

KONAN UNIVERSITY

モデル予測多目的最適化

著者	中山 弘隆
雑誌名	甲南大学紀要. 知能情報学編
巻	1
号	1
ページ	1-22
発行年	2008-07-20
URL	http://doi.org/10.14990/00001255

総説

モデル予測多目的最適化

中山弘隆

甲南大学 知能情報学部 知能情報学科
神戸市東灘区岡本 8-9-1, 658-8501

(受理日 2008年5月28日)

概要

現実の工学設計問題では評価関数が設計変数の陽な関数として与えず、種々のシミュレーションや実際の実験を通じてその値が得られる場合が多い。通常、このようなシミュレーションや実験には多大の計算時間あるいはコストを必要とする。このようなシミュレーションや実験の回数をできる限り少なくするために、評価関数形を近似しながら最適化を行う方法がモデル予測最適化である。

一方、実際の意味決定問題ではあれもこれもよくしたいという多目的最適化問題になることが多い。多目的最適化においては、「あちらをたてればこちらがたたず」といった目的関数をギリギリまで高めた解を Pareto 解といい、通常これを意思決定の解の候補とする。Pareto 解は一般に唯一に定まらず、集合となる。これを Pareto フロンティアといい、実際の意味決定では、Pareto フロンティアからどの解の一つ選ぶかが重要である。古くからある対話型解法はいかに効率よく意思決定者の望む解を得るかに腐心している。

本論文では、多目的最適化に対するモデル予測最適化手法について解説を行う。この中で、モデル予測のための手法として、最近の話題になっている機械学習の方法であるサポートベクターマシン (Support Vector Machine: 略して SVM) について多目的最適化のアプローチからいくつかの方法が開発されていることを紹介する。さらに、Pareto フロンティアの近似生成のための進化型計算法や他の方法についても言及する。

キーワード: 多目的最適化, 計算知能, サポートベクターマシン, サポートベクター回帰, 逐次近似最適化

1 多目的最適化における予備知識

多目的最適化問題は、一般に次のように定式化される。

$$\begin{aligned} \text{(MOP)} \quad & \text{目的: } f(\mathbf{x}) := (f_1(\mathbf{x}), \dots, f_r(\mathbf{x})) \longrightarrow \min \\ & \text{制約: } \mathbf{x} \in X \subset \mathbb{R}^n. \end{aligned}$$

定義 1.1 (ベクトル不等式). $\mathbf{y}^1, \mathbf{y}^2 \in \mathbb{R}^r$ に対し

$$\mathbf{y}^1 < \mathbf{y}^2 \iff y_i^1 < y_i^2, \quad \forall i = 1, \dots, r \quad (1)$$

$$\mathbf{y}^1 \leq \mathbf{y}^2 \iff y_i^1 \leq y_i^2, \quad \forall i = 1, \dots, r \quad (2)$$

$$\mathbf{y}^1 \leq \mathbf{y}^2 \iff \mathbf{y}^1 \leq \mathbf{y}^2, \quad \mathbf{y}^1 \neq \mathbf{y}^2 \quad (3)$$

定義 1.2 (Pareto 解). $f(x) \leq f(\hat{x})$ となるような $x \in X$ が存在しないとき、この \hat{x} を **Pareto 最適**、あるいは単に **Pareto 解** という。

今後の分析には設計変数 x の空間で考えるよりも、図 1 のように目的関数 f の空間で考える方が便利ことが多い。 \hat{x} が Pareto 解であるとは幾何学的には $\hat{y}(= f(\hat{x}))$ に対し $\hat{y} + R_-^n$ が \hat{y} 以外に $f(X)$ と共通点を持たないことである。ただし、 $R_-^n := \{y \in R^n \mid y \leq 0\}$ 。

目的関数の空間における Pareto 解の集合を **Pareto フロンティア** あるいは**効率フロンティア** という。実際の意思決定では最終的には何か 1 つの決定案を決めなければならないから、通常は各目的のバランスを考慮してこの Pareto 解の中から 1 つの最終案が選ばれる。

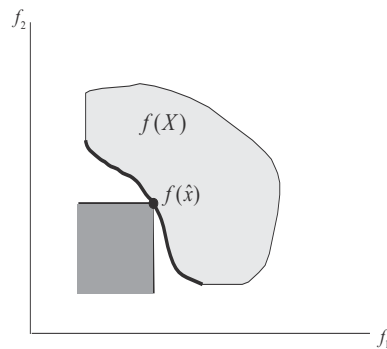


図 1: Pareto 解 (太線の部分)

注 1.1. $f(x) < f(\hat{x})$ となるような $x \in X$ が存在しないとき、この \hat{x} を**弱 Pareto 最適**、あるいは単に**弱 Pareto 解** (weak Pareto solution) という。 \hat{x} が弱 Pareto 解であるとは幾何学的には $\hat{y}(= f(\hat{x}))$ に対し $\hat{y} + \text{int } R_-^n$ が $f(X)$ と共通点をもたないことである。図 3 のように、弱 Pareto 解は一部の目的関数とその値に留めておいて、他の目的関数を改善できる余地があるので、意思決定の観点からは必ずしも満足できる解ではないが、理論上、弱 Pareto 最適性しか保証できないことがしばしば起こる。

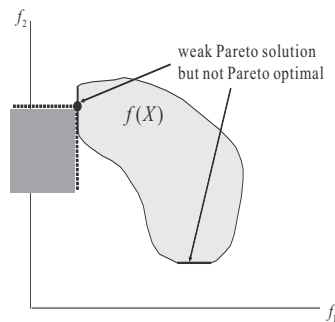


図 2: 弱 Pareto 解

Pareto 解を求めるために古くから最も良く用いられている方法が**スカラー化**である。多目的最適

化における目的関数はベクトル値であるので、これをいったんスカラー値に変換する。このスカラー化目的関数に対する本質的な性質は次の定理によって与えられる。

定理 1.1. F をベクトル値目的関数 $\mathbf{y} = \mathbf{f}(\mathbf{x})$ をスカラー化するスカラー化関数としよう。 F が \mathbf{y} に関して Pareto 順序を保存するとき、すなわち、任意の $\mathbf{y}^1, \mathbf{y}^2 \in \mathbf{f}(X)$ に対して

$$\mathbf{y}^1 \leq \mathbf{y}^2 \Rightarrow F(\mathbf{y}^1) < F(\mathbf{y}^2) \tag{4}$$

を満たすとき、 F を X 上で最小化する解 \mathbf{x}^0 は Pareto 解である。

代表的なスカラー化関数の例としては

- 1) $F_1(\mathbf{f}(\mathbf{x})) = w_1 f_1(\mathbf{x}) + \dots + w_r f_r(\mathbf{x}),$
- 2) $F_\infty(\mathbf{f}(\mathbf{x})) = \max_{1 \leq i \leq r} w_i f_i(\mathbf{x}),$
- 3) $\tilde{F}_\infty(\mathbf{f}(\mathbf{x})) = \max_{1 \leq i \leq r} w_i f_i(\mathbf{x}) + \alpha \sum_{i=1}^r w_i f_i(\mathbf{x})$

などがある。 1) は線形加重和、 2) は Tchebyshev スカラー化関数、 3) は拡大 Tchebyshev スカラー化関数と呼ばれている。

1) の線形加重和は Pareto フロンティアが非凸な場合、窪んだ所にある Pareto 解を抽出できないこと、 2) の Tchebyshev スカラー化関数では抽出した解が弱 Pareto 解である可能性があること、 3) の拡大 Tchebyshev スカラー化関数が最も良く用いられる。

