

パラメトリック多項式最適化問題専用 Cylindrical Algebraic Decomposition と 動的計画法への適用

岩根秀直
(株) 富士通研究所*

HIDENAO IWANE
FUJITSU LABORATORIES LTD

吉良知文
九州大学†

AKIFUMI KIRA
KYUSHU UNIVERSITY

穴井宏和
(株) 富士通研究所/九州大学‡

HIROKAZU ANAI
FUJITSU LABORATORIES LTD/KYUSHU UNIVERSITY

1 はじめに

限量記号消去アルゴリズム (quantifier elimination (QE) algorithm) とは, 与えられた形式的理論 (formal theory) について「限定記号付きの式 (一階述語論理式)」を入力とし「等価で限定記号無しの式」を出力するアルゴリズムのことである. **Cylindrical Algebraic Decomposition** (CAD) [4] は, 与えられた多項式集合に対して, 変数空間を多項式の符号が不変である領域に分割する手法で, CAD を用いた QE の解法が提案されている.

QE は工学や産業上の問題などの多くの応用があり重要なアルゴリズムである. QE を用いると, 多項式最適化問題を正確に解くことができる. QE は最悪の場合の計算量が限量記号がついた変数の数に対し二重指数であることが示されており, 本質的に大きな問題を解くことができない.

本稿では, 多項式最適化問題において, 効率的に最適値関数を求めるパラメトリック最適化問題専用 CAD アルゴリズムについて述べる. また, 専用 CAD の適用事例として, 確定的な多項式動的計画問題への適用を紹介する.

2 パラメトリック多項式最適化問題と数式処理による解法

2.1 パラメトリック多項式最適化問題

最適化問題 (optimization problem) とは, 与えられた**制約条件** (constraints) の元で, **目的関数** (objective function) $f(x)$ が最小 (または最大) になるような**決定変数** (decision variables) $x = (x_1, \dots, x_n) \in \mathbb{R}^n$ の

*iwane@jp.fujitsu.com

†a-kira@math.kyushu-u.ac.jp

‡anai@jp.fujitsu.com

値を見つける問題である。一般に最適化問題は以下のように定式化される。

$$\begin{aligned} & \text{Minimize} && f(x) \\ & \text{subject to} && \varphi(x) \end{aligned} \quad (1)$$

制約条件 φ により表される集合を**実行可能領域** (feasible region), 実行可能領域の目的関数 f による像を**目的関数の実行可能領域** (feasible objective region), 目的関数 f が最小となるとき x を**最適解** (optimal solution), そのときの目的関数値を**最適値** (optimal value) という。

パラメータ $\theta = (\theta_1, \dots, \theta_m)$ を含む最適化問題はパラメトリック最適化問題と呼ばれ, 以下のように定式化される。

$$\begin{aligned} & \text{Minimize} && f(x, \theta) \\ & \text{subject to} && \varphi(x, \theta) \end{aligned} \quad (2)$$

パラメトリック最適化問題の場合, 最適値はパラメータを用いて表現され, **最適値関数** (optimal value function) とよぶ。本稿では, 目的関数や制約条件がすべて多項式で記述される**多項式最適化問題** (polynomial optimization problem) を扱う。ここで, 制約条件 $\varphi(x, \theta)$ は半代数的集合で与えられており, 問題を簡単にするため制約条件を満たす点集合はコンパクトであると仮定する。

多項式最適化問題の記述能力は高く, 工学および産業などでさまざまな応用がある。そのため半正定値緩和を用いるアプローチなど多項式最適化問題に対する研究が活発に行われているが, 非凸な問題やパラメトリックな最適化問題を正確に解くことは困難である。しかし, 次節で紹介する限量記号消去法を用いると正確に解を得ることができる。

2.2 限量記号消去法

限量記号消去 (Quantifier Elimination: QE) とは実数を動く変数に対して $\exists, \forall$ のような限量記号のついた一階述語論理式 (first-order formula) (多項式の等式, 不等式とそれらを $\wedge, \vee$ 等で結合した論理式) からそれと等価で限量記号のない論理式を計算するアルゴリズムのことである。例えば $\exists x (x^2 + bx + c = 0)$ に対して, QE は x がいない等価な論理式 $b^2 - 4c \geq 0$ を返す。この入力 $x^2 + bx + c$ が x 軸と交わる条件を求める問題で, その判別式が非負であることに等価である。

QE は, 豊かな記述能力を持つ一階述語論理式によって表現される広い範囲の重要な応用問題を統一的にシステムチックに取り扱うことが可能であるため, 計算機科学や各種の理工学分野の研究者が QE を活用するようになってきている。

2.3 QE によるパラメトリック多項式最適化手法

本節では, パラメトリック多項式最適化問題を QE を用いて正確に解く手法について述べる。

多項式最適化問題 (2) を QE を用いて解くために, 新しい変数 y を導入して, 以下の一階述語論理式を考える。

$$\exists x (y \equiv f(x, \theta) \wedge \varphi(x, \theta)) \quad (3)$$

この一階述語論理式は, y が目的関数の実行可能領域に属する場合に真となる論理式である。したがって, この論理式から決定変数 x を消去するとパラメータ θ を用いた目的関数の実行可能領域を表す論理式 $\psi_{\text{Feasible}}(\theta, y)$ を得ることができる。

ψ_{Feasible} を用いて, 最適値関数は次のように定式化できる。

$$\psi_{\text{Feasible}}(\theta, y) \wedge \neg \exists z (\psi_{\text{Feasible}}(\theta, z) \wedge z < y) \quad (4)$$

ここで、 $\neg$ は否定を表す論理記号である。 $\neg$ 以下の論理式で、 y よりも小さい目的関数値が存在しないことを表現している。この論理式から、変数 z を消去することで最適値関数を表す論理式 $\psi_{Opt}(\theta, y)$ を得ることができる。

最後に、最適解を求める。最適解は、その f による像が最適値となるような実行可能解なので、以下のよう

$$\exists y (y \equiv f(x, \theta) \wedge \varphi(x, \theta) \wedge \psi_{Opt}(\theta, y))$$

この論理式から変数 y を消去すると、最適解を表す論理式を得ることができる。

例 1

以下の多項式最適化問題を考える。

$$\begin{aligned} & \text{Maximize} && \sum_{i=1}^4 x_i \\ & \text{subject to} && \sum_{i=1}^4 x_i^2 \leq 1, \\ & && x_i \geq 0 \ (i = 1, \dots, 4) \end{aligned}$$

