# Parallel Implementation of the Polyhedral Homotopy Method

IMA Annual Program Year Workshop
"Software for Algebraic Geometry"
Minneapolis, October 23-27, 2006


Masakazu Kojima

Tokyo Institute of Technology, Tokyo, Japan


- PHoMpara — K. Fujisawa, T. Gunji, S. Kim and M. Kojima
- Multivariate Hornor scheme — M. Kojima

## Contents

1. Polyhedral homotopy method
2. PHoMpara
   - Numerical results
3. Enumeration of all mixed cells
   - Parallel implementation
   - Dynamic enumeration ---> Takeda’s Talk
4. Multivariate Hornor Scheme
   - Numerical results
5. Concluding remarks

## Contents

1. **Polyhedral homotopy method**
2. PHoMpara
    - Numerical results
3. Enumeration of all mixed cells
    - Parallel implementation
    - Dynamic enumeration ---> Takeda's Talk
4. Multivariate Hornor Scheme
    - Numerical results
5. Concluding remarks

A system of polynomial equations $f(x) = 0$, where

$$\begin{aligned} \mathbb{C} &= \text{the set of complex numbers}, \\ x &= (x_1, x_2, \ldots, x_n) \in \mathbb{C}^n, \\ f(x) &= (f_1(x), f_2(x), \ldots, f_n(x)), \\ f_j(x) &= \text{a polynomial in } n \text{ complex variables } x_1, x_2, \ldots, x_n. \end{aligned}$$

**Find all isolated solutions in $\mathbb{C}^n$**
**by the polyhedral homotopy method**

## Rough sketch of the polyhedral homotopy method

- Based on Bernshtein's theory on bounding the number of solutions of a polynomial system in terms of its mixed volume. [Bernshtein '75]

- Currently the most powerful and practical method for computing all isolated solutions of a (large & sparse) system of polynomial equations.

Implementation on a single CPU:

- PHCpack [Verschelde]
- HOM4PS [Li and Gao]
- PHoM [Gunji, Kim, Kojima, Takeda, Fujisawa and Mizutani]

- Suitable for parallel computation;
  all isolated solutions can be computed independently in parallel.

- PHoMpara [Gunji, Kim, Fujisawa and Kojima]
- Verschelde and Zhuang '06

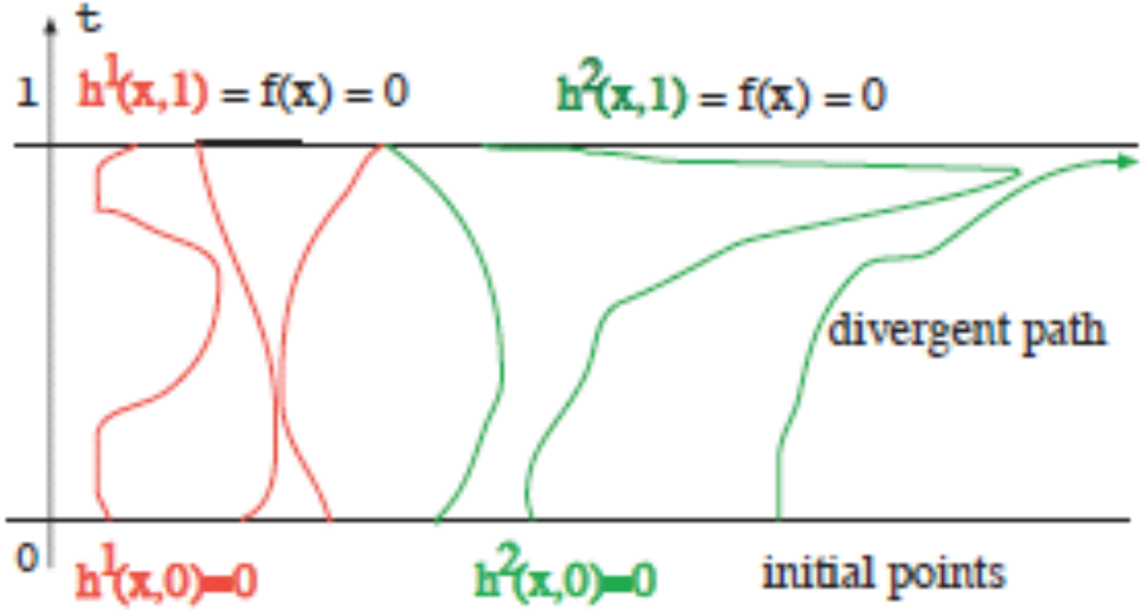
## Rough sketch of the polyhedral homotopy method — 2

