数理解析研究所講究録
第 1814 巻 2012 年 142-147

数値数式手法による多目的最適化

岩根秀直
(株) 富士通研究所*
HIDENAO IWANE
FUJITSU LABORATORIES LTD

穴井宏和
(株) 富士通研究所/九州大学†
HIROKAZU ANAI
FUJITSU LABORATORIES LTD/KYUSHU UNIVERSITY

屋並仁史
(株) 富士通研究所‡
HITOSHI YANAMI
FUJITSU LABORATORIES LTD

1 はじめに

制約条件下で目的関数を最小，もしくは最大にするような問題は**最適化問題**と呼ばれる．そのうち目的関数が複数のものを多目的最適化問題と呼ぶ．実問題では単目的の問題のほうが稀で多くの場合多目的最適化問題になる．

多目的最適化問題における意思決定方法としてさまざまな数値手法を用いた方法が研究されている．それらは高速に最適に近い解を求めるが，正しい解が得られているか確認する方法がないという問題がある．一方で，数値手法では解決が困難であるパラメータ付きの最適化問題や非凸最適化問題の強力なツールとして限量記号消去（quantifier elimination: QE）アルゴリズムを用いる試み [9] が数多く現れている．しかし，QE は計算量が大きいために工学・産業上で求められる問題の規模・計算時間の点ではまだ実用的ではない．

本稿では QE のみを適用した場合には解けない多目的最適化問題にたいして数値手法を組み合わせることで高速に精度良く求める方法を提案する．

2 多目的最適化問題

ものづくりの現場ではさまざまな設計がおこなわれている．設計とは，ある制約条件の下で所望の性能・性質を満足するような設計対象を規定することといえ，それらを最適化（最小化／最大化）するように設計パラメータを求めることを設計最適化問題と呼ぶ．

最適化問題は以下のように定式化される．

$$\begin{array}{ll} \text{minimize} & y_1 = f_1(x),\, y_2 = f_2(x),\, \ldots,\, y_n = f_n(x) \\ \text{subject to} & x \in C(x) = \{x \in \mathbb{R}^k \mid g_1(x) \le 0,\, g_2(x) \le 0,\, \ldots,\, g_m(x) \le 0\} \end{array}$$

*iwane@jp.fujitsu.com
†anai@jp.fujitsu.com
‡yanami@labs.fujitsu.com

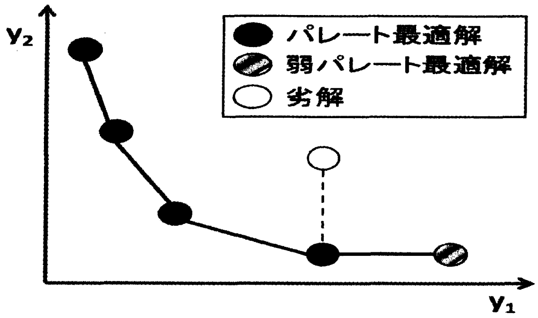

図 1: パレート最適解

ここで $f_i(x)$ は**目的関数**，$C(x)$ は**制約条件**とよばれる．複数の目的関数をもつ最適化問題を**多目的最適化問題**といい，目的関数が 1 つの場合には**単目的最適化問題**という．

単目的最適問題であれば，実数の大小関係が全順序であるから最適解の概念は明瞭であるが，多目的最適化の場合には目的関数の間に一方が良くなれば他方が悪くなる**トレードオフ関係**があるため最適解は一つではない．そこで多目的最適化問題では次に示すパレート最適解の集合を最適解として扱う．

定義 1
任意の $x' \in C(x)$ に対して以下の条件を満たすものがないとき，実行可能解 $x \in C(x)$ は**パレート最適解**であるという．

$$f_i(x') \le f_i(x) \qquad (\forall i = 1, \ldots, n)$$

$$f_j(x') < f_j(x) \qquad (\exists j \in \{1, \ldots, n\})$$

多目的最適化に対しては様々な手法が知られているが大きく 2 つに分類される．1 つは問題を単目的最適問題に落として最適解を求める方法で，他方はパレート最適解そのものを求める方法である．

前者の古典的な方法は，目的関数に対する重み w_i を用いて単目的最適化問題に変換する．

$$\begin{array}{ll} \text{minimize} & y = \sum_{i=1}^{n} w_i f_i(x) \\ \text{subject to} & x \in C(x) = \{g_1(x) \le 0,\ g_2(x) \le 0,\ \ldots,\ g_m(x) \le 0\} \end{array}$$

