Problems, Evolutionary Computation, 4(1), pp.1–32 (1996).

[3] Arturo Hernandez Aguirre et.al., Handling Constraints using Multiobjective Optimization Concepts, Int. J. Numer. Meth. Engng, Vol.59, 15 , pp.1989–2017 (2004).

[4] Lester Ingber, Adaptive Simulated Annealing (ASA), Lester Ingber Research, McLean, VA, (1993).