目的関数の実行可能領域は新しい変数 y を用いて以下のように定式化される。

$$\exists x_1 \exists x_2 \exists x_3 \exists x_4 \ (y = \sum_{i=1}^4 x_i \wedge \sum_{i=1}^4 x_i^2 \leq 1 \wedge x_1 \geq 0 \wedge x_2 \geq 0 \wedge x_3 \geq 0 \wedge x_4 \geq 0)$$

QE を用いて決定変数 x_i ($i = 1, \dots, 4$) を消去すると、以下の論理式を得る。

$$0 \leq y \leq 2$$

この結果から目的関数の最大値は 2 となることがわかる。

例 2

以下のパラメトリック最適化問題を考える。

Problem P(θ), $\theta \geq 0$:

$$\begin{aligned} & \text{Minimize} && -x_1 - \theta \\ & \text{subject to} && x_1 \geq 0 \wedge x_1^2 + \theta^2 \leq 1 \end{aligned}$$

目的関数の実行可能領域は以下のように定式化される。

$$\exists x_1 (y = -x_1 - \theta \wedge x_1 \geq 0 \wedge \theta \geq 0 \wedge x_1^2 + \theta^2 \leq 1)$$

QE を用いて決定変数 x_1 を消去すると、以下のような論理式を得る。

$$\psi_{Feasible}(\theta, y) \equiv y^2 + 2\theta y + 2\theta^2 \leq 1 \wedge y \leq \theta \wedge 0 \leq \theta \leq 1$$

図 1 は $\psi_{Feasible}$ を描画したものである。目的関数の最小値のパラメータ表現は以下のように定式化できる。

$$\begin{aligned} & \psi_{Feasible}(\theta, y) \wedge \neg \exists z (\psi_{Feasible}(\theta, z) \wedge z < y) \\ &= \forall z (\psi_{Feasible}(\theta, y) \wedge (\psi_{Feasible}(\theta, z) \rightarrow (z \geq y))) \\ &= \forall z (y^2 + 2\theta y + 2\theta^2 \leq 1 \wedge y \leq \theta \wedge 0 \leq \theta \leq 1 \wedge ((z^2 + 2\theta z + 2\theta^2 \leq 1 \wedge z \leq \theta) \rightarrow (z \geq y))) \end{aligned}$$

変数 z を消去すると以下の論理式が得られる。

$$\psi_{Opt}(\theta, y) \equiv (y^2 + 2\theta y + 2\theta^2 = 1 \wedge y \leq \theta \wedge 0 \leq \theta \leq 1)$$

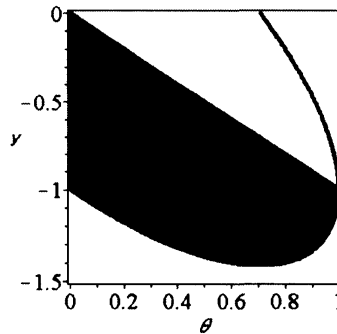


図 1: 例 2 の目的関数の実行可能領域

最小値を与える最適解のパラメータ表現は以下のように定式化される.

$$\exists y (y = -x_1 - \theta \wedge x_1 \geq 0 \wedge \theta \geq 0 \wedge x_1^2 + \theta^2 \leq 1 \wedge \psi_{Opt}(\theta, y))$$

QE を適用して変数 y を消去すると, 最適解を表す論理式

$$x^2 + \theta^2 = 1 \wedge x \geq 0$$

を得ることができる.

QE を用いると, 上記のようにパラメトリック多項式最適化問題のすべての大域的最適解を誤差なく正確に求めることができる. QE は式の等価変換を行うだけで, その入力に次数や制約式の数などの制限はなく, 数値手法が得意でない非凸な最適化問題でも正確に解くことができることが特徴である. しかし, 最悪の場合の計算量が限量記号がついた変数の数に対し二重指数 [7] であるため, 本質的に規模が大きい問題が解くことができない. そのため, 数値手法と組み合わせた方法 [1, 6, 11, 12, 16] や特別な入力に対する専用アルゴリズム [13, 17, 18] などの研究が行われている.

制約条件を満たす点集合はコンパクトであると仮定したが, コンパクトでない場合にも QE によって正確に解くことは可能である. ただし, 最小値が存在しない場合, (4) に QE を適用すると偽 (false) を復帰するため, 注意が必要である. 最小値を持たない場合にも極小値を定式化することは可能であるが本稿では説明は省略する.

2.4 Cylindrical Algebraic Decomposition

Cylindrical Algebraic Decomposition (CAD) [4] は 1975 年に, G. E. Collins により提案された代数的手法で, 与えられた多項式集合に対して変数空間を各多項式の符号が不変であるセル (cell) とよばれる領域に分割する. CAD を用いることにより QE 問題を効率的に解くことが可能となる.

CAD アルゴリズムは射影段階 (projection phase), 底段階 (base phase), 持ち上げ段階 (lifting phase) の 3 つの段階から構成される. 射影段階では, 入力の多項式集合 $F_r \subset \mathbb{Q}[x_1, \dots, x_r]$ に射影演算を繰り返し適用して, 多項式集合列 $\{F_i\}_{i=1, \dots, r-1}$ を計算する. $F_{i+1} \subset \mathbb{Q}[x_1, \dots, x_{i+1}]$ に射影演算を適用すると, 一変数消去された $F_i \subset \mathbb{Q}[x_1, \dots, x_i]$ が得られる. $\{F_i\}_{i=1, \dots, r}$ に含まれる多項式を既約因子分解して得られる多項式集合を射影因子という. 底段階では, $\mathbb{R}(=\mathbb{R}^1)$ の分解を行う. これは射影段階で得られた 1 変数多項式の集合 F_1 の実根の分離により求める. 最後の持ち上げ段階では, $\mathbb{R}^k$ の分解 D_k と F_{k+1} を用い

て $\mathbb{R}^{k+1}$ の分解 $\mathcal{D}_{k+1}$ を構築する ($k = 1, \dots, r-1$). CAD により得られるセルでは多項式集合の符号が一定なので, 標本点 (sample point) と呼ばれる任意の一点の符号を評価すればセル全体での符号を得られる. 図 2 は CAD の 3 つの段階の流れを表している.