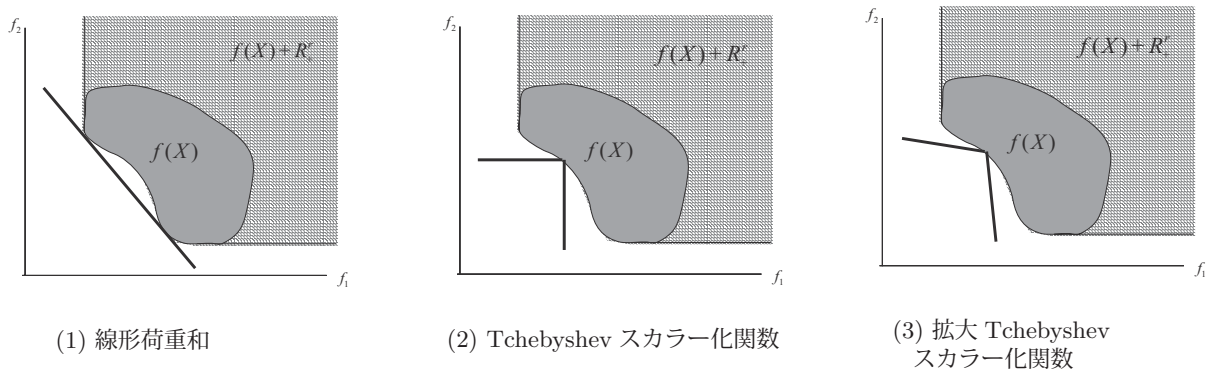


図 3: 弱 Pareto 解

2 対話型解法

Pareto 解は「あちらを立てればこちらが立たず」といった状態であるが、実際の意思決定では Pareto 解の中から目的関数のバランスを考えて一つの決定解を選ぶ必要がある。このとき、「あちらをどの程度立て、こちらをどの程度犠牲にするか」を考えるトレードオフ分析が重要となる。多目的意思決定においてはこのトレードオフ分析をいかにやりやすくして、意思決定者が思いのままの解を簡単に得られるようにするのが最も大きな課題である。

対話型解法においては意思決定者の価値観に関する情報として何をとるかが重要となる。通常、スカラー化関数として線形加重和をとり、意思決定者の価値観に関する情報として重みをとることが多いが、これには重大な欠陥がある。非凸な問題に適用できないことはもちろん、重みを大きくすれば対応する目的関数が改善されるという誤った常識によっていわゆるモグラたたきの現象が起り、なかなか思い通りの解が得られないことが日常的に起こっている。このような問題を克服するものとして現在では意思決定者の価値観に関する情報として各目的に対する希求水準をとった希求水準法がよく用いられている。以下に、希求水準法の中の一つである満足化トレードオフ法 [8] を紹介する。

満足化トレードオフ法では意思決定者の望み通りの解が得られるまで、希求水準を変更する。アルゴリズムはおおまかにいって次の2つのステップからなりたつ。

Step 1. (補助最適化) 与えられた希求水準 \bar{f}_i ($i = 1, \dots, r$) に対し、次のような補助最適化問題を解く。

$$\begin{aligned} \text{(AP)} \quad \text{目的} &: z + \alpha \sum_{i=1}^r w_i f_i(x) \rightarrow \min \\ \text{制約} &: w_i^k (f_i(x) - \bar{f}_i^k) \leq z \quad (i = 1, \dots, r) \\ &: x \in X. \end{aligned}$$

Step 2. (トレードオフ) 示された解に納得できないとき、全体のバランスを考え、意思決定者は希求水準を設定し直す。現在の解はすでに Pareto 最適だから、ある目的を改善しようと思えば、別の目的を犠牲にせざるをえない。また、現在の解はある程度最初の価値判断に沿ったものであるから、その解に納得できないといっても、普通は新たな希望値は現在の解から極端に離れるということはない。あちらをどの程度たて、こちらをどの程度犠牲にするかというトレードオフ分析は現在の解を基準にしてそこからどう動くかを判断すればよい。

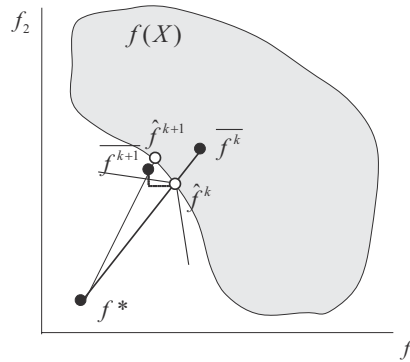


図 4: 満足化トレードオフ法

注 2.1. 補助最適化問題 (AP) はもともと拡大 Tchebyshev スカラー化関数最小化

$$\max_{1 \leq i \leq r} w_i (f_i(x) - \bar{f}_i) + \alpha \sum_{i=1}^r w_i f_i(x) \rightarrow \min. \quad (5)$$

であるが、このままでは目的関数が滑らかでないため、等価な滑らかな問題に変換したものである。さて、第 i 番目の目的関数に対する理想値を f_i^* としよう。これは、たとえば f_i の値が小さいほどよいときには

$$f_i^* < \min\{f_i(x) \mid x \in X\}$$

で定めてもよいし、経験上分かっていたらそれを用いてもよい。このとき、 w_i^k を

$$w_i^k = \frac{1}{\bar{f}_i^k - f_i^*} \quad (6)$$

とすることができる。このようにすれば、(5) 式の第 1 項は理想点 f_i^* を基準として測った希求水準 \bar{f}_i までの不達成度とみなすことができる。ここで、(5) 式第 1 項にマイナス値をとることを許しているから、(5) 式を最小にするということは希求水準を達成してもなおかつできる限り f_i の値が小さくなるように努力することになる。これが Pareto 最適性を保証する大きな原因となる。

注 2.2. 目的関数の数が多くなると全ての目的関数に対し、改善量や犠牲量を答えるのは意思決定者にとって大きな負担となる。そこで、意思決定者にとってどちらかといえば改善すべき目的の方が重要であるから、改善量だけ答えてもらい、犠牲にすべき量は数理計画の感度解析の理論やパラメトリック分析の手法を用いて自動的に計算する方法も提案されている [9]。

満足化トレードオフ法のこれまでの応用例としては、斜張橋架設時精度管理支援システム、債券ポートフォリオ決定支援システム、飼料配合設計支援システム、セメントの原料石最適配合、プラスチック成形材の原料調合、鉄鋼製造プロセスにおけるトライ選択問題、射線外照射治療計画、広域水運用計画への対話型多目的計画法の応用等がある [10, 18]。

3 Pareto フロントニアの生成法

遺伝アルゴリズム等の進化型計算法を多目的最適化における Pareto フロントニアの近似生成に適用したのは 1985 年の Shaffer による VEGA (Vector Evaluated Genetic Algorithm) が最初とされるが、本格的に研究されたのは 1989 年の Goldberg や、1993 年の Fonseca-Fleming によるランクという概念が出されてからである。ランクの概念は Pareto 順序に基づき他の点に比べてどれだけ優位にあるかの指標で、注目している点よりも優れている点他に何個あるか、あるいはその注目している点が優越している他の点は何個あるかを数えてランクを決める。このランクにより正確な Pareto フロントニアにどれだけ近いか（すなわち、convergence）を考慮するが、Pareto フロントニア全域を万遍なく近似する（divergence）ということも重要である。この divergence のために、ニッチとシェアリングという概念が考えられている。ニッチは注目している点の近くに（すなわち、ニッチ半径と呼ばれる一定距離内に）どれだけ他の点があるかを表すもので、シェアリングは適合度を近傍にある点でシェアすることにより、それらの適合度を減少させるものである。これにより、疎な Pareto 解の適合度が高くなり、Pareto 解が密集している所よりも優先される。

