Automatic Generation of Ranking of Variables for Efficient Computation of Gröbner Bases in Engineering Applications

Hiroyuki Sawada
E-mail: h.sawada@aist.go.jp

National Institute of Advanced Industrial Science and Technology
1-2-1 Namiki, Tsukuba, Ibaraki 305-8564, Japan

Abstract. It is well known that a term order has a great influence on computational efficiency of Gröbner bases. This paper proposes a new method of determining a term order that enables efficient computation of Gröbner bases. The advantages of this method have been shown through experiments. Furthermore, since it helps users without sufficient knowledge about the computational algorithms in computing Gröbner bases efficiently, it can create a new application area of Gröbner bases including engineering.

1 Introduction

There have been many attempts to formalize practical problems, including engineering design problems, as constraint satisfaction problems, and to solve them by applying generic constraint solving methods. According to this approach, the authors have developed innovative constraint solving methods based on techniques of Gröbner Basis and integrated them into our engineering design support system *DeCoSolver (Design Constraint Solver)* [8–10].

Those constraint solving methods have been proven to be effective through case studies of multidisciplinary engineering design: a robotic arm system design and a heat pump system design. On the other hand, their limitations in computational efficiency have also been disclosed. Since the most time consuming part of those constraint solving methods is the Gröbner bases computation, in order to apply them to practical use in industry, it is necessary to improve efficiency of computing Gröbner bases.

It is well known that a term order has a great influence on computational efficiency of Gröbner bases. Though the degree reverse lexicographic order (DRL) has been widely used as an *efficient* term order, it is not always good for computational efficiency.

Pistone *et al.* [6] have chosen a preferable term order based on engineering meanings of each variable in their engine design problem. Rust [7] has studied relationships between ranking of variables and solvability of partial differential algebraic equations. Though they give some useful guidelines to determine a term order, it cannot be determined automatically. Since people in the area of engineering do not have sufficient knowledge about Gröbner bases, its computational algorithms and efficiency, it is quite difficult for them to decide a term order according to some guidelines for themselves.

The objective of this research is to establish a method of determining a term order automatically that makes it possible to compute Gröbner bases efficiently. The structure of this paper is as follows. Section 2 describes our new method of determining a term order. In Section 3, the procedure of computing Gröbner bases is explained. Section 4 shows the results of different experiments, which illustrate the great advantages of our method. Since we focus on engineering applications, we deal with Gröbner bases over infinite fields.

2 Determination of Term Order

We give a few definitions [1] necessary to describe our new method.

Definition 1. *(Term Order)*
Let $T(x_1, \ldots, x_n)$ be a set of all terms in variables $x_1, \ldots, x_n$. A term order, notated by $>$, is a linear order on T that satisfies the following conditions.

1. *$t > 1$ for all $t \in T$.*
2. *$t_1 > t_2$ implies $t_1 \cdot s > t_2 \cdot s$ for all $s, t_1, t_2 \in T$.*

Definition 2. *(Block Order)*
Let $T(x_1, \ldots, x_n)$ be a set of all terms in variables $x_1, \ldots, x_n$. Let $1 \leq i < n$, and set

$$T_1 = T(x_1, \ldots, x_i), \text{ and } T_2 = T(x_{i+1}, \ldots, x_n).$$

Let $>_1$ and $>_2$ be term orders on T_1 and T_2 respectively. Any $t \in T$ may be written uniquely as $t = t_1 t_2$ with $t_i \in T_i$ for $i = 1, 2$. A block order $>$ on T is defined by $s > t$ iff

$$s_1 >_1 t_1, \text{ or } s_1 = t_1 \text{ and } s_2 >_2 t_2.$$

The above definition of block order can be easily extended to a block order with more than two blocks. Our method generates a block order where DRL is used in each block.

Notation 1 *(Block Order)*
"$x_1 > \ldots > x_i >> x_{i+1} > \ldots > x_n$" means that $x_1, \ldots, x_i$ and $x_{i+1}, \ldots, x_n$ belong to the upper and lower block respectively.

It is well known that to compute Gröbner bases efficiently, dependent variables should be put in the upper block. Based on this experiential fact, our method determines a term order in four steps: reduction by bivariate linear polynomials, determination of blocks, determination of ranking of variables in each block, and rearrangement of blocks.

