

グレブナ基底計算を効率的に行うための 項順序自動設定法

沢田 浩之*

産業技術総合研究所

(RECEIVED 2001/12/14 REVISED 2002/9/2)

Abstract

It is well known that term order has a great influence on computational efficiency of Gröbner bases. This paper proposes a new method of determining term order that enables efficient computation of Gröbner bases. The advantages of this method have been shown through experiments. The results of experiments have also illustrated that degree-reverse-lexicographic ordering is not always good for computational efficiency, and that, even in the case of degree-reverse-lexicographic ordering, the order of variables has a significant effect on efficiency of Gröbner basis computation.

1 はじめに

グレブナ基底の計算において、用いられる項順序がその計算効率に大きな影響を持つことはすでによく知られている [7]。与えられた項順序におけるグレブナ基底を直接計算することは往々にして非効率的であるため、それに代わる方法の1つとして、基底変換法の研究が進められている [2, 5]。基底変換法では、以下の手順に従ってグレブナ基底計算が行われる。

1. 計算効率のよい項順序を用いてグレブナ基底を計算する。
2. 1で得られたグレブナ基底を用いて、与えられた項順序のグレブナ基底を計算する。

従来、いわゆる「計算効率のよい項順序」として、全次数逆辞書式順序 (degree reverse lexicographic ordering) を用いることが多いが [5]、以下のような問題が存在する。

- 入力多項式に線形項と非線形項が混在するような場合、全次数逆辞書式順序は必ずしも効率的な項順序を与えるわけではない。

*h.sawada@aist.go.jp

- 全次数逆辞書式順序を用いるとしても、特に変数の数が多い場合には、その変数相互の順序関係によって計算効率が大きく異なることがある。

本論文では、グレブナ基底を効率的に計算する項順序の自動設定方法を提案するとともに、その実験結果について報告する。まず、次章で、提案する項順序設定法について述べる。第3章では、その項順序設定法を用いたグレブナ基底の計算手順について説明する。その後第4章において、4つの異なる問題を用いて、本手法の有効性について検証する。

なお、本手法で設定する項順序はブロック式順序 (block order)[5] とし、各ブロック内における項順序は全次数逆辞書式順序とする。

記法 1 (ブロック式順序)

変数 y_1, \dots, y_η が上位ブロック、変数 z_1, \dots, z_ζ が下位ブロックに属し、各ブロックにおいて、「 $i < j \Leftrightarrow y_i$ は y_j よりも上位」および「 $i < j \Leftrightarrow z_i$ は z_j よりも上位」であるとき、これを以下のように記す。

$$y_1 > \dots > y_\eta \gg z_1 > \dots > z_\zeta$$

2 項順序設定法

与えられた連立方程式のグレブナ基底を計算する際、従属変数を独立変数よりも上位のブロックへ配置することが計算効率上望ましいということは、経験的によく知られている。これを、より形式的に表現すると以下ようになる。

与えられた多項式集合に含まれるある多項式 $f(x_1, \dots, x_n)$ が $x_n - g(x_1, \dots, x_{n-1})$ (g は x_1, \dots, x_{n-1} の多項式) の形をしているとき、 x_n を他の変数よりも上位のブロックへ配置することが計算効率上望ましい。

本論文で提案する方法は、個々の入力多項式からこのような順序関係を導き出し、それをもとにして変数をブロックへ振り分け、その後各ブロック内における全次数逆辞書式順序を設定するものである。

項順序の設定は以下の手順で行われる。

1. 前処理
2. 変数のブロックへの振り分け
3. ブロック内順序の設定
4. 後処理

以降、各節においてそれぞれの具体的な方法について述べる。

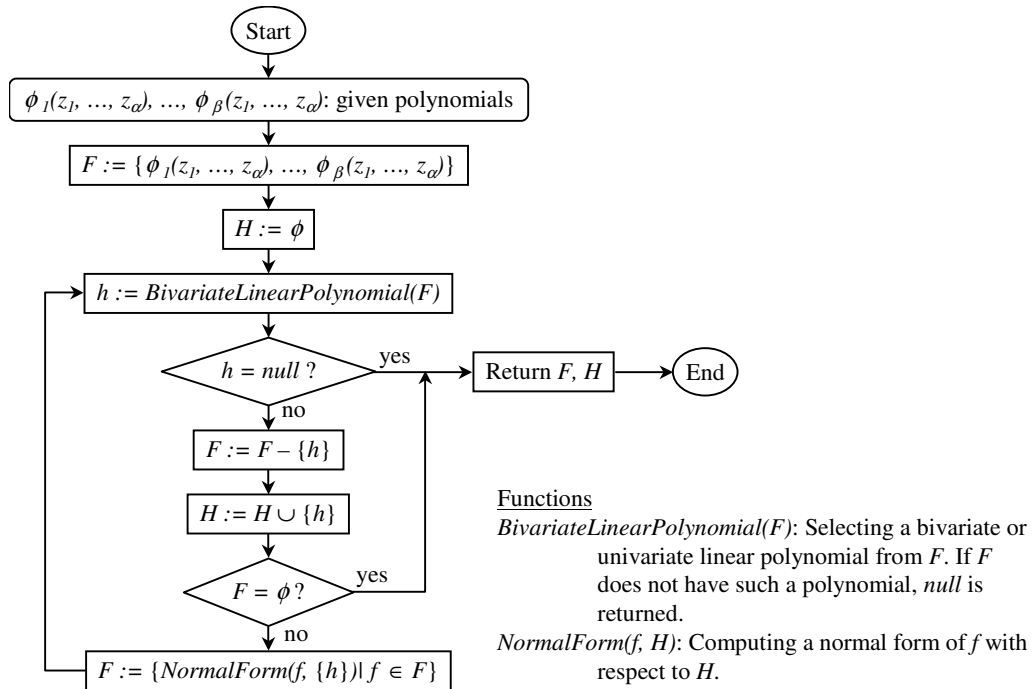


図 1: 前処理-2変数線形多項式による逐次簡約化

2.1 前処理

前処理として、2変数線形多項式を逐次選び出し、それを用いて他の多項式に対する簡約化を施す。その手順を図1に示す。この操作は以下の意味を持つ。

グレブナ基底の計算過程において、本質的に同一の順位を持つことになる変数が2つ以上存在する場合には、余分な変数を予め消去しておく。

これについての考察を以下に述べる。

与えられた多項式集合の中に、次式のような2変数線形多項式が含まれているとする。

$$ax + by + c. \quad (1)$$

ただし、 a 、 b 、 c は定数である。また、仮に x の順位は y よりも上位であるものとする。グレブナ基底計算過程において行われる式(1)による簡約化は、本質的に、 x に対して $-(b/a)y - (c/a)$ を代入することに他ならない。すなわち、式(1)による簡約化によって x は y に置き換えられてしまうため、 x に与えられた順位は以降の計算において意味を持たなくなる。結果として、その計算効率は、 x と y が同じ順位を与えられた場合と大差がなくなる。

本質的に同一の順位を持つことになる複数の変数を残しておくことは、計算効率向上に有効な項順序を決定する上で混乱を引き起こす可能性があるため、2変数線形多項式による簡約化を事前に施しておくことが必要となる。

ここで、この前処理の目的は変数の数を減らすことではない、ということを改めて強調しておきたい。グレブナ基底計算過程で行なわれる2変数線形多項式による簡約化が本質的に変数の1対1の代入であり、それゆえ、それらの変数に与えられた順位が計算効率上意味を持たなくなるという事態を避けるために、この前処理が必要となるのである。変数の数を減らすことを目的とするならば、与えられた多項式集合に含まれるすべての線形多項式による簡約化を施すことが望ましいが、一般に、これは変数の1対1の代入とはならない。それどころか、グレブナ基底計算の一部を、暗黙のうちにある項順序を仮定して実行することにつながるため、本論文で提案する方法にとってはむしろ不適切といえる。

なお、変数に与えられた順位が計算効率上意味を持たなくなるという意味では、1変数線形多項式も同様である。したがって、図1に示すフローチャートでは、1変数線形多項式による簡約化も行うように記述されている。

2.2 変数のブロックへの振り分け

すでに述べたように、与えられた多項式集合に $x_n - g(x_1, \dots, x_{n-1})$ (g は多項式) の形のものが含まれているとき、 x_n を他の変数よりも上位のブロックへ配置することが計算効率上望ましいことは、経験的によく知られている。これは、以下のように拡張できる。

経験則 1

与えられた多項式集合に $x_i - g_i(x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n)$ ($i = k, k+1, \dots, n$, g_i は多項式) なる形の多項式が含まれているとき、 x_k, \dots, x_n を x_1, \dots, x_{k-1} よりも上位のブロックへ配置することが、計算効率上望ましい。

また、経験則1を補完する意味で、経験則2をおく。

経験則 2

与えられた多項式集合に、どのような i についても $x_i - g(x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n)$ (g は多項式) の形にはならない多項式が含まれているとき、 x_1, \dots, x_n を同一のブロックへ配置することが、計算効率上望ましい。