この方法では多くの場合パレート最適解の 1 つを得ることができるが，非凸問題ではパレート解を求められないことがある．さらに各 f_i が所望の性能・性質を得られなかった場合には重みを変えて最適化をやりなおす必要がある．後者は，パレート最適解を求めるのでやり直しの必要はないが十分な精度の解を得るには計算量が大きくなるという問題がある．

3 Quantifier Elimination

ここでは多目的最適化問題に対する数式処理アプローチである QE について説明する．

限量記号消去アルゴリズム (quantifier elimination algorithm) [8, 1, 6] とは，与えられた形式理論 (formal theory) について「限量記号付きの式（一階述語論理式）」を入力とし「等価で限量記号無しの式」を出力するアルゴリズムのことである．

例えば QE は以下のような QE 問題を限量記号なしの等価な式に変換する．

$$\exists x\ (x^2 + bx + c = 0) \quad \Rightarrow \quad b^2 - 4c \ge 0$$

$$\exists x_1, \exists x_2\ (y_1^2 = 4x_1\ \wedge\ y_2 = x_1 - x_1x_2 + 5\ \wedge\ 1 \le x_1 \le 4\ \wedge\ 1 \le x_2 \le 2)$$
$$\Rightarrow\ (y_1 \ge 2\ \vee\ y_1 \le -2)\ \wedge\ -4 \le y_1 \le 4\ \wedge\ y_2 \le 5\ \wedge\ 4y_2 + y_1^2 - 20 \ge 0)$$

QE は一階述語論理式によって表現される広範囲の重要な応用問題（最適化問題）に適用できる強力なツールである．また等価な変換を行うため数値手法が扱いが困難な非凸問題も正確に解くことができる．ところが，QE の計算量の下限が指数的であること示されている [3]．計算量の壁を乗り越えるために従来の数値的な手法で使われているアイデアを QE の代数的な算法に取り入れて高速に実現する研究 [2, 7, 5] も行われている．

4　数値手法と数式処理による多目的最適化

ここでは例を用いて数値手法と数式処理手法の解法の違いを説明する．

典型的な数値手法のアルゴリズムは以下のように記述できる．代表的な手法である遺伝的アルゴリズム (Genetic Algorithm: GA) [4] 等は x をうまく選択することで少ない繰り返し回数で局所解に陥らずに最適解を得るように工夫している．

アルゴリズム 1

<u>*num_opt*$(\{f_1(x), \ldots, f_n(x)\}, C(x))$</u>
入力: $f_i(x)$: 目的関数
　　$C(x)$: 制約条件
出力: パレート最適解集合
　Loop
　　$x \leftarrow$ 任意に選択
　　if $C(x)$ *then*
　　　$y_i \leftarrow f_i(x)$
　　　plot $(y_1, \ldots, y_n)$

　return パレート最適解集合

一方，QE を用いて多目的最適化問題を解く場合には最初に多目的最適化問題を以下のような QE 問題に置き換える．

$$\exists x\ (y_1 = f_1(x)\ \wedge\ y_2 = f_2(x)\ \wedge \cdots \wedge\ y_n = f_n(x)\ \wedge\ C(x))$$

QE で x を消去することで y_i の可能領域を得ることができ，パレート最適解が容易に得られる．

図 2，図 3 は以下の問題をそれぞれ数値手法，数式処理手法を用いて解いた結果を図示したものである．

$$\begin{array}{ll} \text{minimize} & y_1 = 2\sqrt{x_1},\ y_2 = x_1 - x_1x_2 + 5 \\ \text{subject to} & 1 \le x_1 \le 4,\ 1 \le x_2 \le 2 \end{array}$$