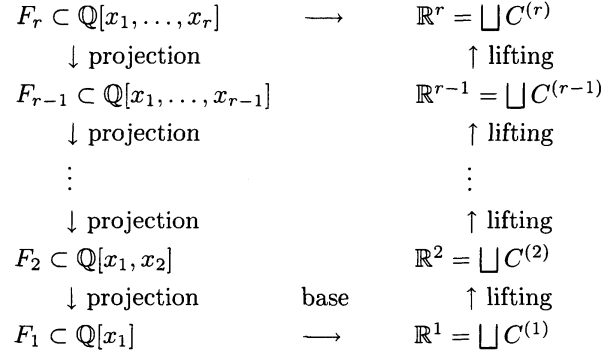


図 2: Flow of CAD

CAD を用いて QE 問題を解く場合には, 変数順序を適当な変更と, 同値な論理式への基本的な変形を行うことで以下の形式に変形する.

$$\mathcal{Q}_q x_q \mathcal{Q}_{q+1} x_{q+1} \cdots \mathcal{Q}_r x_r (\psi(x_1, \dots, x_r))$$

ここで, $\mathcal{Q}_i \in \{\exists, \forall\}$ ($i = q, q+1, \dots, r$) であり, ψ は限量記号を持たない論理式である. 次に, この変数順序で CAD を構築し, できたセルで代数的命題文 ψ が成り立つかどうかを判定する. これは各セル C の標本点を代入することで評価する. 各セルでは多項式の符号が一定であることが保証されているので, 一階述語論理式の真偽値も一定であることが保証される. この情報をもとに C が有効なセル (true cell) であるかを判断する. 最後に有効なセルをすべて集め, それらを定義する代数的命題文の論理和をとれば, それが限量記号のない等価な論理式となる.

持ち上げ段階についてもう少し詳細に述べる. セル $C \subset \mathbb{R}^k$ を持ち上げる場合には, 標本点 $s \in C$ を F_{k+1} に代入することで x_{k+1} に関する一変数多項式の集合を得る. 底段階と同様に, 実根の分離により実現される. C 上で F_{k+1} の異なる実根の個数は一定で, これらを表す代数関数が交わらないことを描画可能といい, 描画可能であることは射影演算で保証されている. $B_1, \dots, B_p$ を F_{k+1} の C 上での実根を与える代数関数とし, C 上 $B_1 < \dots < B_p$ とする. このとき, $C \times \mathbb{R}$ の中の F_{k+1} が符号不変なセルは,

$$\begin{aligned}
 C_1 &= \{(\alpha_1, \dots, \alpha_k, \alpha_{k+1}) \mid (\alpha_1, \dots, \alpha_k) \in C, \alpha_{k+1} < B_1\} \\
 C_2 &= \{(\alpha_1, \dots, \alpha_k, \alpha_{k+1}) \mid (\alpha_1, \dots, \alpha_k) \in C, \alpha_{k+1} = B_1\} \\
 C_3 &= \{(\alpha_1, \dots, \alpha_k, \alpha_{k+1}) \mid (\alpha_1, \dots, \alpha_k) \in C, B_1 < \alpha_{k+1} < B_2\} \\
 C_4 &= \{(\alpha_1, \dots, \alpha_k, \alpha_{k+1}) \mid (\alpha_1, \dots, \alpha_k) \in C, \alpha_{k+1} = B_2\} \\
 C_5 &= \{(\alpha_1, \dots, \alpha_k, \alpha_{k+1}) \mid (\alpha_1, \dots, \alpha_k) \in C, B_2 < \alpha_{k+1} < B_3\} \\
 &\vdots \\
 C_{2p-1} &= \{(\alpha_1, \dots, \alpha_k, \alpha_{k+1}) \mid (\alpha_1, \dots, \alpha_k) \in C, B_{p-1} < \alpha_{k+1} < B_p\} \\
 C_{2p} &= \{(\alpha_1, \dots, \alpha_k, \alpha_{k+1}) \mid (\alpha_1, \dots, \alpha_k) \in C, B_p = \alpha_{k+1}\} \\
 C_{2p+1} &= \{(\alpha_1, \dots, \alpha_k, \alpha_{k+1}) \mid (\alpha_1, \dots, \alpha_k) \in C, \alpha_{k+1} > B_p\}
 \end{aligned}$$

となる。ここでは、 C 上の分割をスタック、 C_i を C の子供のセル、 i を C_i のインデックスとよぶ。図 3 は C 上のスタックの様子を表したものである。 C_i の標本点を考えると、 $k+1$ 座標はインデックスが偶数の場合には代数関数で表現される値であるため代数的数となり、インデックスが奇数の場合には区間内の任意をとることができるので計算量を小さくするために通常有理数の値を選択する。したがって、一般にインデックスが奇数のセルのほうが偶数のセルに比べて効率的に計算できることに注意する。

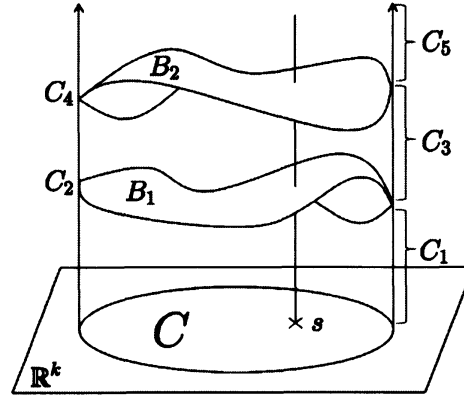


図 3: C 上のスタック

3 パラメトリック多項式最適化問題専用 CAD

QE を用いて最適値関数を求める場合には (3) と (4) の 2 つの QE 問題を解く必要がある。より効率的に最適値関数を求めるため、(3) を入力として、(2) の最適値関数を求める専用 CAD を開発した。

partial CAD [5] は QE 問題に CAD を適用する場合の持ち上げ段階に対する手法で、真偽値をはやく評価することで不要なセルの持ち上げを回避することにより、実行時間の効率化を実現する。専用 CAD は partial CAD の改良で、すべての実行可能領域を求める必要がなくなるため通常の CAD よりも高速に計算が可能となる。