ここで提案する方法では、まず、与えられた多項式集合 F を、上述の経験則1に当てはまる多項式の集合 P と経験則2に対応する多項式の集合 Q に分ける。

$$\begin{aligned} F &= \{f_1(x_1, \dots, x_n), \dots, f_m(x_1, \dots, x_n)\}, \\ P &= \{f \in F \mid \exists i \text{ } f \text{ は } x_i - g(x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n) \text{ と表現可能 } (g \text{ は多項式})\}, \quad (2) \\ Q &= F - P. \end{aligned}$$

次に、 P に含まれる個々の多項式について、変数ブロックの上位-下位関係を示す有向グラフを作成する。そして、このようにして作成された有向グラフを組み合わせることによって、多項式集合全体に関する有向グラフを作成し、これに示される上位-下位関係に基づいて変数が割り振られるブロックを決定する。そのようなブロックが一意に決定しない場合には、上述の経験則2を Q に含まれる多項式について適用することにより、ブロックを決定する。

変数のブロックへの割り振りは、以下の手順で行われる。

1. 有向グラフの作成
2. 閉路の解消
3. 各変数の割り振り先ブロックの決定

以降の各項では、これらの個々の手順の詳細について述べる。

2.2.1 有向グラフの作成

変数ブロックの上位-下位関係を示す有向グラフの作成例を図 2 に示す。図 2 は、式 (3) に示す多項式集合 F が与えられたときに、その部分集合 P に基づいて作成される有向グラフを示しており、グラフの有向辺は、上位の変数から下位の変数へ向かっている。

$$\begin{aligned}
 F &= \{z^2 - 2x + z + 3u - 1, x^2 + x - y - 3, 2y^2 - y + 2z + u, xu - yz - 1\}, \\
 P &= \{z^2 - 2x + z + 3u - 1, x^2 + x - y - 3, 2y^2 - y + 2z + u\}, \\
 Q &= \{xu - yz - 1\}.
 \end{aligned} \tag{3}$$

このように、経験則 1 に基づいて個々の多項式に関する有向グラフが作成され、それらを組み合わせることによって多項式集合全体に関する有向グラフが作成される。各変数が割り当てられるブロックは、この有向グラフで表される上下関係に基づいて決定される。

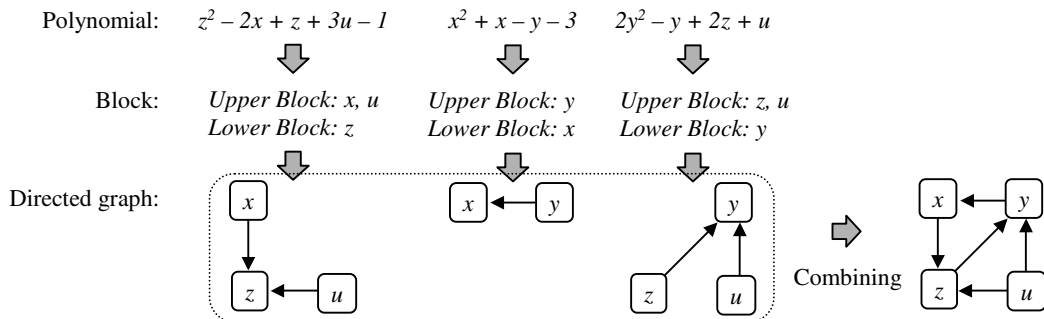


図 2: 有向グラフの作成

有向グラフが閉路を持つ場合、原則として、同一閉路上に存在する変数は同じブロックに割り振られる。経験則 1 に照らしてみた場合、この割り振り方は妥当なものだと考えられるが、2 つの変数ノードを直接結ぶ往復路に対しては必ずしもそうではない。以下に例を示す。

2 つの変数ノードを直接結ぶ往復路が存在する場合における変数のブロック割り振りの例次式 (4) のような多項式が与えられているものとする。

$$x - f(z), y - g(z), z - h(x, y). \tag{4}$$

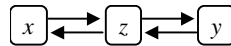


図 3: 2 つの変数ノードを直接結ぶ往復路を含む有向グラフの例

ただし、 $f(z)$ 、 $g(z)$ 、 $h(x, y)$ は非線形多項式である。この場合、作成される有向グラフは図 3 となる。「同一閉路上に存在する変数を同一ブロックに割り振る」という原則に従えば、 x 、 y 、 z はすべて同一ブロックに割り振られる。しかしながら、式 (4) の場合、 x と y を z よりも上位のブロックへ割り振れば、そのグレブナ基底は式 (5) となって直ちに求められる。

$$x - f(z), y - g(z), z - h(f(z), g(z)). \quad (5)$$

これは、 x 、 y 、 z を同一ブロックへ割り振った場合よりも、明らかに計算効率がよい。

したがって、2 つの変数ノードを直接結ぶ往復路が存在する場合には、いずれかの有向辺を削除しておくことが、計算効率を向上させる上で有効と考えられる。本論文で提案する方法では、削除すべき有向辺を、以下の原則に従って選定することにする。

削除すべき有向辺を選定するための原則

往復路を構成する 2 つの変数ノードについて、その近傍にある他の変数ノードとの上下関係を考えたとき、より下位におくことが望ましいと思われる変数ノードから、より上位に置くことが望ましいと思われる変数ノードへ向かう有向辺を削除する。換言すると、上位に置くことが望まれる変数ノードは上位に、下位に置くことが望まれる変数ノードは下位になるように、往復路の一方を取り除く。

削除すべき有向辺の具体的な選定方法として、本研究では、次の 3 つについて実験的検討を行った。

1. 各変数ノードに向かう有向辺の数に着目
ある変数ノードへ向かう有向辺の数が多いということは、そのノードよりも上位に配置される変数ノードが多いということになる。したがって、そのような変数ノードは、下位に置くことが望ましいと考えられる。
2. 各変数ノードから出る有向辺の数に着目
ある変数ノードから出る有向辺の数が多いということは、そのノードよりも下位に配置される変数ノードが多いということになる。したがって、そのような変数ノードは、上位に置くことが望ましいと考えられる。
3. 各変数ノードに向かう有向辺とその変数ノードから出る有向辺の数の差に着目
ある変数ノードに向かう有向辺とその変数ノードから出る有向辺の数の差は、近傍の変数ノードの中におけるその変数ノードの相対的な順位を表すと考えられる。この値が大きい変数ノードは、その近傍における相対的な順位が低くなるので、より下位に置くことが望ましいと考えられる。

実験的検討の結果、1の方法が最も有効だったので、本論文で提案する方法ではこれを採用している。また、それにともない、以下に定義する往復路除去定数を導入する。

定義 1 (往復路除去定数)

変数のブロックへの割り振りに際し、1以上の定数である「往復路除去定数」を定義する。

2つの変数ノード X と Y を直接結ぶ往復路が存在するとき、次式 (6) が成立すれば、 X から Y へ向かう有向辺は削除される。

$$\frac{(X \text{ へ向かう有向辺の数})}{(Y \text{ へ向かう有向辺の数})} > (\text{往復路除去定数}). \quad (6)$$

例えば、式 (4) および図 3 の場合、各ノードへ向かう有向辺の数は以下ようになる。

	{x}	{y}	{z}
各ノードへ向かう有向辺の数	1	1	2

したがって、仮に往復路除去定数を 1 とすると、 z から x および y へ向かう有向辺は削除される。結果として、 x および y は z よりも上位のブロックへ割り振られることになり、計算効率のよいブロック割り振りが実現される。

2.2.2 閉路の解消

第 2.2.1 項で作成された有向グラフに閉路が存在する場合、同一閉路上に存在する変数ノードはすべて 1 つのノードにまとめられる。その例を図 4 に示す。本論文では、閉路の特定に

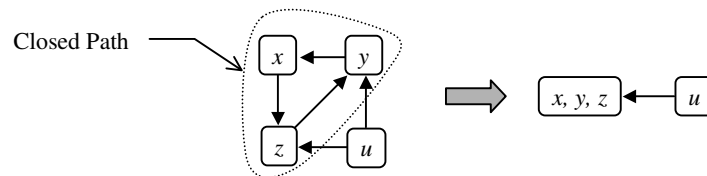


図 4: 閉路の解消

は隣接行列を使った方法 [3] を用いている。閉路の解消は逐次行われ、有向グラフ中の閉路がすべてなくなるまで繰り返される。

2.2.3 各変数の割り振り先ブロックの決定

第 2.2.2 項での操作により、有向グラフにはすでに閉路は存在しない。したがって、この有向グラフは開始ノードと終端ノードを持つ。開始ノードおよび終端ノードは、それぞれ 1 つとは限らない。本論文で提案する方法では、開始ノードから終端ノードまでの最大距離を求め、それに基づいてブロックの数を決定し、それらのブロックに各変数を割り振っていく。この操作について、図 5 を用いて説明する。

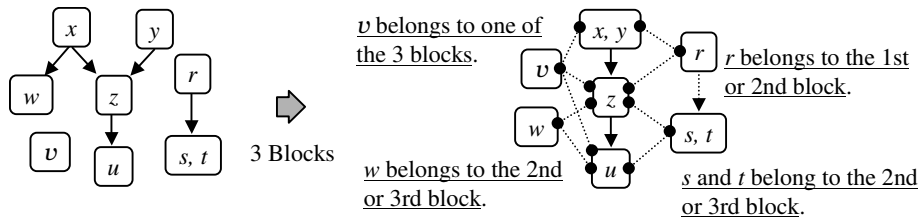


図 5: 各変数ノードの割り振り先ブロックの決定

まず、有向グラフの開始ノードをすべて選び出し、各開始ノードから終端ノードまでの最大距離を計算する。図 5 の場合、開始ノードは $\{x\}$ 、 $\{y\}$ 、 $\{r\}$ の 3 つである。終端ノードまでの距離は、 $\{x\} \rightarrow \{z\} \rightarrow \{u\}$ および $\{y\} \rightarrow \{z\} \rightarrow \{u\}$ の 2 経路の場合に最大となり、その値は 2 である。これにより、ブロックの数は 3 と決定される。