2.1 Preprocess – Reduction by Bivariate Linear Polynomials

Suppose there is a bivariate linear polynomial.

$$ax + by + c \ (a, b \text{ and } c \text{ are constants and } x > y). \tag{1}$$

Since reduction by equation (1) substitutes $-(b/a)y - (c/a)$ for x, the rank of x has no influence on computational efficiency afterwards. To remove such variables, inter-reduction by bivariate linear polynomials is conducted one by one. The given polynomial set is divided into two polynomial subsets: one is a set of bivariate linear polynomials used in the inter-reduction process, the other is a set of polynomials obtained as results of the inter-reductions.

2.2 Determination of Blocks

If there is a polynomial that has a variable appearing in a linear term only once and nowhere else in the polynomial, it would be better to put the variable in the upper block. This can be extended to the following heuristics.

Heuristics 1 *If there is a polynomial that has some variables appearing in linear terms only once and nowhere else in the polynomial, it would be better to put such variables in the upper block and the others in the lower.*

As a supplement, the following heuristics can be obtained.

Heuristics 2 *If there is a polynomial that has no variable appearing in a linear term only once and nowhere else in the polynomial, all the variables of the polynomial should be put in the same block.*

First of all, our method divides a polynomial set F into two polynomial subsets of P and Q corresponding to Heuristics 1 and 2 respectively.

$$\begin{aligned} &F = \{f_1(x_1, \ldots, x_n), \ldots, f_m(x_1, \ldots, x_n)\}, \\ &P = \{f \in F | \exists i \; x_i \text{ appears in a linear term only once and nowhere else in } f\}, \\ &Q = F - P. \end{aligned} \tag{2}$$

Next, for each polynomial in P, a directed graph that shows the upper-lower relationships between variables is constructed. Our method combines these partial directed graphs into a global graph, and determines blocks based on it. Blocks are determined by three steps: constructing partial and global directed graphs, converting each closed path into one node, and determining the order of blocks and variables in each block.

Constructing Partial and Global Directed Graphs Fig. 1 shows an example of constructing the partial and global directed graphs from P, the subset of the given polynomial set F in equation (3).

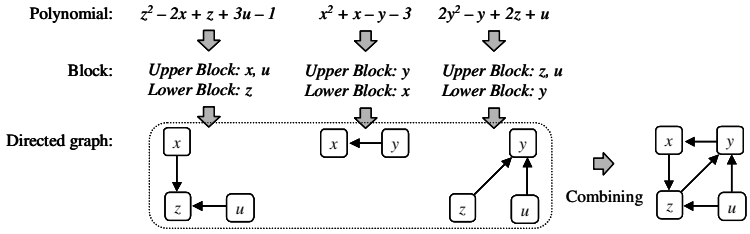


Fig. 1. Construction of directed graphs

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

When the global directed graph has closed paths, variables on the same closed path are put in the same block. Though it seems proper, there is a case where it is not appropriate for a closed path consisting of two nodes.

Example 1. Let equation (4) be a given polynomial set.

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

Fig. 2 shows the global directed graph for equation (4). In this case, if x and y are put in the upper

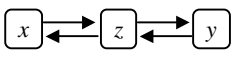


Fig. 2. Closed paths consisting of two nodes

block and z is put in the lower, the Gröbner basis is obtained immediately as equation (5).

$$x - f(z), y - g(z), z - h(f(z), g(z)). \tag{5}$$

Thus, there is a case where removing one of directed edges from a closed path consisting of two nodes is considered to improve computational efficiency. In order to construct a global directed graph, we have introduced a *Control Parameter of Removing Directed Edges* C_{rde} as below.

Control Parameter of Removing Directed Edges C_{rde}

Suppose there is a closed path consisting of two nodes of X and Y. If equation (6) is valid, the directed edge from X to Y is removed.

$$\frac{(\text{Number of directed edges to } X)}{(\text{Number of directed edges to } Y)} > C_{rde}. \tag{6}$$

In Fig. 2, the number of directed edges to each node is given as below.

	$\{x\}$	$\{y\}$	$\{z\}$
Number of directed edges to each node	1	1	2

Therefore, if C_{rde} is defined as 1, directed edges from z to x and y are removed, which leads to good block order for computational efficiency.

Converting Each Closed Path into One Node If the global directed graph has a closed path, all the variables on the same closed path should be put in one node (Fig. 3).