ここではパラメトリック多項式最適化問題 (2) を解く場合を考える。専用 CAD ではまず目的関数の実行可能領域を表す一階述語論理式 (3) を構築する。変数順序を $\theta < y < x$ と設定し、(3) を解くことを考える。射影段階および底段階は通常の CAD と同様に計算を行う。

m を θ の次元とし、 $C \subset \mathbb{R}^m$ をセル、 $s = (s_1, \dots, s_m) = \mathbb{R}^m$ をその標本点とする。セル C の持ち上げにより $c_1, \dots, c_p \subset \mathbb{R}^{m+1}$ が得られたとする。このとき、 c_o が最適なセルとなるとき、それは代数関数で表現されるものなのでインデックス o は偶数で、(3) が真となるすべてのセルのインデックスの値以下である。したがって以下が成立する。

- c_i が (3) で真であるとする、同じスタック上にある c_j ($j > i$) は最適値ではない。
- c_i のインデックス i が奇数であり、(3) が真であるとき、 c_{i-1} が最適値の候補となる。

この性質を用いることで、partial CAD よりも不要なセルの持ち上げを回避することが実現できる。また、計算量の大きい代数拡大体上での演算を回避できる。パラメトリック最適化問題専用 CAD アルゴリズムを Algorithm 1 に示す。

Algorithm 1 パラメトリック最適化問題専用 CAD

Input: パラメトリック最適化問題 Minimize $f(\mathbf{x}, \boldsymbol{\theta})$ subject to $\varphi(\mathbf{x}, \boldsymbol{\theta})$

Output: 最適値関数

```

1:  $\psi = \exists \mathbf{x} (y = f(\mathbf{x}, \boldsymbol{\theta}) \wedge \varphi(\mathbf{x}, \boldsymbol{\theta}))$ 
2: 射影因子および底段階の計算
3: リスト  $L \leftarrow \mathcal{D}_1$  のセルたち
4: while  $L$  が空でない do
5:    $c_i \leftarrow L$  の要素,  $L$  から  $c_i$  を削除
6:   if  $c_i$  の真偽値が未定である then
7:      $c_i$  を持ち上げスタックを構築する. できた子供たちのセルを  $L$  に追加する
8:     partial CAD の手法を用いて  $c_i$  の子供のセルの  $\psi$  に対する真偽値を評価する.
9:     子供の真偽値情報を用いて  $c_i$  および先祖の真偽値情報を設定する.
10:  else if ( $c_i$  の真偽値が真) かつ ( $c_i$  のレベルが  $m+1$ ) then
11:    for  $c_j \in \{c_i \text{ と同じスタック} \}$  do
12:      if  $j > i$  then
13:         $c_j$  の真偽値  $\leftarrow$  偽
14:      end if
15:    end for
16:    if  $i$  が奇数 then
17:       $c_i$  の真偽値  $\leftarrow$  偽
18:       $c_{i-1}$  の真偽値  $\leftarrow$  真
19:    end if
20:  end if
21: end while
22: 真のセルに対応する限量記号のない論理式を構築する.
```

10 行目から 20 行目までが改良部分である。11 行目から 15 行目までの処理により、レベル $m+1$ の真のセルが見つかった場合には、同じスタック上にあるセルの真偽値を決定できるので、それらのセルの持ち上げを回避することが可能となる。また、16 行目から 19 行目の処理により、インデックスが奇数のセルの持ち上げによってインデックスが偶数の真偽値を決定できるため、計算量が大きくなる代数拡大体上での演算の回避につながっている。

3.1 実験結果

SyNRAC 上に実装した通常の CAD アルゴリズムと専用 CAD アルゴリズムをパラメトリック最適化問題に対して適用した場合の実験結果を示す。実験は、Intel(R) Core(TM) i3 CPU U330 1.20 GHz, 2.92 GByte メモリ上で行った。

例 3

以下の非線形パラメトリック最適化問題を考える [10]:

Problem $P(\theta)$, $-\infty < \theta < \infty$:

$$\begin{aligned} \text{Minimize} \quad & g(x_1, x_2, \theta) \equiv 45\theta^2 + 80\theta x_1 + 120\theta + x_2 - 43x_1^2 - 70x_1x_2 - 78x_2^2 \\ \text{subject to} \quad & \theta \geq x_1 + x_2, x_1 \geq 0, x_2 \geq 0, \\ & 15\theta \geq 10x_1 + 19x_2 + 1000000. \end{aligned}$$

目的関数の実行可能領域を求めるための一階述語論理式は以下のようになる。

$$\begin{aligned} \exists x_1 \exists x_2 \quad & (y = g(x_1, x_2, \theta) \wedge \theta \geq x_1 + x_2 \wedge x_1 \geq 0 \wedge x_2 \geq 0 \wedge \\ & 15\theta \geq 10x_1 + 19x_2 + 1000000). \end{aligned} \quad (5)$$

式 (5) に QE を適用すると以下の目的関数の実行可能領域が得られる。

$$\begin{aligned} \psi_{feasible}(\theta, y) \equiv \quad & (3\theta \geq 20000 \wedge 4y \leq 273\theta^2 + 1960480\theta - 17200000000 \wedge \\ & y \geq 45\theta^2 + 120\theta) \vee (312y \leq 14040\theta^2 + 37440\theta + 1 \wedge \\ & 361y \geq -1305\theta^2 + 234043605\theta - 780001900000 \wedge \\ & 2340\theta \geq 15600019) \vee (43y \leq 3535\theta^2 + 5160\theta \wedge \\ & 312y \geq 14040\theta^2 + 37440\theta + 1 \wedge 49\theta \geq 860000). \end{aligned}$$

この計算は 7.54 秒で、50,393 個のセルを生成した。

提案手法を用いると以下の最適値関数を表す論理式が得られる。

$$\begin{aligned} \psi_{opt}(\theta, y) \equiv \quad & (\frac{20000}{3} \leq \theta \leq \frac{7800019}{1170} \wedge y = 45\theta^2 + 120\theta) \vee \\ & (\theta \geq \frac{7800019}{1170} \wedge 361y = -1305\theta^2 + 234043605\theta - 780001900000). \end{aligned}$$

この計算は 3.51 秒で、17,777 個のセルを生成した。

例 4

以下の非線形パラメトリック最適化問題を考える [14, p. 31]:

Problem $P(\theta_1, \theta_2)$, $-\infty < \theta_1 < \infty$, $-\infty < \theta_2 < \infty$:

$$\begin{aligned} \text{Minimize} \quad & f(x_1, x_2) \equiv x_1^3 + 2x_1^2 - 5x_1 + 2x_2^2 - 3x_2 - 6 \\ \text{subject to} \quad & 2x_1 + x_2 \leq 2.5 + \theta_1, \\ & 0.5x_1 + x_2 \leq 1.5 + \theta_2, \\ & x_1 \geq 0, x_2 \geq 0, 0 \leq \theta_1 \leq 0.25, 0 \leq \theta_2 \leq 0.25. \end{aligned}$$