次に、有向グラフで示される上位-下位関係と矛盾しないように、各ブロックへ変数ノードが割り振られる。図 5 の場合、 $\{x\}$ および $\{y\}$ は最上位ブロック、 $\{z\}$ は第 2 ブロック、 $\{u\}$ は第 3 ブロックへの割り振りが一意的に決定される。一方、 $\{v\}$ 、 $\{w\}$ 、 $\{r\}$ 、 $\{s, t\}$ の 4 つの変数ノードについては、経験則 1 のみに基づいた方法では割り振り先が一意には決定しない。

これらの変数ノードについては、式 (2) で定義された多項式集合 Q に対して経験則 2 を適用することによって、その割り振り先ブロックを決定する。その手順は以下の通りである。

変数ノードの割り振り先ブロック決定手順

1. 有効グラフで示される上位-下位関係に基づき、各変数ノードについて、割り振り可能なブロック範囲を決定する。

例えば図 5 の場合、割り振り可能なブロック範囲は以下のようになる。

	$\{v\}$	$\{w\}$	$\{r\}$	$\{s, t\}$
割り振り可能なブロック範囲	1 ~ 3	2 ~ 3	1 ~ 2	2 ~ 3

2. すべての変数ノードについて割り振り先ブロックが一意に決定していたら、それらの変数を対応するブロックへ実際に割り振って終了。
3. 割り振り先未定の変数ノードの中から、最も上位のブロックへ割り振られる可能性があるものを選び出す。選び出されたノードを N とし、 N に含まれる変数の集合を V_N とする。

4. 式 (2) で定義された多項式集合 Q の中から、 V_N に含まれる変数を持つものをすべて選び出す。選び出された多項式に含まれる変数の集合を V_Q とする。

経験則 2 に従えば、ノード N に含まれる変数は、 V_Q 内の変数が含まれているブロックに配置されることが望ましいことになる。

5. 各ブロック B について、 V_Q に含まれる変数が占める割合を計算する。

$$(V_Q \text{ に含まれる変数が占める割合}) = \frac{(B \text{ および } V_Q \text{ に含まれる変数の数})}{(B \text{ に含まれる変数の数})}$$

6. 「 V_Q に含まれる変数が占める割合」の最も大きなブロックを、ノード N の割り振り先ブロックとして決定する。そのようなブロックが複数存在する場合には、その中で最上位のブロックを割り振り先とする。なお、この時点では、変数のブロックへの割り振りは、実際には行わない。
7. ステップ 1 へ進む。

2.3 ブロック内順序の設定

ブロック割り振りとは異なり、計算効率のよいブロック内順序を与える経験則として広く知られているものは、現時点では存在しない。著者は、多項式集合の複雑さ、延いてはグレブナ基底計算の効率に大きく関わる要素として、1) 各変数の次数、2) 各単項の全次数、3) 単項の個数、に着目し実験的検討を行った。その結果、ブロック内順序を以下の手順で決定した場合、グレブナ基底計算が比較的効率よく行われることを見出した。

ブロック内の全次数逆辞書式順序決定手順

1. 各変数について、与えられた多項式集合 F における最大次数を求める。最大次数の小さい変数を上位にする。最大次数が等しい変数が複数存在する場合、ステップ 2 の規準に従って順序関係を決定する。
2. 各変数について、 F に含まれる各多項式が持つ単項を、その変数の次数ごとにグループ分けする。そして、各グループに含まれる単項の数にその変数の次数を乗じたものの総和を計算する。この値が大きい変数を上位とする。
この値は、変数の出現回数をその次数で重み付けしたものであるといえる。

例えば、式 (7) に示す多項式集合 Katsura-2 [4] の場合、ブロック内順序の決定に用いられる各値は以下ようになる。

$$\{u_0 + 2u_2 + 2u_1 - 1, 2u_1u_0 + 2u_1u_2 - u_1, u_0^2 - u_0 + 2u_2^2 + 2u_1^2\}. \quad (7)$$

	u_0	u_1	u_2
最大次数	2	2	2
1 次の単項の数	3	4	2
2 次の単項の数	1	1	1
次数で重み付けされた出現回数	5	6	4

これにより、項順序は $u_1 > u_0 > u_2$ と決定される。

2.4 後処理

第 2.2 節で述べたブロック決定法は、基本的に経験則 1 に基づいたものであり、経験則 1 は経験則 2 よりも優先することを暗黙のうちに仮定したものであった。しかしながら、経験則 2 に基づいて同一ブロックへの割り振りを示唆する多項式が多数存在するような場合には、これを優先させる方が合理的と考えられる。

ここで提案する後処理とは、前節までの手順で決定された項順序を、各多項式によって示唆されるブロック割り振りと照らし合わせ、その結果によって、隣接するブロックを 1 つのブロックに結合するものである。

定義 2 (ブロック結合定数)

隣接するブロックを結合すべきか否かを判定するために、「ブロック結合変数」を定義する。

隣接するブロック B_1, \dots, B_j に対し、式 (8) が成立するとき、これらのブロックは 1 つのブロックとして結合される。

$$\frac{(B_1, \dots, B_j \text{ の結合を示唆する多項式の数})}{(B_1, \dots, B_j \text{ と矛盾しないブロックを示唆する多項式の数})} > (\text{ブロック結合変数}). \quad (8)$$

なお、結合されたブロック内の変数の順序は、結合前の順序と同一とする。

3 グレブナ基底計算手順

グレブナ基底は以下の手順で計算される。

グレブナ基底計算手順

1. 与えられた多項式集合 Φ に対して、第 2.1 節で述べた前処理を施す。その結果得られた 1 変数あるいは 2 変数の線形多項式集合を H 、それ以外の多項式の集合を F とする。 F に含まれる変数を x_1, \dots, x_n とする。また、 H に含まれ、かつ、 F に含まれない変数を y_1, \dots, y_γ とする。
2. 第 2.2 節から第 2.4 節までに述べた方法に従い、 F に関する項順序 T_F を決定する。
3. 項順序 T_F のもとで、 F のグレブナ基底 G を計算する。
4. $G \cup H$ は、項順序 $y_1 > \dots > y_\gamma \gg T_F$ のもとにおける Φ のグレブナ基底である。

$G \cup H$ が項順序 $y_1 > \dots > y_\gamma \gg T_F$ のもとにおける Φ のグレブナ基底であることは、以下のようにして示すことができる。

$G \cup H$ により生成されるイデアルが Φ により生成されるイデアルと同一であることは明らかなので、 $G \cup H$ がグレブナ基底であることを示せばよい。 F に含まれるすべての多項式は、 H に含まれる線形多項式によりすでに簡約化されている。したがって、 H に含まれる各多項式の頭項は y_1, \dots, y_γ のいずれかであり、 G に含まれる多項式の頭項とは互いに素となる。また、第 2.1 節で述べた手続き上、 H に含まれる複数の線形多項式が同一の頭項を持つことはない。すなわち、 $G \cup H$ に含まれる多項式により生成されるすべての S -多項式は $G \cup H$ によって 0 に簡約化されることが保証される。ゆえに、 $G \cup H$ は Φ のグレブナ基底である。

4 評価実験

本論文で提案する項順序設定法の有効性を検証するため、以下の 4 つの例題を用いて評価実験を行った。それぞれの多項式集合の詳細については、末尾の付録を参照されたい。

実験に用いた例題

1. katsura-8 [4] (9 variables and 9 polynomimals)

Variables: $u_0, u_1, u_2, u_3, u_4, u_5, u_6, u_7, u_8$

2. Heatpump [9] (21 variables and 20 polynomials)

Variables: $p_{220}, p_{223}, p_{240}, p_{249}, p_{259}, p_{275}, p_{276}, p_{277}, p_{278}, p_{283}, x_{11}, x_{12}, x_{13}, x_{14}, x_{15},$
 $x_{16}, x_{17}, x_{18}, x_{19}, x_{20}, x_{21}$

3. Mckay [6] (4 variables and 20 polynomials)

Variables: a_1, a_2, a_3, a_5

4. Robot [9] (49 variables and 49 polynomials)

Variables: $p_{385}, p_{440}, x_{41}, x_{42}, x_{43}, x_{44}, x_{45}, x_{46}, x_{47}, x_{49}, x_{50}, x_{51}, x_{52}, x_{53}, x_{54}, x_{55},$
 $x_{57}, x_{58}, x_{59}, x_{60}, x_{61}, x_{62}, x_{63}, x_{64}, x_{65}, x_{66}, x_{67}, x_{68}, x_{69}, x_{70}, x_{71}, x_{72}, x_{73},$
 $x_{74}, x_{75}, x_{76}, x_{77}, x_{78}, x_{79}, x_{80}, x_{81}, x_{82}, x_{83}, x_{84}, x_{85}, x_{86}, x_{87}, x_{88}, x_{89}$