以上が EMO (Evolutionary Multi-Objective Optimization) と呼ばれる進化型計算法による多目的最適化の基本アイデアであるが、convergence と divergence の観点から種々の工夫が施されて、現在では Deb らによる NSGA (Non-dominated Sorting Genetic algorithm)、およびその改良版 NSGA-II や

Zitzler-Thiele による SPEA(Strength Pareto Evolutionary Algorithm), およびその改良版 SPEA2 がよく用いられている [2, 3]. 進化型計算法の適用以外にも, DEA(Data Envelopment Analysis) あるいはそれを一般化した GDEA(Generalized Data Envelopment Analysis) を用いる方法 [26] や Pareto フロンティアの近似生成を 1 クラス分類の問題と見なし SVM(Support Vector Machine) を適用する方法 [27] も考えられている.

Pareto フロンティアの近似生成は, 2 個の目的関数のときは図示された Pareto フロンティアによって目的間のトレードオフ状況が一目で分かるので意思決定にも役立つ. しかし, 3 個の目的関数になると Pareto フロンティアを図示することはできても目的間のトレードオフ状況を把握することが困難になる上, それ以上の目的関数になると図示することさえできなくなる. 通常, 意思決定の解は Pareto フロンティアから一つ選ぶ必要があり, 意思決定の立場からは Pareto フロンティアを求めてからその後どうするかが問題となる.

実際の多くの意思決定問題では意思決定者は必ずしも Pareto フロンティア全域に関心があるわけではなく, Pareto フロンティアをこの 関心ある領域で近似生成すれば十分と考えるアプローチも出てきている. 多くの場合, この関心ある領域は意思決定者の希求水準に最も近い領域になるので, ここに希求水準法との融合が考えられる. これについては, 第 6 章で詳述する. また, 多数ある目的のトレードオフ分析については [26] を参照.

4 モデル予測最適化

実際の工学設計の問題では目的関数が設計変数の陽な関数として表すことができず, 設計変数の値を決めたとき, 構造解析, 熱解析, 流体解析, 振動解析等の数値シミュレーションや実モデルでの実験を行うことにより初めて目的関数の値が求められることが多い. 通常, このような数値シミュレーションや実験には多大の時間やコストを要する. 多目的最適化を実際の工学設計に応用しようとするとき, これらのシミュレーションや実験の回数はできる限り少ない方がよい. このために目的関数や制約関数を近似しながら最適解を求めようとする方法が研究されている. これらはメタモデリング, 逐次近似最適化, モデル予測最適化等, いくつかの名前で呼ばれているが, 基本的には同じである.

《近似関数の生成》

近似関数の生成にはこれまで最適な実験配置によってより少ない実験点で精度良く近似関数を生成しようとする応答曲面法がよく知られている. サンプル点での関数値の測定にノイズを含む通常の物理的実験を元々の考察の対象とする実験計画法では誤差を同分布に従う独立な確率変数とすることが多く, また, 近似関数は多項式に取られることが多い. 一方, コンピュータ実験のように関数値の評価にノイズを含まない場合には, 地質調査に端を発する Kriging 法を用いる研究も盛んである. Kriging 法では異なるサンプル点での誤差は互いに独立ではなく, 相関をもち, それらのサンプル点の距離が近いほどその相関も高いと仮定される. 近似関数としては一定値プラスそれからの誤差(乖離)とする場合 (ordinary Kriging) や, 適当な基底関数による線形回帰関数プラス誤差を用いる場合 (universal Kriging) などがある. また, RBF ネットワークやサポートベクターマシンによる回帰 (SVR) 等の計算知能を用いた回帰も良く用いられる [15, 16].

《追加点の選定》

関数の評価回数をできる限り少なくするために、少なめのサンプル点から出発し、十分な近似最適解が得られるまで、少しずつサンプル点を追加し、それに基づいて近似関数の修正を行うということで近似精度を向上させる。モデル予測最適化における最大のポイントはいかにうまくこの追加サンプル点を定めるかにある。

もし、単に近似目的関数の最適値を追加サンプル点とすると、初期近似で得られた関数の最適値付近のみを追加し続け、正しい最適値とは異なる極値に収束する可能性が高い。これは目的関数の近似に対し、大局的情報が不足するのが原因である。また、大域的な情報のみを求めて、予測した最適値付近の点を追加しないでいると、今度は逆に、最適値付近の詳細な情報を得ることができなくなり、求めたい最適値を精度良く見つけだすことが困難になる。そこで、局所情報と大局情報を同時に追加する方法が提案されている [12].

伝統的によく用いられている応答曲面法は比較的低次の多項式を用い、実験計画法の理論に基づいて近似関数を生成している。追加点の選定には実験計画法における D 最適性や A 最適性を用いることが多いが、追加点の一部の領域に偏ってしまう欠点がある。

1998 年に提案された EGO (Efficient Global optimization) においては、Kriging 法による期待改善量 (Expected Improvement; EI) が用いられている。EI とは、目的関数の最小値をとる可能性のある部分、またはサンプル点の密度が疎な部分に高い値を与える指標である。また、あるパラメータ g を導入することによりこれらのバランスを調節することも提案されている [22].

以上の他にも、機械学習理論において予測点での汎化誤差を最小にするという観点からの能動学習もある [25].

《近似関数に対する最適化》

近似された関数は通常は滑らかな関数であるので、その最適化には微分情報を用いる通常の最適化手法の他、遺伝アルゴリズムや PSO (Particle Swarm Optimization) 等のメタヒューリスティック手法が適用可能であり、問題に応じて使い分ければよい。

5 非線形関数の効率的な近似

ここでは一般の非線形関数の近似に有効な RBF (Radial Basis Function) ネットワークとサポートベクターマシン (Support Vector Machine, 略して SVM) について簡単に紹介する。

5.1 RBF ネットワーク

RBF ネットワークは、入力層 (素子数 n 個)、中間層 (素子数 m 個)、出力層 (本論文では 1 個) の 3 層からなる階層型ニューラルネットワークの 1 つである。 j 番目の中間素子の出力関数として主にガウス関数

$$h_j(\mathbf{x}) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{c}_j\|^2}{r^2}\right)$$