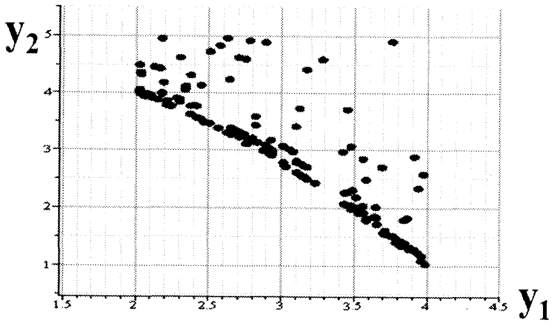

図 2: 数値手法（GA）による最適化

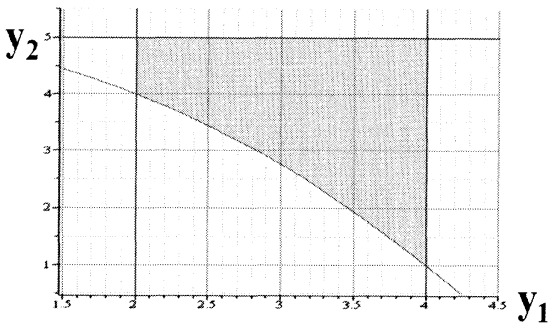

図 3: 数式処理手法（QE）による最適化

数値手法は高速で少ない繰り返しで最適解が得られることが多いが非凸問題は取扱いが困難である．また x の探索範囲が広い場合には十分な解の精度を得るために繰り返し回数が多くなる傾向にある．また，図 2 のような結果からパレート最適解を得ているか確かめる方法がないためいつアルゴリズムを停止していいかわからないという問題がある．それに対し，数式処理手法では図 3 のように正確に可能領域を得ることができるが，計算量が大きいために計算が停止しないことがある．

本稿では単に QE を用いただけでは解けないような大きな問題に対して，問題の特徴をいかして数値手法を組み合わせることで高速に精度良い解を得る方法を提案する．

5 数値数式手法による多目的最適化

例 1

数値数式手法を適用した具体的な例として SRAM セル（図 4，図 5）の設計最適化問題を以下に示す．

$$\begin{array}{ll} \text{minimize} & y_1 = x_9,\ \ y_2 = x_7 x_8 \\ \text{subject to} & 0 \le x_i \le 1\ (i = 1, \ldots, 6), \\ & x_7 = 3 + x2 + x4,\ x_8 = 5 + 2x_3 + 2x_5, \\ & x_9 = a_0 + a_1 x_1 + \cdots + a_6 x_6 \end{array}$$

ここで a_i は定数で，y_1 は不良率，y_2 はセルサイズ（面積）を表している．x_i を最適化によりうまく選ぶことでトレードオフ関係をもつ不良率とセルサイズが小さくなるような設計をしたい．

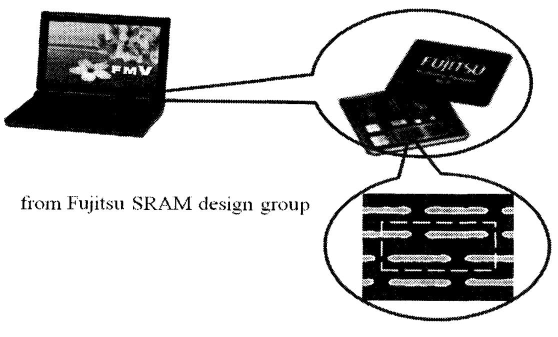

図 4: SRAM

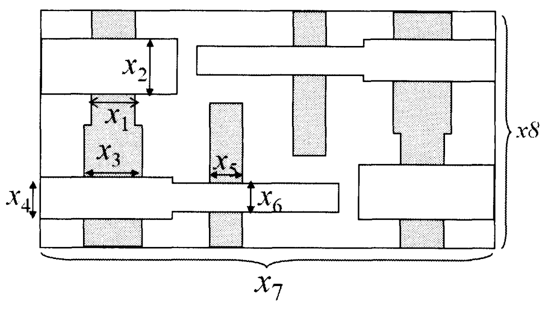

図 5: SRAM セルモデル

この例は計算量が大きいため QE だけでは解くことができなかった．具体的には，y_2 の部分に 2 次の項が含まれていることが計算量の増加をもたらす原因であった．

この問題では目的関数の部分に含まれるのは一部の変数で，多くの変数は制約条件にのみ含まれている．この性質を利用して問題を以下のように変形する．

$$
\begin{aligned}
& \exists x\ (y = f(x)\ \wedge\ C(x)) \\
\Leftrightarrow\quad & \exists \bar{x}\ \exists \underline{\mathrm{x}}\ (y = f(\bar{x})\ \wedge\ C(\bar{x}, \underline{\mathrm{x}})) \\
\Leftrightarrow\quad & \exists \bar{x}\ (y = f(\bar{x})\ \wedge\ \exists \underline{\mathrm{x}}\ (C(\bar{x}, \underline{\mathrm{x}})))
\end{aligned}
$$