評価は、本手法により生成された項順序によるグレブナ計算時間と、以下の 2 通りの項順序による計算時間を比較することによって行われた。

評価の方法

- (a) ランダムに生成された全次数逆辞書式順序と比較
これとの比較により、主としてブロック設定の有効性を検証する。
- (b) 本手法で生成したブロックのもとでブロック内順序をランダム生成したものと比較
これとの比較により、主としてブロック内順序設定の有効性を検証する。

また、本手法における 2 つの制御パラメータである往復路除去定数 (定義 1) およびブロック結合定数 (定義 2) には、以下の値を用いた。

制御パラメータの値

往復路除去定数: 1

ブロック結合定数: 1

本実験は、以下の計算機環境およびソフトウェアを用いて行われた。

実験に用いた計算機環境およびソフトウェア

CPU: Pentium-4 2GHz

RAM: 2 GB

OS: Windows 2000

グレブナ基底計算ソフトウェア: Risa/Asir [8] および CoCoA [1]

なお、これらのソフトウェアは、それぞれ以下の WEB ページより入手できる。

Risa/Asir Risa/Asir ホームページ <http://risa.cs.ehime-u.ac.jp/>

CoCoA CoCoA System ホームページ <http://cocoa.dima.unige.it/>

表 1: 実験結果一覧

例題	ソフトウェア	計算時間	全次数逆辞書式順序との比較 (順位)	同一ブロック構成の項順序との比較 (順位)
Katsura-8	Risa/Asir	6 分 32 秒	8/100	左に同じ (*1)
	CoCoA	137 分 21 秒	8/100	左に同じ (*1)
Heatpump	Risa/Asir	1 秒	– (*2)	3/100
	CoCoA	8 秒	– (*2)	2/100
Mckay	Risa/Asir	12 時間 2 分	2/24	左に同じ (*1)
	CoCoA	– (*3)	– (*3)	– (*1, *3)
Robot	Risa/Asir	1 分 7 秒	– (*2)	1/100
	CoCoA	9 時間 17 分	– (*4)	– (*4)

- *1 生成されたブロックが1つだったので、全次数逆辞書式順序についてのみ実験を行った。
- *2 10 通りの全次数逆辞書式順序について計算した結果、いずれも計算時間が5時間を越えた。これ以上の計算は比較上無意味と判断し、処理を打ち切った。
- *3 生成された項順序を含め、いずれの項順序を用いても計算時間が5日を越えた。その時点で処理を打ち切った。
- *4 10 通りの全次数逆辞書式順序ならびに 10 通りの同一ブロック構成の項順序について計算した結果、いずれも計算時間が2日を越えた。これ以上の計算は比較上無意味と判断し、処理を打ち切った。

表 1 に実験結果の一覧を、図 6 から図 11 に実験結果の詳細を示す。各図には、本手法で生成された項順序、その項順序におけるグレブナ基底計算時間、他の項順序と計算時間を比較した場合の順位、そして計算時間のヒストグラムが示されている。それぞれのヒストグラムでは、横軸は計算時間、縦軸は度数を表している。計算時間は CPU 時間と GC 時間を合計したものである。

これらの実験結果から、本手法で生成された項順序が、グレブナ基底を効率的に計算する上で有利なものであることがわかる。なお、本手法による項順序設定に要した計算時間はいずれも数秒であった。

図 12 から図 15 は、より詳細なデータとして、計算時間と簡約化が行われた多項式の数、ならびに係数のビット長の和の最大値との相関関係を示したものである。図中、灰色の四角が、本手法で生成された項順序のデータを表している。これらのデータは、Risa/Asir によ

Generated Term Order: $u_1 > u_2 > u_3 > u_4 > u_5 > u_6 > u_7 > u_0 > u_8$

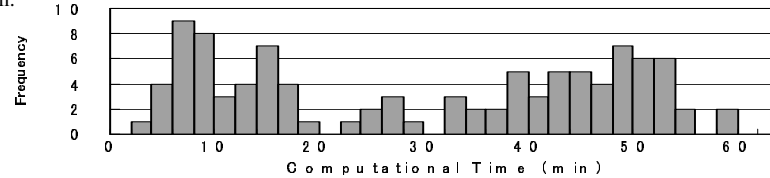
Computational Time: 6 min 32 sec

Results of comparison with other orders

(a) Randomly generated degree-reverse-lexicographic orders

Ranking: 8/100

Histogram:



(b) Randomly generated block orders in which each block has the same variables in the generated term order same as (a)

図 6: 実験結果-Katsura-8 (Risa/Asir)

Generated Term Order: $u_1 > u_2 > u_3 > u_4 > u_5 > u_6 > u_7 > u_0 > u_8$

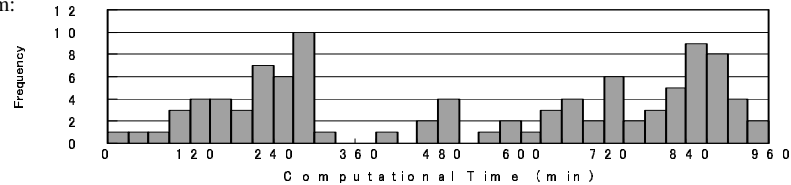
Computational Time: 137 min 21 sec

Results of comparison with other orders

(a) Randomly generated degree-reverse-lexicographic orders

Ranking: 8/100

Histogram:



(b) Randomly generated block orders in which each block has the same variables in the generated term order same as (a)

図 7: 実験結果-Katsura-8 (CoCoA)

る実験から得られた。CoCoA については、このようなデータを得る方法を見つけられなかったため、本論文では言及しない。

図 12 から図 15 に示す集計データは、各ブロックに含まれる変数はすべて同じという条件で得られたものである。したがって、これらの図は、ブロック内順序設定方法の効果のみを示しているということに注意されたい。ブロック設定方法が持つ効果の詳細については、全次数逆辞書式順序による計算結果が Heatpump と Robot で得られなかったため、確認できていない。

図 12 から図 15 により、本手法により設定された項順序では、係数膨張が抑えられていることが分かり、このことが計算効率の面で有利に働いていることが推測される。しかしながら、Katsura-8 (図 12) を除けば、計算時間と係数のビット長の和の最大値との間に強い相関関係は見られない。また、計算時間と簡約化が行われた多項式の数との間にも相関関係が見られないことから、各多項式に含まれる単項式の数や実際に行われた多倍長整数演算の回数など、より複雑な要因が計算効率に関係していることが考えられる。

Generated Term Order: $x_{13} > p_{278} > p_{249} > x_{11} > x_{12} > p_{223} > > p_{275} > p_{276} > p_{277} > p_{240} > p_{220} > > > x_{21} > x_{20} > x_{19} > p_{283} > x_{18} > x_{17} > x_{16} > x_{15} > p_{259} > x_{14}$

Computational Time: 1 sec

Results of comparison with other orders:

(a) Randomly generated degree-reverse-lexicographic orders

It takes more than 5 hours to compute a Groebner basis.

(b) Randomly generated block orders in which each block has the same variables in the generated term order

Ranking: 3/100

Histogram:

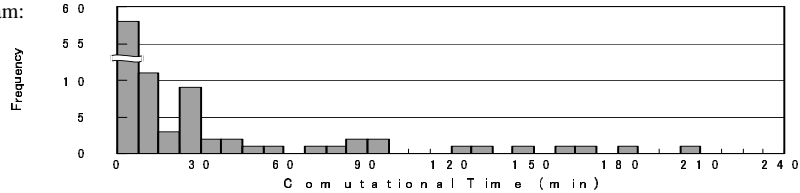


図 8: 実験結果-Heatpump (Risa/Asir)

Generated Term Order: $x_{13} > p_{278} > p_{249} > x_{11} > x_{12} > p_{223} > > p_{275} > p_{276} > p_{277} > p_{240} > p_{220} > > > x_{21} > x_{20} > x_{19} > p_{283} > x_{18} > x_{17} > x_{16} > x_{15} > p_{259} > x_{14}$

Computational Time: 8 sec

Results of comparison with other orders:

(a) Randomly generated degree-reverse-lexicographic orders

It takes more than 5 hours to compute a Groebner basis.