を用いる。出力素子からの出力は、中間層素子の出力の加重和

$$O(\mathbf{x}) = \sum_{j=1}^m w_j h_j(\mathbf{x})$$

である。RBF ネットワークの学習は、 \mathbf{x}_i における教師値 y_i ($i = 1, \dots, p$) とネットワークの出力値との差の2乗和と結合係数 w の抑制項を加えた式

$$Cost = \sum_{i=1}^p (y_i - O_i(\mathbf{x}))^2 + \lambda \sum_{j=1}^m w_j^2$$

が最小となるような $\mathbf{w} = (w_1, \dots, w_m)$ を求めることである。

ここで、

$$H_p^T = [\mathbf{h}_1 \ \dots \ \mathbf{h}_p]$$

で、 $\mathbf{h}_j^T = [h_1(\mathbf{x}_j), \dots, h_m(\mathbf{x}_j)]$, および Λ は λ を対角成分にもつ対角行列とすると、

$$A = (H_p^T H_p + \Lambda)$$

とおけば、上記最小化の必要条件として、 $\mathbf{y} = (y_1, \dots, y_p)$ とするとき

$$A\hat{\mathbf{w}} = H_p^T \mathbf{y}$$

を得る。したがって、RBF ネットワークにおける学習は逆行列

$$A^{-1} = (H_p^T H_p + \Lambda)^{-1}$$

を求めることに帰着される。

注 5.1. RBF ネットワークにおいて、パラメータ \mathbf{c}_j , λ および r の値を適切に定めることは重要である。ここでは、各学習点 \mathbf{x}_i ($i = 1, \dots, p$) の位置に基底関数を置くことにする。すなわち、 $\mathbf{c}_i = \mathbf{x}_i$ ($i = 1, \dots, p$)。さらに r および λ に関しては、通常、交差検証法 (cross validation) によって定めるが、簡易法として [7] で与えられている目安の式を若干修正した

$$r = \frac{d_{max}}{\sqrt{nm}}$$

による方法も提案されている [13]。ただし、 d_{max} は学習点間の最大距離、 n は学習点の次元、 m は基底関数の数である。他に、非線形最適化の手法を用いて基底毎の最適な r を求める方法も提案されている [1]。

5.2 サポートベクターマシン

サポートベクターマシン (Support Vector Machine: 略して SVM) は、近年マシンラーニングとくにパターン分類や関数近似の効果的な方法として注目を集めている。この方法におけるキーワードは最大マージン分離超平面、カーネルトリックである。以下、これらについて簡単に述べよう。

5.2.1 パターン分類

《最大マージン分離超平面》

今、パターン (\mathbf{x}_i, y_i) ($i = 1, \dots, \ell$) に対し y_i が +1 かあるいは -1 のいずれかをとるという 2 値のカテゴリ分類問題を考える。ここで、各 \mathbf{x}_i は n 次元ベクトルである。与えられた問題は一般には線形分離可能ではない。そこで、何らかの非線形写像 $\Phi: R^n \rightarrow R^q$ ($q = \infty$ も可) を行い線形分離可能になったとする。このとき、 $\mathbf{z} = \Phi(\mathbf{x})$ としよう。写像先の空間 R^q (これを特徴空間という) における超平面

$$H(\mathbf{z}) = \mathbf{w}^T \mathbf{z} + b$$

によって

$$H(\mathbf{z}) \geq 0 \quad \text{for } \mathbf{z} \in \mathcal{A} = \{\mathbf{z}_i \mid y_i = 1\}$$

$$H(\mathbf{z}) \leq 0 \quad \text{for } \mathbf{z} \in \mathcal{B} = \{\mathbf{z}_i \mid y_i = -1\}$$

のように分離されるとする。このような分離超平面は一般に複数個存在するが、その中でも、分離超平面と \mathcal{A} および \mathcal{B} との距離 (マージン) ができる限り大きいものが望ましい。 \mathbf{w} および b には同じスカラー倍の恣意性が残るので、

$$\min_{i=1, \dots, \ell} y_i (\mathbf{w}^T \mathbf{z}_i + b) = 1$$

によって正規化することにより、学習点と超平面の最小距離は $1/\|\mathbf{w}\|$ となる。このとき、所望の性質をもつ分離超平面は次の最適化問題

$$\begin{aligned} (\text{SVM}_P) \quad & \text{目的} : \quad \frac{1}{2} \mathbf{w}^T \mathbf{w} \rightarrow \text{Min} \\ & \text{制約} : \quad y_i (\mathbf{w}^T \mathbf{z}_i + b) \geq 1 \quad (i = 1, \dots, \ell) \end{aligned}$$

を解くことにより得られる。

問題 (SVM_P) の双対問題は上記制約に対するラグランジュ乗数を α_i とすると

$$\begin{aligned} (\text{SVM}_D) \quad & \text{目的} : \quad \sum_{i=1}^{\ell} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{\ell} \alpha_i \alpha_j y_i y_j \mathbf{z}_i^T \mathbf{z}_j \rightarrow \text{Max} \\ & \text{制約} : \quad \alpha_i \geq 0 \quad (i = 1, \dots, \ell), \quad \sum_{i=1}^{\ell} \alpha_i y_i = 0 \end{aligned}$$

となる。明らかに $\alpha_i^* \neq 0$ となる i に対しては (SVM_P) における制約がアクティブ、すなわちそのときの \mathbf{z}_i が分離超平面に対し最短距離 (=1) を与える点となっている。このような点 \mathbf{z}_i およびその原像 \mathbf{x}_i のことをサポートベクトル (support vector) と呼んでいる。このようなサポートベクトルの添字集合を S で表すことにしよう。このとき、(SVM_P) に対する必要条件から分離超平面の法線ベクトル \mathbf{w}^* は

$$\mathbf{w}^* = \sum_{i=1}^{\ell} y_i \alpha_i^* \mathbf{z}_i = \sum_{i \in S} y_i \alpha_i^* \mathbf{z}_i$$

によって与えられ、また b は

$$b^* = -\frac{\max_{y_i=-1} \mathbf{w}^{*T} \mathbf{z}_i + \min_{y_i=1} \mathbf{w}^{*T} \mathbf{z}_i}{2}$$

で与えられる。

《カーネルトリック》

写像 Φ は像空間において線形分離ができる限り可能になるように一般には像空間が高次元（無次元も可）になるようにとられる。このとき、(SVM_D)における目的関数には高次元ベクトルの内積 $\mathbf{z}_i^T \mathbf{z}_j$ がでてきて、計算に負荷がかかる。そこで、原空間の距離を像空間の内積として保存するようなカーネル関数

$$K(\mathbf{x}, \mathbf{x}') = \Phi(\mathbf{x})^T \Phi(\mathbf{x}') \quad (7)$$

を導入する。 $K(\mathbf{x}, \mathbf{x}')$ が正定であればこのような関係を満たす写像 Φ が存在することが知られている (Mercer の定理)。上記のようなカーネル $K(\mathbf{x}, \mathbf{x}')$ を用いることにより (SVM_D) は高次元ベクトルの内積の計算をすることなくカーネルの値を求めるだけで簡単に最適化を行うことができる。また、像空間における分離超平面は $\mathbf{z} = \Phi(\mathbf{x})$ とおくと

$$\begin{aligned} H(\mathbf{z}) &= \mathbf{w}^{*T} \mathbf{z} + b^* \\ &= \sum_{i \in S} y_i \alpha_i^* \Phi(\mathbf{x})^T \Phi(\mathbf{x}_i) + b^* \\ &= \sum_{i \in S} y_i \alpha_i^* K(\mathbf{x}, \mathbf{x}_i) + b^* \end{aligned}$$

で与えられる。このように具体的に写像 Φ を用いることなく、カーネルの値のみによって Φ の像空間における分離超平面を求めることはカーネルトリックと呼ばれている。