表 1: 実験結果

問題	手法	計算時間 (秒)	生成したセルの数
例 3	従来手法	7.54	50393
	提案手法	3.51	17777
例 4	従来手法	86.03	325613
	提案手法	7.69	35825

目的関数の実行可能領域を求めるための一階述語論理式は以下のようになる.

$$\begin{aligned} \exists x_1 \exists x_2 \quad & (y = f(x_1, x_2) \wedge 2x_1 + x_2 \leq 2.5 + \theta_1 \wedge \\ & 0.5x_1 + x_2 \leq 1.5 + \theta_2 \wedge \\ & x_1 \geq 0 \wedge x_2 \geq 0 \wedge 0 \leq \theta_1 \leq 0.25 \wedge 0 \leq \theta_2 \leq 0.25). \end{aligned} \quad (6)$$

式 (6) に QE を適用すると以下の目的関数の実行可能領域が得られる.

$$\begin{aligned} \psi_{feasible}(\theta_1, \theta_2, y) \equiv \quad & 1728y^2 + 11056y - 47349 \leq 0 \wedge y \leq 2\theta_2^2 + 3\theta_2 - 6 \wedge \\ & 0 \leq \theta_1 \leq \frac{1}{4} \wedge 0 \leq \theta_2 \leq \frac{1}{4}. \end{aligned}$$

この計算は 86.03 秒で, 325,613 個のセルを生成した.

提案手法を用いると以下の最適値関数を表す論理式が得られる.

$$\psi_{opt}(\theta_1, \theta_2, y) \equiv 1728y^2 + 11056y - 47349 = 0 \wedge y \leq -6 \wedge 0 \leq \theta_1 \leq \frac{1}{4} \wedge 0 \leq \theta_2 \leq \frac{1}{4}.$$

この計算は 7.69 秒で, 35,825 個のセルを生成した.

提案手法により不要なセルの生成を回避し, 効率的に最適値関数の計算ができていることが確認できる.

4 動的計画法

4.1 最適性の原理とパラメトリック最適化

動的計画法 (Dynamic Programming: DP) は, 最適化問題の計算効率を高めるために R. E. Bellman によって提案された逐次的手法であり, 問題を解きやすい小問題 (部分問題) の列に分割して繰り返し解くテクニックである [2, 15, 19]. すなわち, 多変数の最適化問題を 1 つ解く代わりに, パラメータを含む 1 変数の最適化問題を繰り返し解くというものである. DP は最短経路問題, ナップサック問題, 巡回セールスマン問題など実世界の様々な最適化問題に対し適用できる. また効率的ではないが線形計画問題も適用範囲であり非常に広い範囲の問題を含んでいる.

DP を解く場合には何らかの方法で小問題であるパラメトリック最適化問題を正確に解くことが必要となる. 確定的な多項式 DP 問題の場合には, ここに QE を用いればシステムチックに解くことができる.

DP の基本的な考え方を示すために, マクシマックス定理について述べる. まず $\mathcal{Y}, \mathcal{Z}$ を 2 つの空でない集合とし, 各 $y \in \mathcal{Y}$ に対し, $\mathcal{Z}$ の空でない部分集合 $\mathcal{Z}(y)$ が対応しているとする. また, 2 つの関数 $f: \mathcal{Y} \times \mathbb{R} \rightarrow \mathbb{R}$ と $g: \mathcal{Y} \times \mathcal{Z} \rightarrow \mathbb{R}$ が与えられているとする.

定理 1

(マクシマックス定理 [8]) 各 $y \in \mathcal{Y}$ に対して, $f(y, \cdot): \mathbb{R} \rightarrow \mathbb{R}$ が単調増加であり, $\max_{y \in \mathcal{Y}} f(y, \max_{z \in \mathcal{Z}(y)} g(y, z))$ が存在するならば, $\max_{y \in \mathcal{Y}} f(y, \max_{z \in \mathcal{Z}(y)} g(y, z))$ も存在して両者は等しい.

定理 1 は目的関数 (非線形であっても構わない) に再帰性と単調性があるときに、「2 変数同時最適化を 2 段階逐次最適化で置き換えることができる」ことを保証している。これが**最適性の原理** (principle of optimality) と呼ばれるものである。ただし、2 段階逐次最適化における第 1 段階 $\max_{z \in \mathcal{Z}(y)} g(y, z)$ は最適値を y の関数として求めるパラメトリック最適化問題であることに注意しなければならない。したがって、小問題に分割したのちパラメトリック最適化問題をどのように解くかについては (非線形の常として) 汎用的な万能アルゴリズムは存在せず、問題に依存する。DP の汎用ソルバーに関する研究は探索が容易な離散変数の問題に焦点を当てており、連続変数の問題に対してはグリッド幅を設定した離散近似が行われる。それは連続変数のパラメトリック最適化問題をシステムティックに解く方法がないことが要因となっている。

以下に例を用いて連続変数の最適化問題に対する DP による解法を紹介する。

例 5

(最適配分問題 [2]) 連続変数の最適化問題を考える。

$$\begin{aligned} & \text{Maximize} && \sum_{i=1}^4 \sqrt{x_i} \\ & \text{subject to} && \sum_{i=1}^4 x_i \leq 1, \\ & && x_i \geq 0 \quad (i = 1, \dots, 4) \end{aligned}$$

DP を用いて解く場合には以下のパラメトリック最適化問題を考える。

Problem $P_n(c)$, $0 \leq c < \infty$:

$$\begin{aligned} & \text{Maximize} && \sum_{i=1}^n \sqrt{x_i} \\ & \text{subject to} && \sum_{i=1}^n x_i \leq c, \\ & && x_i \geq 0 \quad (i = 1, \dots, n) \end{aligned}$$

ここで問題 $P_n(c)$ の最適値を $v_n(c)$ とおく。このとき、各部分問題に対する最適値関数 v_n ($n = 1, \dots, 4$) は次の再帰式を満たす。

$$v_n(c) = \max_{0 \leq x_n \leq c} \{ \sqrt{x_n} + v_{n-1}(c - x_n) \}, \quad 0 \leq c < \infty, \quad n = 2, 3, 4. \quad (7)$$