(b) Randomly generated block orders in which each block has the same variables in the generated term order

Ranking: 2/100

Histogram:

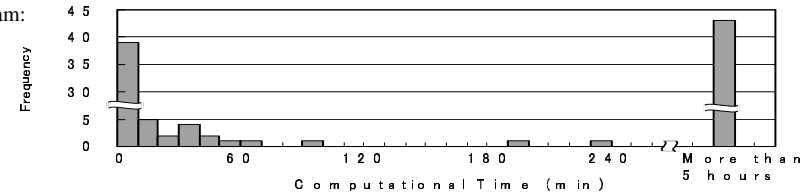


図 9: 実験結果-Heatpump (CoCoA)

5 おわりに

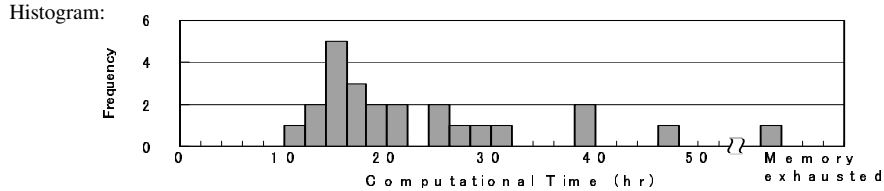
グレブナ基底計算を効率的に行うためのブロック式順序の自動設定方法を提案し、その有効性について検証した。本手法を4つの異なる例題に対して適用した結果、有意な差が認められ、本手法が効率的なグレブナ基底計算を実現する上で有効であることが確認された。

また、実験により以下のことが例証された。

1. 全次数逆辞書式順序を用いたグレブナ基底計算は必ずしも効率的ではない。

Heatpump (図8、図9) および Robot (図11) の結果が示しているように、全次数逆辞書式順序を用いた場合には多くの計算コストを要するような場合であっても、ブロック式順序で効率的に計算できる場合がある。Heatpump に関する結果はこれを端的に示しており、Risa/Asir、CoCoA とともに、全次数逆辞書式順序では5時間を越えても計算が終了しなかったが、ブロック式順序では数秒でグレブナ基底を求めることができている。

Generated Term Order: $a_5 > a_3 > a_2 > a_1$
 Computation Time: 12 h 2 min
 Results of comparison with other orders
 (a) Randomly generated degree-reverse-lexicographic orders
 Ranking: 2/24



(b) Randomly generated block orders in which each block has the same variables in the generated term order same as (a)

図 10: 実験結果-Mckay (Risa/Asir)

Generated Term Order: $x_{42} > x_{41} > x_{67} > x_{64} > x_{69} > x_{84} > x_{83} > x_{77} > x_{75} > x_{74} > x_{54} > x_{53} > x_{89} > x_{88} > x_{79} > x_{61} > x_{60} > x_{63} > x_{82} > x_{81} > x_{76} > x_{58} > x_{51} > x_{50} > x_{49} > x_{87} > x_{86} > x_{78} > x_{66} > x_{55} > x_{62} > x_{57} > x_{80} > x_{65} > x_{85} > x_{440} > p_{385} > x_{47} > x_{46} > x_{73} > x_{72} > x_{70} > x_{52} > x_{71} > x_{59} > x_{68} > x_{43} > x_{45} > x_{44}$
 Computation Time: 1 min 7 sec

Results of comparison with other orders
 (a) Randomly generated degree-reverse-lexicographic orders
 It takes more than 5 hours to compute a Groebner basis.
 (b) Randomly generated block orders in which each block has the same variables in the generated term order
 Ranking: 1/100

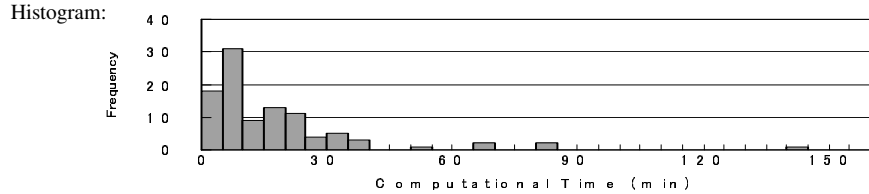


図 11: 実験結果-Robot (Risa/Asir)

2. ブロック内の変数順序も計算効率に大きくかわる。
 変数のブロックへの割り振りのみならず、各ブロック内での変数の順序関係もまた大きく計算効率に影響している。これは全次数逆辞書式順序を用いる場合でも同様であり、図 6 から図 11 のヒストグラムに示すとおりである。Heatpump (図 8、図 9) および Robot (図 11) ではこの影響が顕著であり、Heatpump の場合、速いものは数秒で計算が終了するのに対し、もっとも遅いものでは Risa/Asir で 3 時間半を要し、CoCoA では 5 時間を越えても計算が終了しない。また Robot では、各ブロックに含まれる変数は同じでありながら、Risa/Asir の場合、最も早いものと最も遅いもので百数十倍の計算速度の違いがある。CoCoA の場合、本手法で生成された項順序では 9 時間余りで計算が終了したが、それ以外の項順序では、2 日を越えても計算が終了しなかった (表 1)。

変数の数を n とした場合、ブロック式順序では $2^{n-1}n!$ 通り、全次数逆辞書式順序に限った場合でも $n!$ 通りの項順序が存在する。これらの中から、人間の経験と知識のみに基づいて効

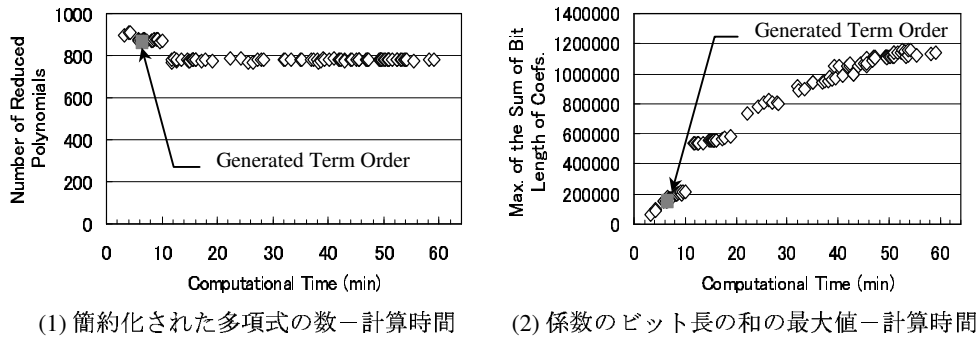


図 12: グレブナ基底計算における集計データと計算時間との相関関係-Katsura-8

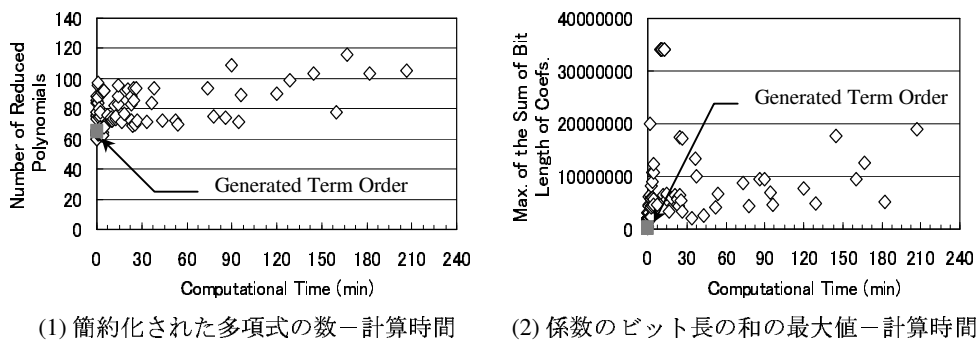
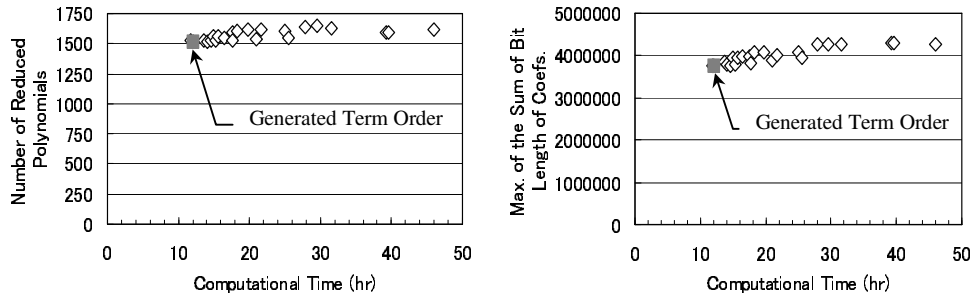


図 13: グレブナ基底計算における集計データと計算時間との相関関係-Heatpump

率的な項順序を決定することはきわめて困難である。また、複数の計算機を用いて様々な項順序におけるグレブナ基底計算を同時に行うことも 1 つの手段ではあるが、変数の数が多い場合には現実的ではない。

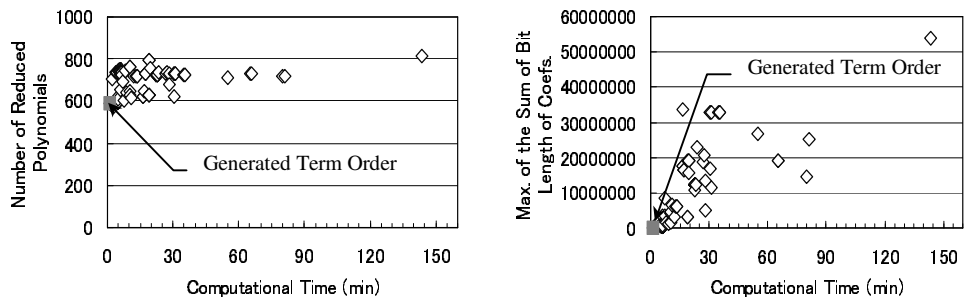
これに対して本手法は、わずかな計算コストで効率的な項順序を自動生成するものであり、実用上きわめて有効なものであると考えられる。特に、計算アルゴリズムや計算ノウハウに関する知識を持たないユーザでも効率的にグレブナ基底を求められるようにすることは、グレブナ基底の応用範囲を広げる上で大きな効果を持つものといえる。