注 5.2. Φ の像空間の次元を非常に大きくとることによって、この空間での A および B は線形分離可能になることが期待できるが、これは必ずしも保証されるものではない。線形分離可能でなければ (SVM_P) の制約は満足されない。また、データは通常ノイズを含んでいるため、完全分離する超平面は時として過学習となりマシンラーニングの汎化能力の点から必ずしも歓迎すべきことではない。これらのことを考慮に入れて、制約が必ずしも完全に満たされていなくてもよいようにすることが考えられている。すなわち、スラック変数 ξ_i ($i = 1, \dots, \ell$) を導入して制約を

$$y_i(\mathbf{w}^T \mathbf{z}_i + b) + \xi_i \geq 1 \quad (i = 1, \dots, \ell)$$

のように緩和し、目的を

$$\frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{i=1}^{\ell} \xi_i \rightarrow \text{Min}$$

とすればよい。この定式化をソフトマージン SVM という。

5.2.2 関数近似への応用

パターン分類の問題では各 y_i の値は ± 1 であった。SVM を関数近似の問題に適用するには y_i を \mathbf{x}_i におけるその関数値（もしくはその観測値）とし、 \mathbf{z} の空間すなわち特徴空間において線形回帰とする。このとき、制約を

$$\begin{aligned} (\mathbf{w}^T \mathbf{z}_i + b) - y_i &\leq \epsilon + \xi_i \\ y_i - (\mathbf{w}^T \mathbf{z}_i + b) &\leq \epsilon + \eta_i \\ \xi_i, \eta_i &\geq 0 \quad (i = 1, \dots, \ell) \end{aligned}$$

とし、目的を上記ソフトマージンの時と同じようにすればよい。 ϵ は誤差の許容幅とみなせるが、これを導入しなければすべてのサンプル点がサポートベクトルとなり、スパース性に欠ける。

5.2.3 カーネルのとりかた

SVM におけるカーネルとしては種々のものが提案されているが、ここでは多項式カーネル

$$K(\mathbf{x}, \mathbf{x}') = (\mathbf{x}^T \mathbf{x}' + 1)^\gamma$$

および、ガウシアンカーネル

$$K(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{r^2}\right)$$

をあげておこう。

ガウシアンカーネルを用いた SVM は一見 RBF ネットワークと同じに見えるが、基底を置く場所がサポートベクトルの点として自動的に出てくるところが大きく異なる。

5.2.4 μ - ν -SVM

ソフトマージン SVM ではスラック変数、すなわち外部偏差 ξ_i を導入していた。しかし、制約を満足していてもさらに分離超平面からなるべく遠くにサンプル点がある方が分離性能がよいと考えられる。このため、内部偏差（すなわち余剰変数） η_i を導入する。このとき、つぎのような目的が考えられる。

- i) 最大外部偏差を最小化
- ii) 最小内部偏差を最大化
- iii) 内部偏差の和を最大化
- iv) 外部偏差の和を最小化

(iv) を採用したのがソフトマージン SVM であり, (ii) と (iv) を採用すれば,

$$\begin{aligned} \min \quad & \frac{1}{2} \|\mathbf{w}\|_2^2 - \nu\rho + \frac{1}{\ell} \sum_{i=1}^{\ell} \xi_i & (\nu\text{-SVM})_P \\ \text{s.t.} \quad & y_i (\mathbf{w}^T \mathbf{z}_i + b) \geq \rho - \xi_i, \\ & \rho \geq 0, \xi_i \geq 0, i = 1, \dots, \ell. \end{aligned}$$

となる. この定式化は Schölkopf et al. [20] によって提案された ν -SVM と同じであり, ここでは $0 \leq \nu \leq 1$ はサポートベクトルの割合を反映させるパラメータとなっている.

(iii), (iv) を考慮したトータルマージン SVM や (i), (iii) を考慮した μ -SVM も考えられているが, ここでは (i), (ii) を考慮した μ - ν -SVM の定式化を下に記す.

$$\begin{aligned} \min \quad & \frac{1}{2} \|\mathbf{w}\|_2^2 - \nu\rho + \mu\sigma & (\mu\nu\text{-SVM})_P \\ \text{s.t.} \quad & y_i (\mathbf{w}^T \mathbf{z}_i + b) \geq \rho - \sigma, i = 1, \dots, \ell, \\ & \rho \geq 0, \sigma \geq 0. \end{aligned}$$

この双対問題は

$$\begin{aligned} \max \quad & -\frac{1}{2} \sum_{i,j=1}^{\ell} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) & (\mu\nu\text{-SVM}) \\ \text{s.t.} \quad & \sum_{i=1}^{\ell} \alpha_i y_i = 0, \\ & \nu \leq \sum_{i=1}^{\ell} \alpha_i \leq \mu, \\ & \alpha_i \geq 0, i = 1, \dots, \ell. \end{aligned}$$

で与えられる.

多くの実験例で, μ - ν -SVM は数ある SVM の定式化の中でサポートベクトルの数が最も少ないことが知られている.

5.2.5 μ - ν -SVR

μ - ν -SVM を関数近似の問題に適用するには, 次の μ - ν -SVR のようにすればよい.

$$\begin{aligned} \min \quad & \frac{1}{2} \|\mathbf{w}\|_2^2 + \nu\varepsilon + \mu(\xi + \xi') & (\mu\nu\text{-SVR})_P \\ \text{s.t.} \quad & (\mathbf{w}^T \mathbf{z}_i + b) - y_i \leq \varepsilon + \xi, i = 1, \dots, \ell, \\ & y_i - (\mathbf{w}^T \mathbf{z}_i + b) \leq \varepsilon + \xi', i = 1, \dots, \ell, \\ & \varepsilon, \xi, \xi' \geq 0, \end{aligned}$$

ここで ν および μ はそれぞれ $\|\mathbf{w}\|$ と ε , ξ の間のトレードオフ比である.

双対問題としての定式化は次のようになる.

$$\begin{aligned} \max \quad & -\frac{1}{2} \sum_{i,j=1}^{\ell} (\alpha'_i - \alpha_i) (\alpha'_j - \alpha_j) K(\mathbf{x}_i, \mathbf{x}_j) + \sum_{i=1}^{\ell} (\alpha'_i - \alpha_i) y_i \\ \text{s.t.} \quad & \sum_{i=1}^{\ell} (\alpha'_i - \alpha_i) = 0, \\ & \sum_{i=1}^{\ell} \alpha'_i \leq \mu, \quad \sum_{i=1}^{\ell} \alpha_i \leq \mu, \\ & \sum_{i=1}^{\ell} (\alpha'_i + \alpha_i) \leq \nu, \\ & \alpha'_i \geq 0, \quad \alpha_i \geq 0, \quad i = 1, \dots, \ell. \end{aligned} \tag{\mu-\nu-SVR}$$

注 5.3. RBF ネットワークや SVM をはじめとして数多くのマシンラーニングの手法が提案されているが、性能だけを問題にするならば、パラメータチューニングさえうまくやればどれもほとんど同じ性能まで高めることができる。したがって、ラーニングマシンの性能を論ずるとき、高性能を実現するためのパラメータチューニングがやりやすいかどうか、また最適パラメータの感度が低く安定しているかどうかを見るのが重要である。RBF ネットワークでは c_j および λ の感度はそれほど大きくはないが、 r によっては大きく性能が変わってくる。しかし、ひとたびこれらのパラメータを決めてしまうと、その学習は単に正規方程式を解くという線形計算であるので、計算が非常に速い。一方、SVM では通常は 2 次計画問題を解く必要があり、一般に計算コストは高い。パラメータ r の重要性は RBF ネットワークと同様である。しかし、基底を置く場所がサポートベクトルの所だけでよい、ということは要はサポートベクトルはいわばツボの役割を果たしているということになり、この性質をうまく使うと実際の問題によっては大きくコストを削減することが出来る。