このような最適値関数を満たす関数方程式は**ベルマン方程式** (Bellman equation) あるいは **DP 方程式** (DP equation) と呼ばれる。再帰式 (7) を用いて部分問題を次のように逐次的に解くことができる。

$$\begin{aligned} v_1(c) &= \max_{0 \leq x_1 \leq c} \sqrt{x_1} = \sqrt{c}, & \pi_1^*(c) &= c/1, \\ v_2(c) &= \max_{0 \leq x_2 \leq c} \{ \sqrt{x_2} + \sqrt{c - x_2} \} = \sqrt{2c}, & \pi_2^*(c) &= c/2, \\ v_3(c) &= \max_{0 \leq x_3 \leq c} \{ \sqrt{x_3} + \sqrt{2(c - x_3)} \} = \sqrt{3c}, & \pi_3^*(c) &= c/3, \\ v_4(c) &= \max_{0 \leq x_4 \leq c} \{ \sqrt{x_4} + \sqrt{3(c - x_4)} \} = \sqrt{4c}, & \pi_4^*(c) &= c/4. \end{aligned}$$

ただし、 π_i^* は上式において第 i 行の右辺を最大にする x_i を c の関数として表したものである。したがって、与問題の最適値 $v_4(1) = \sqrt{4 \cdot 1} = 2$ が得られる。また、最適解 $(x_1^*, x_2^*, x_3^*, x_4^*)$ は次のようにして得られる。

$$\begin{aligned} x_4^* &= \pi_4^*(1) = 1/4, \\ x_3^* &= \pi_3^*(1 - x_4^*) = \pi_3^*(3/4) = 1/4, \\ x_2^* &= \pi_2^*(1 - x_4^* - x_3^*) = \pi_2^*(1/2) = 1/4, \\ x_1^* &= \pi_1^*(1 - x_4^* - x_3^* - x_2^*) = \pi_1^*(3/4) = 1/4. \end{aligned}$$

例 5 は、 $\sqrt{x_i}$ を新しい変数に置き換えることにより、多項式最適化問題に変換することができる。そのため QE を用いて直接解くこともできる。例 1 は QE で例 5 を解いた結果となっている。

4.2 多段階決定過程

現実に遭遇する意思決定問題の多くは、決定という行為が一度で完了せず、一度とった決定の結果から生じる状況の変化に応じて何度も決定を下すという多段階の問題に帰着されることが多い。DP はこの種の多段階決定過程 (multi-stage decision processes) に効果を発揮する。一般に、確定的な状態推移の下での n 段階決定過程は以下のような構造を持つ： i 段階において、状態が $x_i \in \mathcal{X}_i$ であり、決定 $u_i \in \mathcal{U}_i(x_i)$ が選択された場合、コスト $c_i(x_i, u_i)$ が発生し、状態は確定的推移法則 f に従って推移し、次の段階における状態は $x_{i+1} = f_i(x_i, u_i)$ となる。ただし、 $\mathcal{U}_i(x_i)$ は i 段階での状態が x_i であるときに選択可能な決定全体である。図 4 は 3 段階決定過程を表す。したがって、コストの総和を最小化する問題は以下のように定式化される。

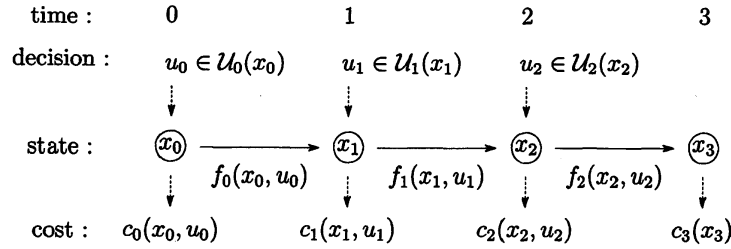


図 4: 3 段階決定過程

Problem $P(x_0)$, $x_0 \in \mathcal{X}_0$:

$$\begin{aligned} & \text{Minimize} \quad \sum_{i=0}^{n-1} c_i(x_i, u_i) + c_n(x_n) \\ & \text{subject to} \quad x_{i+1} = f_i(x_i, u_i), \quad i = 0, 1, \dots, n-1, \\ & \quad \quad \quad u_i \in \mathcal{U}_i(x_i), \quad i = 0, 1, \dots, n-1. \end{aligned}$$

ここで、任意の初期状態 $x \in \mathcal{X}_0$ に対し、 $v_0(x)$ を問題 $P(x)$ の最適値とする。このとき、以下のベルマン方程式を後ろ向きに解くことにより最適値関数 $v_0: \mathcal{X}_0 \rightarrow \mathbb{R}$ を求めることができる。

$$v_n(x) = c_n(x), \quad x \in \mathcal{X}_n, \quad (8a)$$

$$v_i(x) = \min_{u \in \mathcal{U}_i(x)} \{c_i(x, u) + v_{i+1}(x_{i+1})\}, \quad x \in \mathcal{X}_i, \quad i = 0, \dots, n-1 \quad (8b)$$

$$= \min_{u \in \mathcal{U}_i(x)} \{c_i(x, u) + v_{i+1}(f_i(x, u))\}, \quad x \in \mathcal{X}_i, \quad i = 0, \dots, n-1. \quad (8c)$$

部分問題の最適値関数 v_i を求めるためには、式 (8c) の右辺で表現される 1 変数、1 パラメータのパラメトリック最適化問題を解く必要がある。DP はこの部分問題を繰り返し解くことによる解法の正当性は保証している。しかし、部分問題が可解かどうか、あるいはどんな手法が必要かは c_i および f_i に依存する。

4.3 多項式動的計画問題への専用 CAD の適用

DP と QE の特徴をもう一度挙げる。

- DP は与えられた最適化問題を 1 変数、1 パラメータの複数の小問題に分割し、その小問題の正確な最適値関数を要求する。
- QE はパラメトリック多項式最適化問題を正確に解くことができる。しかし、その計算量は最悪の場合、限量記号がついた変数の数に対し二重指数的であり、大きな問題を解くことができない。

DP は正確な最適値関数を要求し、確定的な多項式 DP 問題を解く場合には QE はそれを実現する。また QE は計算量が大きいため問題を分割し、変数の数が少ない問題にすることを要求し、DP はそれを実現する。このように、DP と QE はお互いの不足部分を補完しあう関係にあると考えられる。以下に、DP と QE を組み合わせることにより効率的にとける多項式最適化問題を紹介する。ここで最適値関数を求める際に、専用 CAD を用いることで効率的に計算している。