本論文で提案した手法は、すべて経験則および実験的観測に基づいたものであり、理論的根拠は皆無に等しい。特に、ブロック内順序の設定方法は、著者が多くの実験データから傾向を掴んだものであり、理論的考察の結果得られたものではない。本手法の持つ効果に関しても、ブロック内順序の設定方法については係数膨張を抑える効果のあることが確認されたものの、ブロック設定方法の持つ効果はまだ分かっていない。今後、グレブナ基底計算の実装アルゴリズムへの依存性に関する考察も含め、本手法の有効性についての検証をさらに進める必要があるだろう。



(1) 簡約化された多項式の数—計算時間 (2) 係数のビット長の和の最大値—計算時間

図 14: グレブナ基底計算における集計データと計算時間との相関関係—Mckay



(1) 簡約化された多項式の数—計算時間 (2) 係数のビット長の和の最大値—計算時間

図 15: グレブナ基底計算における集計データと計算時間との相関関係—Robot

付記

本論文で提案している手法は、Risa/Asir プログラムとして実装され、フリーソフトウェアとして公開されている。Risa/Asir ホームページ (<http://risa.cs.ehime-u.ac.jp/>) よりリンクをたどることができる。

参 考 文 献

- [1] Capani, A., Niesi, G.: The CoCoA 3 Framework for a Family of Buchberger-like Algorithms, *Gröbner Bases and Applications* (Buchberger, B. and Winkler, F., eds.), London Mathematical Society Lecture Notes Series, Vol. 251, Cambridge University Press, 1998, 338–350.
- [2] Faugère, J. C., Gianni, P., Lazard, D., Mora, T.: Efficient Computation of Zero-dimensional Gröbner Bases by Change of Ordering, *J. Symb. Comp.*, **16**(4), 1993, 329–344.
- [3] 加藤靖: 情報数学, 情報処理入門シリーズ, 共立出版, 東京, 1988.

- [4] Katsura, S.: Theory of Spin Glass by the Method of the Distribution Function of an Effective Field, *Progress of Theoretical Physics*, Supplement, **87**, 1986, 139–154.
- [5] 野呂正行: “グレブナ基底—理論, 計算の効率化, 応用”, 数式処理における理論と応用の研究, 数理解析研究所講義録, **1138**, 京都大学数理解析研究所, 2000, 127–171.
- [6] Noro, M., McKay, J.: Computation of Replicable Functions on Risa/Asir, *Proc. PASC0'97*, ACM Press, 1997, 130–138.
- [7] Ponder, C. G.: Evaluation of “Performance Enhancements” in Algebraic Manipulation Systems, PhD Thesis, Computer Science Division, University of California, USA, 1988.
- [8] 齋藤友克, 竹島卓, 平野照比古: 日本で生まれた数式処理ソフト リサアジールガイドブック, SEG 出版, 東京, 1998.
- [9] Sawada, H.: Constraint-Based Computer Support for Insightful Multidisciplinary Engineering Design, PhD Thesis, University of Strathclyde, UK, 2001.

付録—実験で用いた多項式集合

1. katsura-8 [4] (9 variables and 9 polynomials)

Variables: $u_0, u_1, u_2, u_3, u_4, u_5, u_6, u_7, u_8$

$$u_0 + 2u_8 + 2u_7 + 2u_6 + 2u_5 + 2u_4 + 2u_3 + 2u_2 + 2u_1 - 1,$$

$$2u_7u_0 + 2u_1u_8 - u_7 + 2u_1u_6 + 2u_2u_5 + 2u_3u_4,$$

$$2u_6u_0 + 2u_2u_8 + 2u_1u_7 - u_6 + 2u_1u_5 + 2u_2u_4 + u_3^2,$$

$$2u_5u_0 + 2u_3u_8 + 2u_2u_7 + 2u_1u_6 - u_5 + 2u_1u_4 + 2u_2u_3,$$

$$2u_4u_0 + 2u_4u_8 + 2u_3u_7 + 2u_2u_6 + 2u_1u_5 - u_4 + 2u_1u_3 + u_2^2,$$

$$2u_3u_0 + 2u_5u_8 + 2u_4u_7 + 2u_3u_6 + 2u_2u_5 + 2u_1u_4 - u_3 + 2u_1u_2,$$

$$2u_2u_0 + 2u_6u_8 + 2u_5u_7 + 2u_4u_6 + 2u_3u_5 + 2u_2u_4 + 2u_1u_3 - u_2 + u_1^2,$$

$$2u_1u_0 + 2u_7u_8 + 2u_6u_7 + 2u_5u_6 + 2u_4u_5 + 2u_3u_4 + 2u_2u_3 + 2u_1u_2 - u_1,$$

$$u_0^2 - u_0 + 2u_8^2 + 2u_7^2 + 2u_6^2 + 2u_5^2 + 2u_4^2 + 2u_3^2 + 2u_2^2 + 2u_1^2.$$

2. Heatpump¹⁾ [9] (21 variables and 20 polynomials)

Variables: $p_{220}, p_{223}, p_{240}, p_{249}, p_{259}, p_{275}, p_{276}, p_{277}, p_{278}, p_{283}, x_{11}, x_{12}, x_{13}, x_{14}, x_{15},$

$x_{16}, x_{17}, x_{18}, x_{19}, x_{20}, x_{21}$

$$-99720p_{220} + 173p_{240} - 173p_{276} + 99720x_{14} + 22237560,$$

$$-p_{275}p_{240} + p_{275}p_{276} + x_{11},$$

$$(-50x_{13} + 97500000)p_{220}^6 + (169x_{14}^6 + 226122x_{14}^5 + 126063015x_{14}^4 + 37482736460x_{14}^3 +$$

$$6268987672935x_{14}^2 + 559193700425802x_{14} + 20783365865825641)x_{13} - 329550000x_{14}^6 -$$

$$440937900000x_{14}^5 - 245822879250000x_{14}^4 - 73091336097000000x_{14}^3 -$$

$$12224525962223250000x_{14}^2 - 1090427715830313900000x_{14} - 4052756343835999950000,$$

$$-3619p_{259}^2 - 970900p_{259} + 2500p_{277} + 34685000,$$

¹⁾<http://unit.aist.go.jp/digital-mfg/sawada/heatpump>

$$\begin{aligned}
&5066p_{259}^2 - 2314900p_{259} + 25x_{13} + 219025000, \\
&-p_{275}p_{240} + 15200p_{249} + p_{275}p_{277}, \\
&-15200p_{249} + 12600x_{20} + 12600x_{21} + 214200, \\
&-2p_{249} + 2p_{259} + x_{20} - x_{21} - 623, \\
&-1250x_{13} - 856154x_{14}^2 + 9373416x_{14} + 1849479034, \\
&5066p_{259}^2 - 2314900p_{259} + 25x_{13} + 219025000, \\
&2000p_{276} + 3627x_{14}^2 - 1044958x_{14} - 766118717, \\
&-1250x_{13} - 856154x_{14}^2 + 9373416x_{14} + 1849479034, \\
&-p_{275}p_{276} + p_{275}p_{277} + 14700p_{278}, \\
&73500p_{278} - 50400p_{283} + 50400x_{19} + 13759200, \\
&2p_{278} - p_{283} + 2x_{14} - x_{19} + 173, \\
&p_{223} - x_{11} - x_{12}, \\
&p_{259} - x_{15} - x_{21} - 320, \\
&-p_{259} + x_{16} - x_{20} + 303, \\
&-x_{14} - x_{17} + x_{19} + 50, \\
&p_{283} - x_{14} - x_{18} - 223.
\end{aligned}$$

3. Mckay [6] (4 variables and 20 polynomials)

Ref. <http://www-calfor.lip6.fr/~jcf/Benchs/@benchs/mckay.fgb>

4. Robot²) [9] (49 variables and 49 polynomials)

Variables: $p_{385}, p_{440}, x_{41}, x_{42}, x_{43}, x_{44}, x_{45}, x_{46}, x_{47}, x_{49}, x_{50}, x_{51}, x_{52}, x_{53}, x_{54}, x_{55}, x_{57},$
 $x_{58}, x_{59}, x_{60}, x_{61}, x_{62}, x_{63}, x_{64}, x_{65}, x_{66}, x_{67}, x_{68}, x_{69}, x_{70}, x_{71}, x_{72}, x_{73}, x_{74},$
 $x_{75}, x_{76}, x_{77}, x_{78}, x_{79}, x_{80}, x_{81}, x_{82}, x_{83}, x_{84}, x_{85}, x_{86}, x_{87}, x_{88}, x_{89}$

$$\begin{aligned}
&-9x_{47} + 31250000x_{44}^3 + 187500x_{45}x_{44}^2 + 2000x_{44} + 12x_{45}, \\
&125000000x_{44}^3 + (-750000x_{43} - 750000)x_{44}^2 + (3000x_{43} + 9500)x_{44} + 36x_{46} - 51x_{43} - 17, \\
&250000x_{44}^2 + (-1500x_{43} - 1000)x_{44} + 3x_{43} - 1152x_{49} - 15, \\
&12250000x_{44}^2 + (-73500x_{43} - 1323000)x_{44} + 3969x_{43} - 3200000000x_{50} + 1813, \\
&12250000x_{44}^2 + (-73500x_{43} - 49000)x_{44} + 1617x_{43} - 3200000000x_{51} + 833, \\
&-6250000000p_{440}x_{44}^4 + 500000000p_{440}x_{44}^3 + (-500000p_{440} - 5000000)x_{44}^2 + (22500x_{46} - \\
&2000p_{440} + 20000)x_{44} - 45x_{46} + 323p_{440} + 380, \\
&(-1000000p_{440} - 1250000)x_{44}^2 + (7500x_{43} + 4000p_{440} + 5000)x_{44} - 15x_{43} + 4p_{440} - \\
&10368x_{53} - 85, \\
&(-1000000p_{440} - 1250000)x_{44}^2 + (7500x_{43} + 4000p_{440} + 5000)x_{44} - 15x_{43} - 68p_{440} - \\
&10368x_{54} - 85, \\
&-6125000x_{44}^2 - 36750x_{45}x_{44} - 735x_{45} - 1600000000x_{55} - 392, \\
&62500x_{44}^2 + 375x_{45}x_{44} + 288x_{57} - 4, \\
&-6125000x_{44}^2 + (-36750x_{45} - 637000)x_{44} - 1911x_{45} - 1600000000x_{58} + 392, \\
&-1125x_{44}x_{47} - 3125000000p_{385}x_{44}^4 + (50000p_{385} - 250000)x_{44}^2 + 16p_{385} + 20,
\end{aligned}$$