6 逐次近似多目的最適化 –対話型解法と進化型解法の融合–

多目的最適化においては Pareto フロンティア全体を十分に良く近似しようという研究も盛んであるが、実際の意思決定においては意思決定者が関心を持つ領域は必ずしも Pareto フロンティア全体ではないことが多い。一方、伝統的対話型解法を用いた場合にも、トレードオフ分析を行うときには目的間のトレードオフ状況を大まかでも把握したいことが多い。伝統的対話型解法では与えられた希求水準に最も近い Pareto 解を提示したときに、その Pareto 解の近傍でのトレードオフ状況を Lagrange 乗数の値から局所的に与えることはできるが、その有効性はその Pareto 解の極めて小さな近傍に限定される。そこで、これらの両者のアプローチの特長を生かした方法も提案されている [28]。以下、簡単に紹介しよう。

Step 1. 与えられたサンプル点から必要な近似関数を生成する。

Step 2. 得られた近似関数に対し、満足化トレードオフ法用い、意思決定者の希求水準にある意味で最も近い Pareto 解を提示する。(ここは通常の満足化トレードオフ法と同じ手順)

Step 3. Step 2 における最適化に対し進化型計算法を用いることにより、生成した個体群から (近似) Pareto フロンティアを生成する。(ここは通常の多目的進化型計算法と同じ手順)

Step 4. 学習サンプル点を追加して Step 1 に戻る.

学習サンプル点の追加の方法は種々の可能性が考えられる. 毎回, 2つのサンプル点を追加するとき, 1つは希求水準に対応する Pareto 解の精度を上げるために得られた Pareto 解の近傍に, またあと1つは Pareto フロンティアを出来る限り広域に近似するために次の手順で与える手法が考えられている.

- (i) Step 1 で計算した目的関数の実測値 $f(\mathbf{x}_i)$, $i = 1, \dots, p$ に対し, ランキング法を用いてランク R_i , $i = 1, \dots, p$ を計算する.
- (ii) $\{(\mathbf{x}_i, R_i), i = 1, \dots, p\}$ を学習データとし RBFN や SVM などにより近似ランク関数 $\hat{R}(\mathbf{x})$ を求める.
- (iii) Step 3 で得られた近似 Pareto 最適解の中から近似ランク関数の値が一番高い点

$$\mathbf{x}^b = \arg \max_{j=1, \dots, k} \hat{R}(\mathbf{x}^j)$$

を追加する.

さらに, 設計変数の空間で既学習サンプル点が粗な所に追加することも一法である.

[例]

ベンチマーク問題としてよく知られている片持梁の問題に対し上記手法を適用する [3, 18]. この問題は設計変数が 4 個で, 4 個の非線形不等号制約, および設計変数に対する両端制約がある問題で 2 つの目的関数をもつ:

$$\text{(片持梁問題) 目的: } f_1(h, l, t, b) = 1.10471h^2l + 0.04811tb(14 + l)$$

→ min

$$f_2(h, l, t, b) = \frac{2.1952}{t^3b} \rightarrow \min$$

$$\text{制約: } g_1(h, l, t, b) = \tau \leq 13600$$

$$g_2(h, l, t, b) = \sigma \leq 30000$$

$$g_3(h, l, t, b) = h - b \leq 0$$

$$g_4(h, l, t, b) = P_c \geq 6000$$

$$0.125 \leq h, b \leq 5.0, 0.1 \leq l, t \leq 10.0$$

ここで,

$$\begin{aligned} \tau &= \sqrt{(\tau')^2 + (\tau'')^2 + \frac{l\tau'\tau''}{\sqrt{0.25(l^2 + (h+t)^2)}}} \\ \tau' &= \frac{6000}{\sqrt{2hl}} \\ \tau'' &= \frac{6000(14 + 0.5l)\sqrt{0.25(l^2 + (h+t)^2)}}{\sqrt{2hl} \left(\frac{l^2}{12} + 0.25(h+t)^2 \right)} \\ \sigma &= \frac{504000}{t^2b}, \quad P_c = 64746.022(1 - 0.0282346t)tb^3 \end{aligned}$$

である.

例として,

$$\begin{aligned} \text{理想点} &:= (f_1^*, f_2^*) = (0, 0) \\ \text{希求水準} &:= (\bar{f}_1, \bar{f}_2) = (4.0, 0.003) \end{aligned}$$

とし, 逐次近似多目的最適化法によるシミュレーションでは, 初期サンプル点をランダムで 50 個発生し, 異なる初期サンプル点で 10 回の試行を行う.

関数形が分かっているとして解析微分を用いた非線形最適化手法によって満足化トレードオフ法で解いた結果との比較は表 1 の通りである.

表 1: 満足化トレードオフ法におけるモデル予測最適化と解析微分を用いた方法の比較

	f_1	f_2	評価回数
解析微分	5.0102	3.78E-03	249.9
モデル予測	5.0598	3.76E-03	87.4

逐次近似多目的最適化法によって得られた意思決定者の希求水準に最も近い近似 Pareto 最適解は, 解析微分を用いた通常最適化による結果より少し劣っているが, それなりの精度をもつ近似最適解が求められている.

表 1 に示す評価回数は, 目的関数が既知としているため, 解析的微分情報を使っている結果である. もし目的関数の形が設計変数に関し陽に分からないとすると, 数値微分を行わなければならないので, 全体の評価回数は (目的関数の数 × 設計変数の数 × 評価回数) の約 1600~2000 くらいである. 一方, モデル予測最適化では平均 87.4 回と極めて少ない評価回数でほとんど同じ精度の解が得られている.

Pareto フロントニアの近似生成に対しては, 逐次近似多目的最適化法によって生成された Pareto フロントニアと NSGAII による結果 (100 個体で 50 世代まで得られた Pareto 個体群) を図 5 に示す. NSGA では 5000 回の評価回数を要しているのに対し, モデル予測多目的最適化では 70 回の評価回数でほぼ同じ程度の近似精度の Pareto フロントニアが得られている.

上記の結果から、逐次近似多目的最適化法は従来の手法に比べ評価回数を $1/100 \sim 1/10$ くらいまで減らすことができると考えられる。また、希求水準を取り入れた逐次近似多目的最適化法は、多目的 GA のように単に Pareto フロンティアを生成するだけでなく、意思決定者の選好を反映した Pareto 最適解の導出とともに Pareto フロンティアの全体もしくはその一部を生成することで、多目的最適化法の本質的な課題である最終解の選択やトレードオフ分析を容易にしている。