例 6

タイルを焼く窯の温度に関する最適化問題 [14, p. 152] について考える。

Problem P(x_0), $-\infty < x_0 < \infty$:

$$\text{Minimize } u_0^2 + u_1^2 + u_2^2 + 100(x_3 - 1500)^2 \quad (9a)$$

$$\text{subject to } x_1 = 0.55x_0 + 0.45u_0, \quad x_2 = 0.60x_1 + 0.40u_1, \quad x_3 = 0.65x_2 + 0.35u_2, \quad (9b)$$

$$200 \leq x_1 \leq 400, \quad 500 \leq x_2 \leq 1000, \quad (9c)$$

$$0 \leq u_i \leq 3000 \quad (i = 0, 1, 2). \quad (9d)$$

窯は図 5 に示すように 3 つのオーブンが連結した構成になっている。窯の温度 u_i はオーブン毎に個別に設定できる。最終的なタイルの温度 x_3 が目標温度になるようにうまく u_i を設定する。

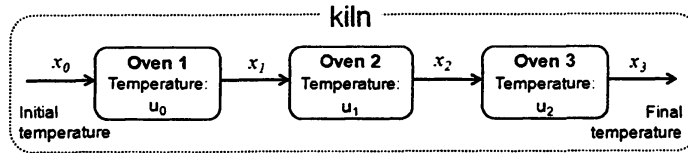


図 5: Schematic diagram of the ceramic kiln

ここで、 x_0 はタイルの初期温度、 x_i ($i = 1, 2, 3$) はオーブン i から出たときのタイルの温度、 u_{i-1} ($i = 1, 2, 3$) はオーブン i の温度を表す。また、式 (9b) は 3 つのオーブンにおける熱伝導現象を表現している。目的関数 (9a) はできるだけ少ないエネルギーでタイルの最終温度を 1500 に近づけることを表現している。式 (9c) はタイルの仕上がりをよくするための制約である。

まず、値関数 v_3 を終端コストで初期化する。

$$v_3(x_3) = 100(x_3 - 1500)^2 \quad (-\infty < x_3 < \infty). \quad (10)$$

次にオーブン 3 での段階は以下の部分問題を考える。

Problem P₂(x_2), $500 \leq x_2 \leq 1000$:

$$\text{Minimize } u_2^2 + v_3(x_3)$$

$$\text{subject to } x_3 = 0.65x_2 + 0.35u_2,$$

$$0 \leq u_2 \leq 3000.$$

x_3 は代入することで消去できるので、P₂(x_2) は以下のように書き換えられる。

Problem P'₂(x_2), $500 \leq x_2 \leq 1000$:

$$\text{Minimize } u_2^2 + v_3(0.65x_2 + 0.35u_2)$$

$$\text{subject to } 0 \leq u_2 \leq 3000.$$

$P'_2(x_2)$ は u_2 が決定変数で x_2 がパラメータの多項式最適化問題である. 専用 CAD を用いることで以下のように正確な最適値関数を効率的に得ることができる.

$$v_2(x_2) = \begin{cases} \frac{169}{4}x_2^2 - 58500x_2 + 29250000 & (500 \leq x_2 \leq \frac{51000}{91}) \\ \frac{169}{53}x_2^2 - \frac{780000}{53}x_2 + \frac{900000000}{53} & (\frac{51000}{91} < x_2 \leq 1000). \end{cases}$$

最適な設定温度 u_2^* は以下ようになる.

$$u_2^* = \pi_2^*(x_2) = \begin{cases} 3000 & (500 \leq x_2 \leq \frac{51000}{91}) \\ -\frac{91}{53}x_2 + \frac{210000}{53} & (\frac{51000}{91} < x_2 \leq 1000). \end{cases}$$

次にオープン 2 におけるパラメトリック最適化問題を考える. これは u_1, x_2 が決定変数で x_1 がパラメータのパラメトリック多項式最適化問題である. 目的関数にオープン 3 での最適値関数 $v_2(x_2)$ を用いる.

Problem $P_1(x_1)$, $200 \leq x_1 \leq 400$:

$$\begin{aligned} &\text{Minimize} && u_1^2 + v_2(x_2) \\ &\text{subject to} && x_2 = 0.60x_1 + 0.40u_1, \\ &&& 500 \leq x_2 \leq 1000, \\ &&& 0 \leq u_1 \leq 3000. \end{aligned}$$

この問題も x_2 を代入により消去することで, 1 変数, 1 パラメータのパラメトリック最適化問題になる.

Problem $P'_1(x_1)$, $200 \leq x_1 \leq 400$:

$$\begin{aligned} &\text{Minimize} && u_1^2 + v_2(0.60x_1 + 0.40u_1) \\ &\text{subject to} && 500 \leq 0.60x_1 + 0.40u_1 \leq 1000, \\ &&& 0 \leq u_1 \leq 3000. \end{aligned}$$

$P'_1(x_1)$ を専用 CAD を用いて解くと, 以下の最適値関数 $v_1(x_1)$ および最適解 u_1^* を得ることができる.

$$\begin{aligned} v_1(x_1) &= \frac{507}{667}x_1^2 - \frac{3900000}{667}x_1 + \frac{7500000000}{667}, \\ u_1^* = \pi_1^*(x_1) &= -\frac{338}{667}x_1 + \frac{1300000}{667}. \end{aligned}$$

最後にオープン 1 におけるパラメトリック最適化問題を考える. ここでは目的関数にオープン 2 での最適値関数 $v_1(x_1)$ を用いる.

Problem $P_0(x_0)$, $-\infty < x_0 < \infty$:

$$\begin{aligned} &\text{Minimize} && u_0^2 + v_1(x_1) \\ &\text{subject to} && x_1 = 0.55x_0 + 0.45u_0, \\ &&& 200 \leq x_1 \leq 400, \\ &&& 0 \leq u_0 \leq 3000. \end{aligned}$$

x_1 を代入により消去することで以下のパラメトリック最適化問題を得る.