²)<http://unit.aist.go.jp/digital-mfg/sawada/robot>

$$\begin{aligned}
 &(-250000p_{385} - 312500)x_{44}^2 - 1875x_{45}x_{44} + 2p_{385} + 2592x_{60} - 20, \\
 &(-250000p_{385} - 312500)x_{44}^2 - 1875x_{45}x_{44} - 16p_{385} + 2592x_{61} - 20, \\
 &6250000000x_{44}^4 - 500000000x_{44}^3 + 500000x_{44}^2 + 2000x_{44} + 81x_{46}^2 - 323, \\
 &-1000000x_{44}^2 + (4500x_{46} + 4000)x_{44} + (-27x_{43} - 9)x_{46} + 76, \\
 &-3p_{440}x_{46} - 5x_{43}, \\
 &81x_{47}^2 + 6250000000x_{44}^4 - 1000000x_{44}^2 - 320, \\
 &(4500x_{44} + 27x_{45})x_{47} + 1000000x_{44}^2 - 80, \\
 &3p_{385}x_{47} + 5x_{45}, \\
 &-200000p_{385}x_{44}^2 + 1500x_{45}x_{44} + 9x_{45}^2 + 16p_{385}, \\
 &(-18000000x_{44}^2 + (108000x_{43} + 90000)x_{44} - 324x_{43} - 108)x_{47} + (-18000000x_{46} - \\
 &16250000)x_{44}^2 + ((-108000x_{46} - 97500)x_{45} - 18000x_{46} + 97500x_{43} + 32500)x_{44} + \\
 &(-108x_{46} + 585x_{43} + 195)x_{45}, \\
 &(30600000000p_{440}x_{44}^3 - 1557000000p_{440}x_{44}^2 + (-24264000p_{440} + 24480000)x_{44} - \\
 &110160x_{46} + 116568p_{440} - 26640)x_{47} + (-348750000x_{46} - 12890000000)x_{44}^2 + \\
 &((-2092500x_{46} - 77340000)x_{45} + 26707500x_{46} - 41892500p_{440} - 12890000)x_{44} + \\
 &(160245x_{46} - 251355p_{440} - 77340)x_{45}, \\
 &-1000000p_{440}x_{44}^2 + (-7500x_{43} + 4000p_{440})x_{44} + 45x_{43}^2 + 15x_{43} + 76p_{440}, \\
 &(69750000x_{44}^2 + (-418500x_{43} + 5062500)x_{44} + 22032x_{46} - 31212x_{43} - 10404)x_{47} + \\
 &61200000000p_{385}x_{46}x_{44}^3 + (-55800000p_{385}x_{46} + 2578000000)x_{44}^2 + ((-5364000p_{385} + \\
 &4896000)x_{46} - 15468000x_{43} - 8378500p_{385} - 12890000)x_{44} + (-12852p_{385} - 4464)x_{46} + \\
 &(50271p_{385} + 46404)x_{43} + 16757p_{385} + 15468, \\
 &(-6750000000p_{440}x_{44}^3 + 11601000000p_{440}x_{44}^2 + (-57775500p_{440} - 540000)x_{44} + 4860x_{46} - \\
 &34884p_{440} - 41040)x_{47} + (6750000000p_{385}x_{46} + 2686000000000p_{440} - \\
 &26860000000000p_{385})x_{44}^3 + (11560500000p_{385}x_{46} - 13667000000p_{440} + 24490000000p_{385})x_{44}^2 + \\
 &((11452500p_{385} + 540000)x_{46} - 212984000p_{440} + 235420000p_{385})x_{44} + (-104085p_{385} - \\
 &42120)x_{46} + (1323803p_{385} + 1023208)p_{440} + 564060p_{385} - 37920, \\
 &12250000p_{440}x_{44}^2 + (-73500p_{440}x_{43} - 539000p_{440})x_{44} + 1617p_{440}x_{43} + 245p_{440} - \\
 &3200000000x_{74} - 1960, \\
 &6125000p_{385}x_{44}^2 + (36750p_{385}x_{45} + 245000p_{385})x_{44} + 735p_{385}x_{45} - 392p_{385} - 1600000000x_{75} - \\
 &980, \\
 &1000p_{440}x_{44} - 3p_{440}x_{43} - 2p_{440} - 20000x_{76}, \\
 &-250000p_{440}x_{44}^2 + (1500p_{440}x_{43} + 1000p_{440})x_{44} - 3p_{440}x_{43} + 15p_{440} - 1280000x_{77}, \\
 &1000p_{385}x_{44} + 3p_{385}x_{45} + 20000x_{78}, \\
 &-62500p_{385}x_{44}^2 - 375p_{385}x_{45}x_{44} + 4p_{385} + 320000x_{79}, \\
 &250000x_{44}^2 - 1000x_{44} - 648x_{80} - 1, \\
 &1000p_{440}x_{44} - 3p_{440}x_{43} - 2p_{440} - 5000x_{81}, \\
 &1000p_{440}x_{44} - 3p_{440}x_{43} - 2p_{440} - 20000x_{82}, \\
 &-250000p_{440}x_{44}^2 + (1500p_{440}x_{43} + 1000p_{440})x_{44} - 3p_{440}x_{43} + 15p_{440} - 320000x_{83}, \\
 &-250000p_{440}x_{44}^2 + (1500p_{440}x_{43} + 1000p_{440})x_{44} - 3p_{440}x_{43} + 15p_{440} - 1280000x_{84} - 40,
 \end{aligned}$$