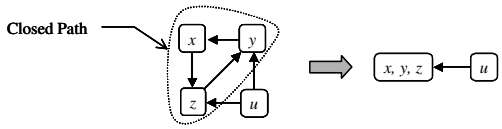


Fig. 3. Conversion of a closed path into one node

Determining the Order of Blocks and Variables in Each Block Since the global directed graph has no closed path, it has starting and terminal nodes. Firstly, the maximum distance between the starting and the terminal nodes is calculated to determine the number and the order of blocks. Then, each variable node is assigned to one of blocks so that the upper-lower relations represented by the global directed graph are kept.

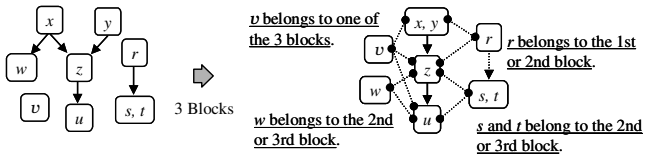


Fig. 4. Determining the order of blocks and variables in each block

Fig. 4 shows an example. In this case, though $\{x\}$, $\{y\}$, $\{z\}$ and $\{u\}$ are assigned to appropriate blocks uniquely, $\{v\}$, $\{w\}$, $\{r\}$ and $\{s,t\}$ cannot be assigned uniquely. Appropriate blocks for these nodes are determined by applying Heuristics 2 to a polynomial set Q defined in equation (2).

1. For each node, blocks to which it may be assigned are determined.
2. If all the variable nodes can be assigned uniquely, they are assigned to appropriate blocks and the computation is finished.

3. Otherwise, one of variable nodes, N, that might be assigned to the highest block is selected. Let V_N be a set of variables included in N.
4. From the polynomial set Q, all the polynomials that have variables included in V_N are selected. Let V_Q be a set of variables that the selected polynomials contain. According to Heuristics 2, the variables of node N should be put in a block that has variables of V_Q.
5. For each block B, percentage of variables included in V_Q is calculated.

$$\text{(Block } B\text{'s percentage of variables included in } V_Q) = \frac{\text{(Number of variables included in both of block } B \text{ and } V_Q)}{\text{(Number of variables included in block } B)}.$$

6. The block whose percentage of variables included in V_Q is the maximum is decided as the appropriate block for N. If there are plural appropriate blocks, the highest block should be selected.
7. Go to step 1.

2.3 Determination of Ranking of Variables in Each Block

Currently, we have no well-known heuristics to determine ranking of variables in a block. The author has conducted a lot of experiments, and found that when variable ranking is determined according to the following procedure, memory consumption is reduced and Gröbner bases can be computed efficiently.

1. For each variable, the maximum degree in the given polynomial set F is calculated. A variable of small maximum degree should be greater than that of large maximum degree. If there are plural variables of the same maximum degree, the order is decided by the way of step 2.
2. For each variable, all the terms of each polynomial in F are classified by the degree of the variable. Then, products of the degree and the number of terms in the corresponding group are summed. A variable of the large summation should be greater than that of the small summation.

For example, in the case of Katsura-2 [3] shown by equation (7), numerical values used to determine the ranking are obtained as below.

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

	u_0	u_1	u_2
Maximum degree	2	2	2
Number of terms of degree 1	3	4	2
Number of terms of degree 2	1	1	1
Sum of (degree × number of terms)	$1 \times 3 + 2 \times 1 = 5$	$1 \times 4 + 2 \times 1 = 6$	$1 \times 2 + 2 \times 1 = 4$

Thus, the ranking is determined as $u_1 > u_0 > u_2$.

2.4 Postprocess – Rearrangement of Blocks

In Section 2.2, blocks have been determined principally based on Heuristics 1. However, if there are many polynomials corresponding to Heuristics 2, it is reasonable to put the variables in the same block.

As a post process, our method combines adjacent blocks into one block if necessary. In order to decide whether adjacent blocks should be combined, we have introduced a *Control Parameter of Combining Blocks* C_{cb} as below.

Control Parameter of Combining Blocks C_{cb}

Let $B_1 >> \ldots >> B_j$ be adjacent blocks determined by the procedure of Section 2.2. If equation (8) is valid, these blocks are combined into one block.

$$\frac{\text{Number of polynomials that disagree to } B_1 >> \ldots >> B_j}{\text{Number of polynomials that agree to } B_1 >> \ldots >> B_j} > C_{cb}. \tag{8}$$

3 Computation of Gröbner Bases

Gröbner bases can be computed by the following steps.