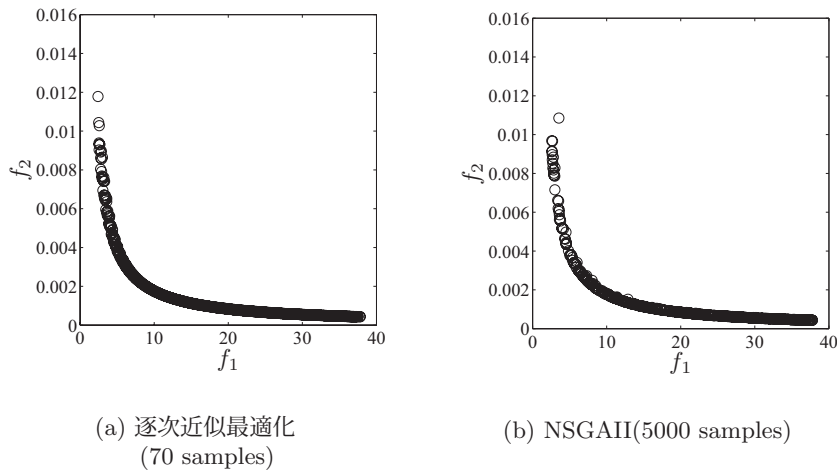


図 5: 近似 Pareto フロンティアの比較

7 実問題への応用

7.1 満足化トレードオフ法の斜張橋架設精度管理への応用

斜張橋はケーブル張力および橋梁のたわみ形状（キャンバー）がその架設段階における計画値に一致するようにケーブルの長さの調整（シム調整）を行っている。このような架設精度管理を主要な架設段階で行うことにより、完成時には応力状態、たわみ形状が設計時の所定の精度内で実現されるよう図っている。これらは各ケーブルごと、あるいは着目点ごとに達成されなければならない、それらの合計数は通常非常に大きくなる。しかも、調整作業は、温度変化の影響をできる限り少なくするため、温度が1日で1番安定している午前2時頃から午前8時頃までという短時間に行わなければならないという制約がある。

従来、このような問題に対しては誤差の2乗和を最小にするという方法で解くことが試みられていたが、このように多数ある目的に適切な重みを所定の時間内で定めることはほとんど不可能であった。そこで、重みを調整するのではなく、希求水準を調整することにより望ましい解を得ようとする満足化トレードオフ法 [2] を用いて、斜張橋ケーブル架設精度管理システムを構築した。

7.1.1 モデル化

この架設精度管理において考慮すべき量には次のようなものがある。

- i. 各ケーブル張力の残留誤差,
- ii. 各ノードにおけるキャンバー残留誤差,
- iii. 各ケーブルに対するシム調整量,
- iv. 調整すべきケーブル本数.



図 6: 斜張橋

ここで、シム調整によるケーブル剛性の変化を無視し、構造に線形性を仮定すると各ケーブル張力の残留誤差はつぎのように与えられる: n をケーブル本数とし、 ΔT_i ($i = 1, \dots, n$) をケーブル張力の計画値と実測値との差とする. 第 k ケーブルに単位長さのシム調整を行ったときの第 i ケーブルの張力変化を X_{ik} とすると、各ケーブルに Δl_k ($k = 1, \dots, n$) だけシム調整を行った場合にケーブルに残留する張力誤差は

$$p_i = \left| \Delta T_i - \sum_{k=1}^n X_{ik} \Delta l_k \right| \quad (i = 1, \dots, n)$$

で与えられる. また、キャンバーについて m をその着目点の数、 Δz_j ($j = 1, \dots, m$) をその計測値と実測値との差とする. 第 k ケーブルに単位長さのシム調整を行ったときの第 j 着目点のキャンバーの変化を ΔY_{jk} とすると、各ケーブルに $\Delta l_1, \dots, \Delta l_n$ だけシム調整を行った場合のキャンバーの残留誤差は

$$q_j = \left| \Delta Z_j - \sum_{k=1}^n y_{jk} \Delta l_k \right| \quad (j = 1, \dots, m)$$

で与えられる. さらに、シム調整量については

$$r_i = |\Delta l_i| \quad (i = 1, \dots, n)$$

なる目的関数として取り扱くと同時に、ケーブル定着部の構造などから決まるシム調整量の上・下限制約

$$\Delta l_{Li} \leq \Delta l_i \leq \Delta l_{Ui} \quad (i = 1, \dots, n)$$

を考慮しなければならない.

7.1.2 処理手順

測定された誤差をモニター画面に橋の形状とともにグラフとして出力する(図7)。初期希求水準はこのモニターに映し出されたグラフィクス上でマウスあるいはキーボードから入力する。メニューから計算を選び、補助最適化を行う。得られた結果が再びモニター画面上に映し出される。補助最適化によって提示された Pareto 解(ケーブル調整量, 応力誤差, キャンバー誤差)をグラフで見て, その状態に満足できなければ新しい希求水準を再びマウスあるいはキーボードから入力する。

このような入力は橋梁技術者が行うが, 試行錯誤の回数を減らすために, 現在の解よりも厳しくしたいと思う目的のみを入力し, 他は自動トレードオフ法を用いて自動的に算出するモード, あるいは適正トレードオフ法を用いて希求水準を連続的に変化させることにより, 結果をアニメーションにすることもできる [3], [4]。このような入出力に対する工夫によって操作は軽便になり, 早く満足できる解が得られることにもつながる。また, ケーブルの調整はその量よりもむしろ調整すべきケーブル数を減らすことの方がコストの点から重要である。ケーブル数そのものをそのまま変数にすると整数計画問題となって解くのに時間がかかるため, 本システムでは, モニター画面を見て橋梁技術者がこのような状態であるならこのケーブルは調整しなくてよいと判断した場合, マウスでそのケーブルをクリックするだけでそのケーブルは固定されるようになっている。すべてを数理計画として定式化するよりも, このように人間の手動をうまく利用する方が全体として効率が向上することも多い。

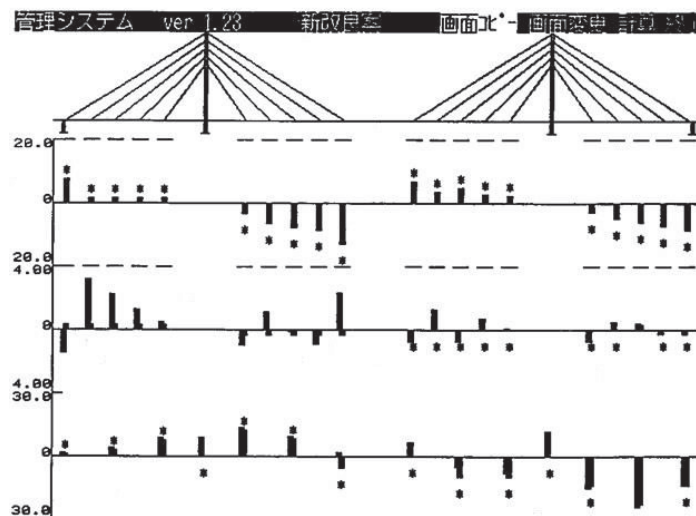


図 7: 斜張橋架設精度管理システム

7.2 モデル予測最適化の斜張橋耐震補強への応用

平成7年兵庫県南部地震以降, 既設構造物の耐震性の見直しが進められている。既設の大型構造物に対し, 各部に耐震要素(ばね, ダンパ, エネルギー吸収部材など)を設置して地震応答を低減する方法が検討されてきている。ここでは斜張橋に対する耐震補強に対する応用例を紹介しよう。

斜張橋主塔の大地震対策として、主塔の橋軸直角水平方向応答低減に対しケーブルにマスを付加することが有効であることが示されており、各ケーブルへの付加マス決定を最適化することにする。最適化は、1) 主塔基部の曲げモーメントを最小にするケースと、2) 塔基部曲げモーメントと同時に質量を付加するケーブル数をできるだけ少なくするケースなどを考える。設計変数は20個である。本問題における大きな困難は与えられた付加マス量に対する各種応答が振動解析によって初めて得られ、実際の非線形振動解析には通常1回あたり数十分かかるということである。そこで、この問題に対しRBFネットワークにより逐次近似最適化を行い解析回数の低減を図った。