$$\begin{aligned}
& 125000x_{44}^2 - 324x_{85} - 1, \\
& 1000p_{385}x_{44} + 3p_{385}x_{45} + 5000x_{86}, \\
& -1000p_{385}x_{44} - 3p_{385}x_{45} + 20000x_{87}, \\
& -62500p_{385}x_{44}^2 - 375p_{385}x_{45}x_{44} + 4p_{385} + 80000x_{88}, \\
& 62500p_{385}x_{44}^2 + 375p_{385}x_{45}x_{44} - 4p_{385} + 320000x_{89} - 190, \\
& -3x_{67}x_{47} - 40000000000x_{59}x_{44}^4 + ((-27000000000x_{73} + 24480000000x_{72})x_{46} + \\
& 1074400000000x_{73})x_{44}^3 + ((-46242000000x_{73} - 223200000x_{72})x_{46} - 8000000x_{60} - \\
& 8000000x_{61} - 98000000x_{75} - 2000000x_{79} - 2000000x_{88} + 2000000x_{89} - 9796000000x_{73} + \\
& 6400000x_{59} - 400000x_{68})x_{44}^2 + ((-588000x_{75} - 12000x_{79} - 12000x_{88} + 12000x_{89})x_{45} + \\
& (-45810000x_{73} - 21456000x_{72})x_{46} - 3920000x_{75} + 2000x_{78} + 2000x_{86} - 2000x_{87} - \\
& 941680000x_{73} - 33514000x_{72})x_{44} + (-11760x_{75} + 6x_{78} + 6x_{86} - 6x_{87})x_{45} + (416340x_{73} - \\
& 51408x_{72})x_{46} + 201084x_{72}x_{43} - 5295212x_{73}p_{440} - 4p_{385} + 64x_{60} - 512x_{61} + 6272x_{75} + \\
& 128x_{79} + 128x_{88} - 128x_{89} - 2256240x_{73} + 2048x_{59} + 67028x_{72} + 32x_{68}, \\
& ((-27000000000x_{73} + 122400000000x_{70})x_{44}^3 + (46404000000x_{73} - 6228000000x_{70})x_{44}^2 + \\
& (-231102000x_{73} - 97056000x_{70})x_{44} - 139536x_{73} + 466272x_{70})x_{47} - 200000000000x_{52}x_{44}^4 + \\
& (1074400000000x_{73} + 16000000000x_{52})x_{44}^3 + (-8000000x_{53} - 8000000x_{54} + 98000000x_{74} - \\
& 2000000x_{77} - 2000000x_{83} - 2000000x_{84} - 54668000000x_{73} - 16000000x_{52} - 2000000x_{71})x_{44}^2 + \\
& ((-588000x_{74} + 12000x_{77} + 12000x_{83} + 12000x_{84})x_{43} + 32000x_{53} + 32000x_{54} - 4312000x_{74} + \\
& 2000x_{76} + 8000x_{77} + 2000x_{81} + 2000x_{82} + 8000x_{83} + 8000x_{84} - 851936000x_{73} - 64000x_{52} - \\
& 167570000x_{70} + 8000x_{71})x_{44} - 1005420x_{70}x_{45} + 3x_{64}x_{46} + (12936x_{74} - 6x_{76} - 24x_{77} - \\
& 6x_{81} - 6x_{82} - 24x_{83} - 24x_{84})x_{43} + 4p_{440} + 32x_{53} - 544x_{54} + 5295212x_{73}p_{385} + 1960x_{74} - \\
& 4x_{76} + 120x_{77} - 4x_{81} - 4x_{82} + 120x_{83} + 120x_{84} + 4092832x_{73} + 10336x_{52} + 152x_{71}, \\
& ((5022000x_{72} + 2592000x_{69})x_{44} + 374544x_{72} - 7776x_{69})x_{47} + 36000000x_{42}x_{44}^2 + (-36000x_{49} - \\
& 1764000x_{50} - 1764000x_{51} + (-1764000x_{74} + 36000x_{77} + 36000x_{83} + 36000x_{84})p_{440} + \\
& 180000x_{53} + 180000x_{54} + 185616000x_{72} - 45000x_{71} + 2340000x_{69} - 144000x_{42} - 1000)x_{44} + \\
& 14040x_{69}x_{45} + 162x_{63}x_{46} + (540x_{71} + 6)x_{43} + 72x_{49} + 95256x_{50} + 38808x_{51} + (38808x_{74} - \\
& 18x_{76} - 72x_{77} - 18x_{81} - 18x_{82} - 72x_{83} - 72x_{84})p_{440} - 360x_{53} - 360x_{54} - 603252x_{72}p_{385} - \\
& 556848x_{72} + 90x_{71} + 15x_{64} + 2448x_{42} + 2, \\
& 162x_{66}x_{47} + 36000000x_{41}x_{44}^2 + ((25110000x_{70} + 2592000x_{69})x_{46} + 1764000x_{55} + 36000x_{57} + \\
& 1764000x_{58} + (-1764000x_{75} - 36000x_{79} - 36000x_{88} + 36000x_{89})p_{385} - 180000x_{60} - \\
& 180000x_{61} + 9000x_{68} + 928080000x_{70} + 2340000x_{69} - 1000)x_{44} + (108x_{68} - 6)x_{45} + \\
& (-1922940x_{70} + 2592x_{69})x_{46} - 14040x_{69}x_{43} + 3016260x_{70}p_{440} + 35280x_{55} + 91728x_{58} + \\
& (-35280x_{75} + 18x_{78} + 18x_{86} - 18x_{87})p_{385} - 15x_{67} + 928080x_{70} - 4680x_{69} + 2304x_{41}, \\
& (5184x_{65} + 1)x_{47} + (-27000000000x_{73} + 122400000000x_{70})p_{440}x_{44}^3 + ((46404000000x_{73} - \\
& 62280000000x_{70})p_{440} - 279000000x_{72} - 144000000x_{69})x_{44}^2 + ((1674000x_{72} + \\
& 864000x_{69})x_{43} + (-231102000x_{73} - 97056000x_{70})p_{440} - 2160000x_{73} + 144000x_{59} - \\
& 20250000x_{72} + 97920000x_{70} + 720000x_{69} - 9000x_{66})x_{44} - 54x_{66}x_{45} + (19440x_{73} - \\
& 88128x_{72} - 440640x_{70})x_{46} + (124848x_{72} - 2592x_{69})x_{43} + (-139536x_{73} + 466272x_{70})p_{440} + \\
& 3x_{67}p_{385} - 164160x_{73} + 41616x_{72} - 106560x_{70} - 864x_{69} + 576x_{41},
\end{aligned}$$

$$\begin{aligned}
 & (-19440x_{73} + 88128x_{72} + 440640x_{70})x_{47} + (-27000000000x_{73} + \\
 & 244800000000x_{72})p_{385}x_{44}^3 + ((-46242000000x_{73} - 223200000x_{72})p_{385} + 1395000000x_{70} + \\
 & 144000000x_{69})x_{44}^2 + ((8370000x_{70} + 864000x_{69})x_{45} + (-45810000x_{73} - 21456000x_{72})p_{385} - \\
 & 2160000x_{73} + 19584000x_{72} - 720000x_{52} - 106830000x_{70} + 144000x_{69} + 9000x_{63})x_{44} + \\
 & (-640980x_{70} + 864x_{69})x_{45} + (-5184x_{62} - 1)x_{46} - 54x_{63}x_{43} - 3x_{64}p_{440} + (416340x_{73} - \\
 & 51408x_{72})p_{385} + 168480x_{73} - 17856x_{72} + 1440x_{52} - 18x_{63} + 576x_{42}, \\
 & ((3645000000x_{73} - 165240000000x_{70})p_{440}x_{44}^2 + ((-4176360000x_{73} + 560520000x_{70})p_{440} + \\
 & 25110000x_{72} + 12960000x_{69})x_{44} + (-75330x_{72} - 38880x_{69})x_{43} + (10399590x_{73} + \\
 & 4367520x_{70})p_{440} + 97200x_{73} - 6480x_{59} + 911250x_{72} - 4406400x_{70} - 32400x_{69} + 405x_{66})x_{47} + \\
 & (36000000000x_{52}p_{440} - 72000000000x_{59}p_{385} - 360000000000x_{65} - 360000000000x_{62})x_{44}^3 + \\
 & ((-3645000000x_{73} + 33048000000x_{72})p_{385}x_{46} + (-1450440000000x_{73} - \\
 & 2160000000x_{52})p_{440} + 1450440000000x_{73}p_{385} + 2700000000x_{42} + 2700000000x_{41} + \\
 & 2160000000x_{62})x_{44}^2 + (1080000x_{41}x_{45} + ((-4161780000x_{73} - 20088000x_{72})p_{385} + \\
 & 125550000x_{70} + 12960000x_{69})x_{46} - 1080000x_{42}x_{43} - 180000x_{49} - 8820000x_{50} - \\
 & 8820000x_{51} + (720000x_{53} + 720000x_{54} - 8820000x_{74} + 180000x_{77} + 180000x_{83} + 180000x_{84} + \\
 & 4920120000x_{73} + 1440000x_{52} + 180000x_{71})p_{440} + 900000x_{53} + 900000x_{54} + 8820000x_{55} + \\
 & 180000x_{57} + 8820000x_{58} + (-720000x_{60} - 720000x_{61} - 8820000x_{75} - 180000x_{79} - \\
 & 180000x_{88} + 180000x_{89} - 881640000x_{73} + 576000x_{59} - 36000x_{68})p_{385} - 900000x_{60} - \\
 & 900000x_{61} - 180000x_{80} - 180000x_{85} - 2880000x_{59} + 928080000x_{72} + 14400000x_{52} + \\
 & 4640400000x_{70} + 11700000x_{69} - 180000x_{63} - 1080000x_{42} + 180000x_{66} + 2880000x_{65} - \\
 & 1440000x_{62} - 10018)x_{44} + ((376650x_{70} + 38880x_{69})x_{46} + 26460x_{55} + 540x_{57} + 26460x_{58} + \\
 & (-26460x_{75} - 540x_{79} - 540x_{88} + 540x_{89})p_{385} - 2700x_{60} - 2700x_{61} + 135x_{68} + 13921200x_{70} + \\
 & 35100x_{69} - 15)x_{45} + ((-2061450x_{73} - 965520x_{72})p_{385} - 97200x_{73} + 881280x_{72} - 32400x_{52} - \\
 & 4807350x_{70} + 6480x_{69} + 405x_{63})x_{46} + (540x_{49} + 26460x_{50} + 26460x_{51} + (26460x_{74} - \\
 & 540x_{77} - 540x_{83} - 540x_{84})p_{440} - 2700x_{53} - 2700x_{54} - 2784240x_{72} + 675x_{71} - 35100x_{69} + \\
 & 2160x_{42} + 15)x_{43} + 360x_{49} + 476280x_{50} + 17640x_{51} + (-1440x_{53} - 1440x_{54} + 194040x_{74} - \\
 & 90x_{76} - 360x_{77} - 90x_{81} - 90x_{82} - 360x_{83} - 360x_{84} + 38337120x_{73} + 2880x_{52} + 7540650x_{70} - \\
 & 360x_{71})p_{440} - 1800x_{53} - 1800x_{54} + 458640x_{58} + (-176400x_{75} + 90x_{78} + 90x_{86} - 90x_{87} - \\
 & 42375600x_{73} - 1508130x_{72})p_{385} + 360x_{80} - 2320200x_{72} - 28800x_{52} + 2320200x_{70} - \\
 & 11700x_{69} + 360x_{63} + 6840x_{42} + 5760x_{41} - 2880x_{62} + 10.
 \end{aligned}$$