1. Let Φ be the given polynomial set. Let H be a set of bivariate linear polynomials obtained by the preprocess (Section 2.1), and F be a set of the others. Let $x_1, \ldots, x_n$ be variables included in F, and $y_1, \ldots, y_\gamma$ be variables that are included in H and excluded from F.
2. The term order T_F for F is determined.
3. The Gröber basis G of F is computed by the term order T_F.
4. $G \cup H$ is a Gröbner basis of Φ of the term order $y_1 > \ldots > y_\gamma >> T_F$.

   Proof. Since all the polynomials in F have been reduced by bivariate linear polynomials in H, head terms of polynomials in H are $y_1, \ldots, y_\gamma$. Thus, head terms of polynomials in H are prime to head terms in G. In addition, since inter-reduction by bivariate linear polynomials are conducted one by one (Section 2.1), every polynomial in H has a different head term from the others. Therefore, all the S-polynomials generated by $G \cup H$ reduce to zero modulo $G \cup H$. It is obvious that $Ideal(\Phi) = Ideal(G \cup H)$, where $Ideal(\Phi)$ is an ideal generated by Φ. Thus, $G \cup H$ is a Gröbner basis of Φ of the term order $y_1 > \ldots > y_\gamma >> T_F$.

4 Experiment

The following polynomial sets have been used to evaluate our method. The notation "$x(d)$" means that the maximum degree of variable x is d.

1. Katsura-8 [3] (9 variables and 9 polynomials)
   Variables (maximum degrees): $u_0(2), u_1(2), u_2(2), u_3(2), u_4(2), u_5(2), u_6(2), u_7(2), u_8(2)$
2. Heatpump[1] (21 variables and 20 polynomials)
   Variables (maximum degrees): $p_{220}(6), p_{223}(1), p_{240}(1), p_{249}(1), p_{259}(2), p_{275}(1), p_{276}(1), p_{277}(1), p_{278}(1), p_{283}(1), x_{11}(1), x_{12}(1), x_{13}(1), x_{14}(6), x_{15}(1), x_{16}(1), x_{17}(1), x_{18}(1), x_{19}(1), x_{20}(1), x_{21}(1)$
   This polynomial set is derived from a heat pump system design problem (ref. Appendix) [8].
3. Mckay[2] [4] (4 variables and 20 polynomials)
   Variables (maximum degrees): $a_1(17), a_2(10), a_3(9), a_5(6)$
4. Robot[3] [8] (49 variables and 49 polynomials)
   Variables (maximum degrees): $p_{385}(1), p_{440}(1), x_{41}(1), x_{42}(1), x_{43}(2), x_{44}(4), x_{45}(2), x_{46}(2), x_{47}(2), x_{49}(1), x_{50}(1), x_{51}(1), x_{52}(1), x_{53}(1), x_{54}(1), x_{55}(1), x_{57}(1), x_{58}(1), x_{59}(1), x_{60}(1), x_{61}(1), x_{62}(1), x_{63}(1), x_{64}(1), x_{65}(1), x_{66}(1), x_{67}(1), x_{68}(1), x_{69}(1), x_{70}(1), x_{71}(1), x_{72}(1), x_{73}(1), x_{74}(1), x_{75}(1), x_{76}(1), x_{77}(1), x_{78}(1), x_{79}(1), x_{80}(1), x_{81}(1), x_{82}(1), x_{83}(1), x_{84}(1), x_{85}(1), x_{86}(1), x_{87}(1), x_{88}(1), x_{89}(1)$
   This polynomial set is derived from a robotic arm system design problem (ref. Appendix) [8].

The term order determined by our method has been compared with two different kinds of term order.

Comparison-1 *Randomly generated DRL order*
This comparison discloses the effectiveness of determining blocks.
Comparison-2 *Randomly generated block order in which each block has the same variables in the determined term order*
This comparison discloses the effectiveness of determining ranking of variables in each block.

C_{rde} and C_{cb} have been assigned the following numerical values.

Control Parameter of Removing Directed Edges C_{rde}: 1
Control Parameter of Combining Blocks C_{cb}: 1

The following hardware and software have been used for this experiment.