上記のように分解できる問題に対する数値数式手法による多目的最適化アルゴリズムを以下に示す．

アルゴリズム 2

$\underline{syn\_opt(\{f_1(\bar{x}), \ldots, f_n(\bar{x})\}, C(\bar{x}, \underline{\mathrm{x}}))}$

入力: $f_i(\bar{x})$: 目的関数

$C(\bar{x}, \underline{\mathrm{x}})$: 制約条件

出力: パレート最適解集合

$G(\bar{x}) \leftarrow QE(\exists \underline{\mathrm{x}}\ (C(\bar{x}, \underline{\mathrm{x}})))$

Loop

$\bar{x} \leftarrow$ 任意に選択

if $G(\bar{x})$ *then*

$y_i \leftarrow f_i(\bar{x})$

plot $(y_1, \ldots, y_n)$

return パレート最適解集合

SRAM の例では $C(\bar{x}, \underline{\mathrm{x}})$ に 2 次の項が含まれていないため $G(\bar{x})$ は数秒で求めることができた．また $\bar{x}$ は 3 変数であり数値手法を用いる場合の探索空間は小さくなるので少ない繰り返し回数で精度の良い解を得ることが可能となった．

図 6 は SRAM の設計最適化問題に対して，数値手法（GA）のみを適用した場合で，図 7 は提案手法である QE で変数を減らした等価な式に対して数値手法を適用した場合の結果を図示したものである．同じ数値手法を適用した場合に提案手法のほうが精度良い解を得ていることを確認できる．

6　おわりに

QE のみを適用して解けないようなクラスの問題について，数値手法を組み合わせることで高速に精度良く多目的最適化問題をとけることを示した．また実際に設計問題に適用しその効果を確認することができた．

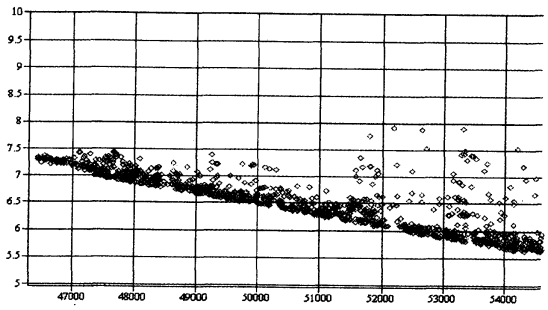

図 6: 数値手法のみ

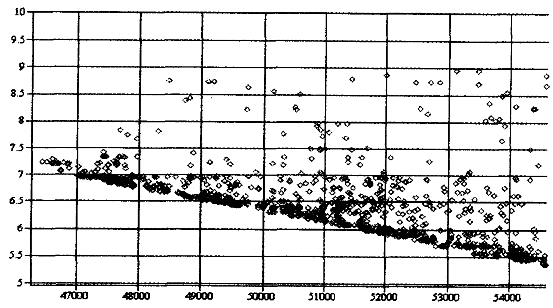

図 7: 提案手法（数値数式）

参 考 文 献

[1] George E. Collins. Quantifier elimination for real closed fields by cylindrical algebraic decomposition. In *LNCS*, volume 32. Springer Verla, 1975.

[2] George E. Collins, Jeremy R. Johnson, and Werner Krandick. Interval arithmetic in cylindrical algebraic decomposition. *J. Symb. Comput.*, 34(2):145–157, 2002.

[3] James H. Davenport and Joos Heintz. Real quantifier elimination is doubly exponential. *J. Symb. Comput.*, 5(1/2):29–35, 1988.

[4] David E. Goldberg. *Genetic Algorithms in Search Optimization and Machine Learning.* Addison-Wesley, 1989.

[5] H. Iwane, H. Yanami, H. Anai, and K. Yokoyama. An effective implementation of a symbolic-numeric cylindrical algebraic decomposition for quantifier elimination. In *Proceedings of the 2009 International Workshop on Symbolic-Numeric Computation*, volume 1, pages 55–64, 2009.

[6] Rüdiger Loos and Volker Weispfenning. Applying linear quantifier elimination. *Comput. J.*, 36(5):450–462, 1993.

[7] Adam W. Strzebonski. Cylindrical algebraic decomposition using validated numerics. *J. Symb. Comput.*, 41(9):1021–1038, 2006.

[8] A. Tarski. *A decision method for elementary algebra and geometry.* University of California Press, 2d edition, 1952.

[9] H. Yanami. Multi-objective design based on symbolic computation and its application to hard disk slider design. *Journal of Math-for-Industry*, 1:149–156, 2009.