Problem $P'_0(x_0)$, $-\infty < x_0 < \infty$:

$$\begin{aligned} &\text{Minimize} && u_0^2 + v_1(0.55x_0 + 0.45u_0) \\ &\text{subject to} && 200 \leq 0.55x_0 + 0.45u_0 \leq 400, \\ &&& 0 \leq u_0 \leq 3000. \end{aligned}$$

P'_0 はパラメータは x_0 , 決定変数は u_0 であり, 専用 CAD を用いて目的関数の最適値および最適解を x_0 の関数として求める.

$$v_0(x_0) = \begin{cases} \frac{121}{81}x_0^2 - \frac{88000}{81}x_0 + \frac{556634680000}{54027} & (-\frac{23000}{11} \leq x_0 \leq -\frac{4818830}{7337}) \\ \frac{61347}{307867}x_0^2 - \frac{858000000}{307867}x_0 + \frac{3000000000000}{307867} & (-\frac{4818830}{7337} < x_0 \leq -\frac{1740160}{7337}) \\ \frac{121}{81}x_0^2 - \frac{176000}{81}x_0 + \frac{530398720000}{54027} & (-\frac{1740160}{7337} < x_0 \leq \frac{8000}{11}), \end{cases}$$

$$u_0^* = \pi_0^*(x_0) = \begin{cases} -\frac{11}{9}x_0 + \frac{4000}{9} & (-\frac{23000}{11} \leq x_0 \leq -\frac{4818830}{7337}) \\ -\frac{50193}{307867}x_0 + \frac{351000000}{307867} & (-\frac{4818830}{7337} < x_0 \leq -\frac{1740160}{7337}) \\ -\frac{11}{9}x_0 + \frac{8000}{9} & (-\frac{1740160}{7337} < x_0 \leq \frac{8000}{11}). \end{cases}$$

これが与えられた最適化問題の最適値および最適解になる.

与えられた最適化問題は多項式最適化問題なので, 理論的には QE を用いて解くことは可能である. しかし, CAD を用いて解いた場合には変数の数が多いために射影段階で射影因子の数が増加し, 1 時間以上経過しても計算が停止しなかった. しかし, 例 6 で現れる部分問題は 1 変数, 1 パラメータであり, CAD による QE の計算はすべて 1 秒以内に停止した.

例 6 のように DP を用いることで直接 QE では解けなかった問題を効率的に解くことができるようになる. ただし, 段の数 n が増えると最適値関数の場合分けが増えることになる. 確定的な多項式 DP 問題に対する QE を用いたツールの実装はまだできておらず, どの程度の規模の問題が解けるかはまだ確認できていない.

5 おわりに

QE を用いたパラメトリック多項式最適化問題の解法を紹介した. より効率的に問題を解くために, パラメトリック多項式最適化問題専用 CAD アルゴリズムを提案し, 実験結果によりその効果を示した. 専用 CAD アルゴリズムの適用事例として確定的多項式動的計画問題を紹介した.

参考文献

- [1] H. Anai and K. Yokoyama. Cylindrical algebraic decomposition via numerical computation with validated symbolic reconstruction. In A. Dolzmann, A. Seidl, and T. Sturm, editors, *Algorithmic Algebra and Logic*, pages 25–30, 2005.
- [2] R. E. Bellman. *Dynamic Programming*. Princeton University Press, Princeton, NJ, 1957.
- [3] C. W. Brown. *Solution formula construction for truth invariant cad's*. PhD thesis, University of Delaware Newark, 1999.
- [4] G. E. Collins. Quantifier elimination for real closed fields by cylindrical algebraic decomposition. In *LNCS*, volume 32. Springer Verla, 1975.
- [5] G. E. Collins and H. Hong. Partial cylindrical algebraic decomposition for quantifier elimination. *Journal of Symbolic Computation*, 12(3):299–328, 1991.
- [6] G. E. Collins, J. R. Johnson, and W. Krandick. Interval arithmetic in cylindrical algebraic decomposition. *Journal of Symbolic Computation*, 34(2):145–157, 2002.

- [7] J. H. Davenport and J. Heintz. Real quantifier elimination is doubly exponential. *Journal of Symbolic Computation*, 5(1/2):29–35, 1988.
- [8] S. Iwamoto. Sequential minimaximization under dynamic programming structure. *Journal of Mathematical Analysis and Applications*, 108(1):267–282, 1985.
- [9] H. Iwane, A. Kira, and H. Anai. Construction of explicit optimal value functions by a symbolic-numeric cylindrical algebraic decomposition. In V. P. Gerdt, W. Koepf, E. W. Mayr, and E. V. Vorozhtsov, editors, *CASC*, volume 6885 of *Lecture Notes in Computer Science*, pages 239–250. Springer, 2011.
- [10] H. Iwane, H. Yanami, and H. Anai. An effective implementation of a symbolic-numeric cylindrical algebraic decomposition for optimization problems. In *Proceedings of the 2011 International Workshop on Symbolic-Numeric Computation*, volume 1, pages 168–177, 2011.
- [11] H. Iwane, H. Yanami, and H. Anai. A symbolic-numeric approach to multi-objective optimization in manufacturing design. *Mathematics in Computer Science*, 5(3):315–334, 2011.
- [12] 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.
- [13] R. Loos and V. Weispfenning. Applying linear quantifier elimination. *The Computer Journal*, 36(5):450–462, 1993.
- [14] E. Pistikopoulos, M. Georgiadis, and V. Dua. *Multi-parametric programming: theory, algorithms, and applications*, volume 1 of *Multi-Parametric Programming*. Wiley-VCH, 2007.
- [15] M. Sniedovich. *Dynamic Programming: Foundations and Principles; 2nd ed.* Pure and Applied Mathematics. CRC Press, Hoboken, 2010.
- [16] A. W. Strzeboński. Cylindrical algebraic decomposition using validated numerics. *Journal of Symbolic Computation*, 41(9):1021–1038, 2006.
- [17] T. Sturm and A. Weber. Investigating generic methods to solve hopf bifurcation problems in algebraic biology. In K. Horimoto, G. Regensburger, M. Rosenkranz, and H. Yoshida, editors, *Algebraic Biology*, volume 5147 of *Lecture Notes in Computer Science*, chapter 15, pages 200–215. Springer Berlin / Heidelberg, Berlin, Heidelberg, 2008.
- [18] V. Weispfenning. Quantifier elimination for real algebra - the quadratic case and beyond. *AAECC*, 8:85–101, 1993.
- [19] 岩本誠一. 動的計画論. 経済工学シリーズ. 九州大学出版会, 1987.
- [20] 穴井宏和, 横山和弘. *QE の計算アルゴリズムとその応用 - 数式処理による最適化*. 東京大学出版会, 8 2011.