[1] `http://unit.aist.go.jp/digital-mfg/staff/sawada/robot`
[2] `http://www-calfor.lip6.fr/~jcf/Benchs/@benchs/mckay.fgb`
[3] `http://unit.aist.go.jp/digital-mfg/staff/sawada/heatpump`

Table 1. Results of Experiments

Problems	Software	Timing	Comparison-1	Comparison-2
Katsura-8	Risa/Asir	6 min 32 sec	8/100 (∗1)	8/100 (∗1)
	CoCoA	137 min 21 sec	8/100 (∗1)	8/100 (∗1)
Heatpump	Risa/Asir	1 sec	– (∗2)	3/100 (∗1)
	CoCoA	8 sec	– (∗2)	2/100 (∗1)
Mckay	Risa/Asir	12 h 2 min	2/24 (∗1)	2/24 (∗1)
	CoCoA	more than 5 days	– (∗2)	– (∗2)
Robot	Risa/Asir	1 min 7 sec	– (∗2)	1/100 (∗1)
	CoCoA	9 h 17 min	– (∗2)	– (∗2)

∗1 "a/b" means that the generated term order is ath fastest among b choices.
∗2 Gröbner bases could not be computed by the other term orders.

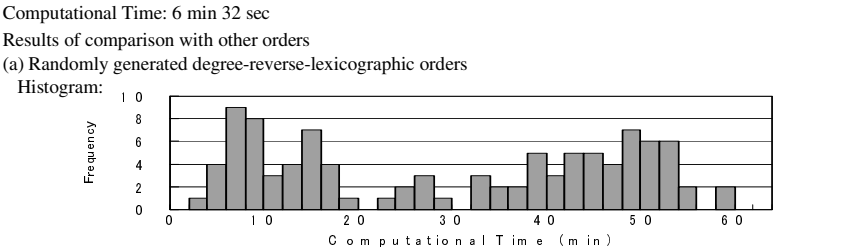


Fig. 5. Result of experiment – Katsura-8 (Risa/Asir)

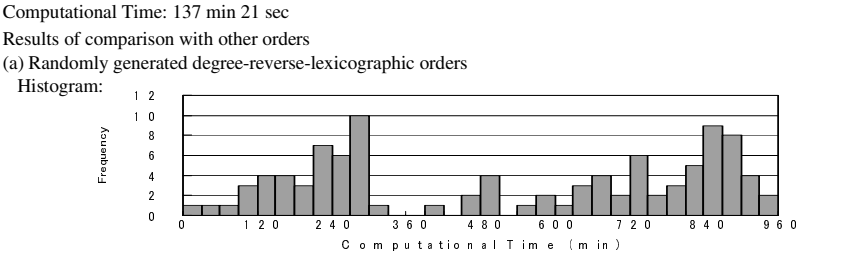


Fig. 6. Result of experiment – Katsura-8 (CoCoA)

Hardware: Pentium-4 2GHz CPU and 2 GB RAM
Software: Risa/Asir[4] [5] and CoCoA[5] [2] on Windows 2000

Table 1 and Figs. 5 to 10 show the results of the experiments. For all the benchmarks, our method took a few seconds to determine the term order.

These figures show that the term order determined by our method has a good effect on computational efficiency of Gröbner bases.

5 Conclusion

In this paper, we have proposed a new method of determining the term order that makes it possible to compute Gröbner bases efficiently, and examined its effectiveness. The benchmark test involving four completely different polynomial sets have proven the effectiveness of our method.

[4] `http://www.math.kobe-u.ac.jp/Asir/asir.html`
[5] `http://cocoa.dima.unige.it/`

Computational Time: 1 sec
Results of comparison with other orders:
(a) Randomly generated degree-reverse-lexicographic orders
Groebner bases could not be computed according to the other term orders.
(b) Randomly generated block orders in which each block has the same variables in the generated term order
Histogram:

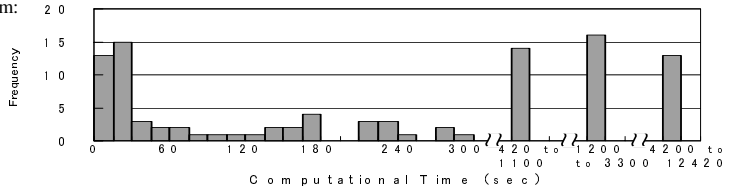


Fig. 7. Result ot experiment – Heatpump (Risa/Asir)