表 2: ケース1に対する結果

	従来法	RBFN	
		最良値	平均
曲げモーメント	50.3	54.90	63.70
解析回数	1365	150.00	124.80

表 3: ケース2に対する結果

	従来法		RBFN	
	最良値	平均	最良値	平均
曲げモーメント	62.6	62.8	67.1	69.7
付加ケーブル数	2	2	6	6.2
目的関数	0.526	0.527	0.756	0.784
解析回数	4100	3780	199	193.3

表2, 3より従来法と比べて1/10~1/20程度の解析回数で実用的に満足できる精度の解が得られていることが分かる。

実際の問題においては種々の状況が考えられるためケース2の結果の評価は容易ではないが、わずか数本のケーブルに付加質量を与えることにより塔基部曲げモーメントを無対策時の半分以下にできるという結果は、初期学習データ作成を含めて従来法の1/20程度の構造解析回数で得られるものとしては悪いものではないと考えられる。構造解析の回数が1/20程度になれば、従来不可能であった大規模非線形構造物の最適化も実用の域に達するものが少なくない。これにより現状では手つかずの状態にある既設大型構造物の実際的な耐震対策の検討が可能になると期待されている。

8 おわりに

多目的最適化の研究においては、伝統的な手法である対話型解法と比較的最近の進化型解法とはこれまで独立して研究が進められてきた。これらの異なる2つの分野の研究者が2004年ドイツのDagstuhlに会し、これら2つのアプローチの融合、共栄について議論された。2006年、2009年の

Dagstuhl 会議を経て、これまで独立に発展してきた伝統的手法と新しい進化的計算による手法とが融合されようとしており、実社会における様々な問題への応用が活発化してきている。さらに実用化という観点からは、シミュレーションや実験の回数を減らすことが非常に重要な課題で、モデル予測最適化は工学問題等においてその有用性がこれからますます期待される。

謝辞

本論文は著者のこれまでの研究の中で特に応用の可能性の大きいものを中心にまとめ解説したものである。これらは、これまで共同研究してきた多数の人たちに拠るものが多く、個人名を全て上げることは紙幅の都合上できないが、ここに感謝の意を表したい。また、科学研究費補助金をはじめとする各種研究助成金・奨学寄付金によって研究環境の側面から多大の援助を頂いたことに篤く御礼を申し上げる。

参考文献

- [1] C. M. Bishop, *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.
- [2] C. A. C. Coello, D. A. Van Veldhuizen, and G. B. Lamont, *Evolutionary Algorithms for Solving Multi-Objective Problems*. Kluwer Academic Publishers, 2001.
- [3] K. Deb, *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley & Sons, 2001.
- [4] S. S. Erenguc and G. J. Koehler, "Survey of Mathematical Programming Models and Experimental Results for Linear Discriminant Analysis," *Managerial and Decision Economics*, vol. 11, pp. 215-225, 1990.
- [5] N. Freed and F. Glover, "Simple but Powerful Goal Programming Models for Discriminant Problems," *European Journal of Operational Research*, vol. 7, pp. 44-60, 1981.
- [6] D. R. Jones, M. Schonlau, and W. J. Welch, "Efficient Global Optimization of Expensive Black-Box Functions," *Journal of Global Optimization*, vol. 13, pp. 455-492, 1998.
- [7] S. Haykin, *Neural Networks: A Comprehensive Foundation*. Macmillan College Publishing Company, 1994.
- [8] 中山, "多目的計画法に対する満足化トレードオフ法の提案," 計測自動制御学会論文集, vol. 20, pp. 29-35, 1984.
- [9] H. Nakayama, "Trade-off Analysis using Parametric Optimization Techniques," *European Journal of Operational Research*, vol. 60, pp. 87-98, 1992.
- [10] 中山, 谷野, 多目的計画法の理論と応用. 計測自動制御学会編, コロナ社, 1994.

- [11] H. Nakayama, K. Kaneshige, S. Takemoto, and Y. Watada, "An Application of a Multi-objective Programming Technique to Construction Accuracy Control of Cable-stayed Bridges," *European Journal of Operational Research*, vol. 87, pp. 731-38, 1995.
- [12] 中山, 荒川, 佐々木, "RBF ネットワークと遺伝アルゴリズムによる未知目的関数の最適化," システム制御情報学会論文誌, vol. 13, pp. 152-154, 2000.
- [13] H. Nakayama, M. Arakawa, and R. Sasaki, "Simulation-based Optimization using Computational Intelligence," *Optimization and Engineering*, vol. 3, pp. 201-214, 2002.
- [14] H. Nakayama, M. Arakawa, and K. Washino, "Optimization for Black-box Objective Functions, Optimization and Optimal Control," (eds.) P. M. Pardalos, I. Tseveendorj and R. Enkhbat, World Scientific, pp. 185-210, 2003.
- [15] H. Nakayama and Y. Yun, "Generating Support Vector Machines using Multiobjective Optimization and Goal Programming Multi-objective Machine Learning," (ed.) Yaochu Jin, Springer Series on *Studies in Computational Intelligence*, pp.173-198, 2006.
- [16] H. Nakayama, and Y. Yun, "Support Vector Regression Based on Goal Programming and Multiobjective Programming," in *Proc. IEEE World Congress on Computational Intelligence*, CD-ROM (Paper ID: 1536), 6 pages, 2006.
- [17] H. Nakayama, K. Inoue, and Y. Yoshimori, "Approximate Optimization Using Computational Intelligence and its Application to Reinforcement of Cable-stayed Bridges," (eds.) Xuan F. Zha and R. J. Howlett, *Integrated Intelligent Systems for Engineering Design*, IOS Press, pp. 289-304, 2006.
- [18] 中山, 岡部, 荒川, 尹, 多目的最適化と工学設計-しなやかシステム工学アプローチ-. 現代図書, 2007.
- [19] Y. Sawaragi, H. Nakayama, and T. Tanino. *Theory of Multiobjective Optimization*. Academic Press, 1985.
- [20] B. Schölkopf, and A. J. Smola, "New Support Vector Algorithms," *NeuroCOLT2 Technical report Series*, NC2-TR-1998-031, 1988.
- [21] B. Schölkopf, and A. J. Smola, "Learning with Kernels: Support Vector Machines, Regularization," *Optimization, and Beyond*, MIT Press, 2002.
- [22] M. Schonlau, *Computer Experiments and Global Optimization*. Ph.D. Thesis, University of Waterloo, 1997.
- [23] 田村, 藤田, 中村, 効用分析の数理と応用. 計測自動制御学会編, コロナ社, 1997.
- [24] V. N. Vapnik, *Statistical Learning Theory*. John Wiley & Sons, New York, 1998.
- [25] 渡邊, 萩原, 赤穂, 本村, 福永, 岡田, 青柳, 学習システムの理論と実現. 森北出版, 2005.

- [26] Y. B. Yun, H. Nakayama, and M. Arakawa, "Multiple criteria decision making with generalized DEA and an aspiration level method," *European Journal of Operational Research*, vol. 158, pp. 697-706, 2004.
- [27] Y. B. Yun, H. Nakayama, and M. Arakawa, "Generation of Pareto Frontiers using Support Vector Machine," in *Proc. 17th International Conference on Multiple Criteria Decision Making*, MCDM, CD-ROM, 9 pages, 2004.
- [28] 尹, 中山, 尹, "計算知能を用いた逐次近似多目的最適化手法," 計測自動制御学会論文集, vol. 43, pp. 672-678, 2007.