Computational Time: 8 sec
Results of comparison with other orders:
(a) Randomly generated degree-reverse-lexicographic orders
Groebner bases could not be computed according to the other term orders.
(b) Randomly generated block orders in which each block has the same variables in the generated term order
Histogram:

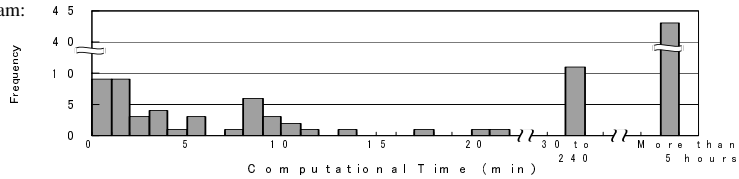
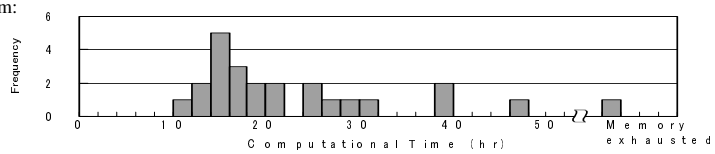


Fig. 8. Result ot experiment – Heatpump (CoCoA)

Computation Time: 12 h 2 min
Results of comparison with other orders
(a) Randomly generated degree-reverse-lexicographic orders
Histogram:



(b) Randomly generated block orders in which each block has the same variables in the generated term order
Same as (a), histogram omitted.

Fig. 9. Result of experiment – Mckay (Risa/Asir)

Since our method can determine an efficient term order automatically with small computational cost, it is considered effective for practical use. Especially, our method helps users without sufficient knowledge about the computational algorithms in computing Gröbner bases efficiently. This means that our method has a possibility of creating a new application area of Gröbner bases in addition to engineering. On the other hand, our method is based on heuristics and experiments, and there are many unknown factors. For example, the effect of two control parameters C_{rde} and C_{cb} on computational efficiency is unclear. In addtion, it is unclear how and why the procedure given in Section 2.3 can reduce memory consumption. Therefore, further study and research is necessary.

Our method has some limitations. Firstly, it does not work on a problem symmetric with respect to the variables. When we computed Gröbner bases of a symmetric problem, which is a subset of Cyclic-7, in 100 random DRLs, the fastest was 33 seconds while the slowest was 114 seconds. Our method cannot select the good ranking for such kind of symmetric problems though the computational efficiency should depend on ranking of variables. Secondly, our method does not check whether the given polynomial set is already a Gröbner basis according to some term order. Actually, when we applied our method to the Gröbner bases computed in the experiments in Section 4, different term orders were obtained. We also recalculated Gröbner bases from each obtained Gröbner basis by the original term order and the newly obtained one. In these cases,

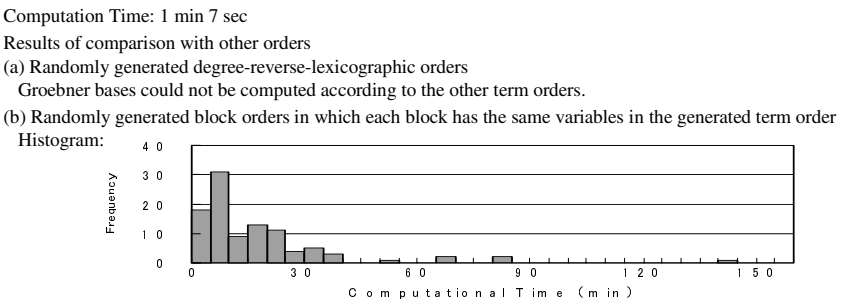


Fig. 10. Result of experiment – Robot (Risa/Asir)

there was no significant difference in computational time between these two different term orders. However, it cannot be extended to general cases.

Our method is now implemented as a Risa/Asir program. It is available from our Web page[6] free of charge.

References

1. T. Becker and V. Weispfenning. *Gröbner Bases.* Springer–Verlag, 1993.
2. A. Capani and G. Niesi. The CoCoA 3 Framework for a Family of Buchberger-like Algorithms. *Gröbner Bases and Applications*, 251:338–350, 1998.
3. S. Katsura. Theory of Spin Glass by the Method of the Distribution Function of an Effective Field. *Progress of Theoretical Physics*, (87):139–154, 1986.
4. M. Noro and J. Mckay. Computation of Replicable Functions on Risa/Asir. In *Proceedings of PASCO'97*, pages 130–138. ACM, 1997.
5. M. Noro and T. Takeshima. Risa/Asir – a Computer Algebra System. In *Proceedings of ISSAC'91*, pages 387–396. ACM, 1992.
6. G. Pistone, E. Riccomagno, and H. P. Wynn. Gröbner Basis Methods for Structuring and Analyzing Complex Industrial Experiments. In *Proceedings of First International Symposium on Industrial Statistics*, 1999.
7. C. J. Rust. *Ranking of Derivatives for Elimination Algorithms and Formal Solvability of Analytic Partial Differential Equations.* PhD Thesis, University of Chicago, 1998.
8. H. Sawada and X.-T. Yan. Applying Generic Constraint Solving Techniques in Providing Insights into Engineering Design. In *Proceedings of ICED 01*, Design Methods for Performance and Sustainability, pages 123–130, 2001.
9. H. Sawada and X.-T. Yan. Application of Gröbner Basis and Quantifier Elimination in Engineering Design: An Introduction for Engineers. In *Proceedings of ASCM 2001*, pages 141–150, 2001.
10. H. Sawada and X.-T. Yan. Computer Support for Insightful Engineering Design based on Generic and Rigorous Principles of Symbolic Algebra. In *Recent Advances in Integrated Design and Manufacturing in Mechanical Engineering*, G. Gogu *et al.* eds., pages 13–22, Kluwer Academic Publishers, 2003.

Appendix. Engineering Design Examples

1. A Heat Pump System Design Problem

Fig. 11 shows the heat pump system. It consists of a compressor, a condenser, an expansion valve and an evaporator. The refrigerant evaporates at the evaporator and collects heat from the water. Then, it is adiabatically compressed by the compressor. In the condenser, it condenses and release heat to heat up the water. After that, it goes through the expansion valve. The design task is to determine the following design parameter values: condensation and evaporation temperature Tc and Te, heat transfer area of the condenser and evaporator Ac and Ae, discharging pressure and compression ratio of the compressor Pd and κ, and mass flow rate of the refrigerant Qr.

[6] `http://staff.aist.go.jp/h.sawada/termorder.html`

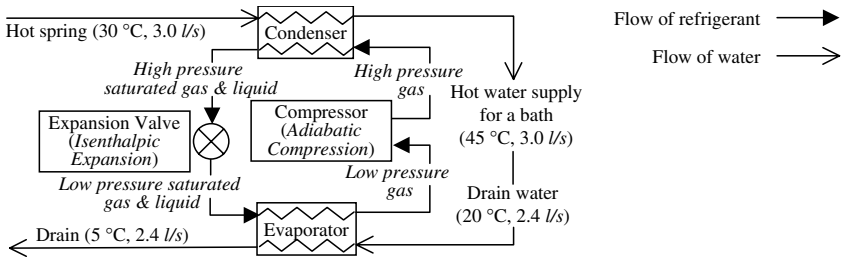


Fig. 11. A heat pump system

2. A Robotic Arm System Design Problem

Fig. 12 shows the robotic arm system. The gear-motor-A and gear-motor-B drive the joint-A and joint-B respectively. The given operation is to lift up the object along to the vertical line. It is assumed that both links have the same lengths L, and that there is a linear relationship between L and their mass m as bellow:

$$m = \gamma L,\ \gamma = 3.5[\mathrm{kg/m}].$$

The design task is to determine the link length L and select appropriate motors and gear ratios for both joints.

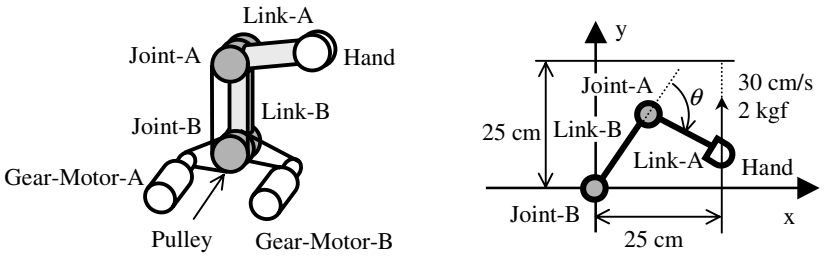


Fig. 12. A robotic arm system