**Phase 1.** Construct a family of homotopy functions.

- Comp. of all fine mixed cells $\Longrightarrow$ Comb. enumeration problem.
- Large scale linear program to reduce the powers of the homotopy parameter.

**Phase 2.** Trace homotopy paths by predictor-corrector methods.

- Highly nonlinear homotopy paths that require sophisticated techniques for step length control.
- Divergent homotopy paths. Convergence to singular solutions ("Polyhedral end game",Morgan-Sommese-Wampler '91 '92 '92, Huber-Verschelde '98).



## Rough sketch of the polyhedral homotopy method — 2

**Phase 1.** Construct a family of homotopy functions.

- Comp. of all fine mixed cells $\Longrightarrow$ Comb. enumeration problem.
- Large scale linear program to reduce the powers of the homotopy parameter.

**Phase 2.** Trace homotopy paths by predictor-corrector methods.

- Highly nonlinear homotopy paths that require sophisticated techniques for step length control.
- Divergent homotopy paths. Convergence to singular solutions ("Polyhedral end game",Morgan-Sommese-Wampler '91 '92 '92, Huber-Verschelde '98).

**Phase 3.** Verify that all isolated solutions are computed.

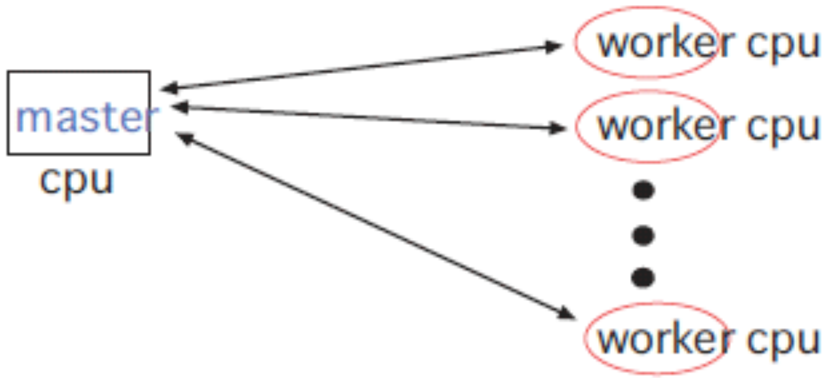
- The number of solutions is unknown in general.
- Approximate solutions are computed but exact solutions are never computed.

## Contents

1. Polyhedral homotopy method
2. PHoMpara
    - Numerical results
3. Enumeration of all mixed cells
    - Parallel implementation
    - Dynamic enumeration ---> Takeda's Talk
4. Multivariate Hornor Scheme
    - Numerical results
5. Concluding remarks

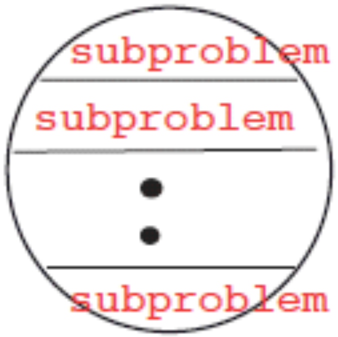
## Midleware used in PHoMpara for parallel computation

- Ninf: Master-worker computing system by Sekiguchi, et al.




(a) Each worker can not communicate with other workers.

Master Problem



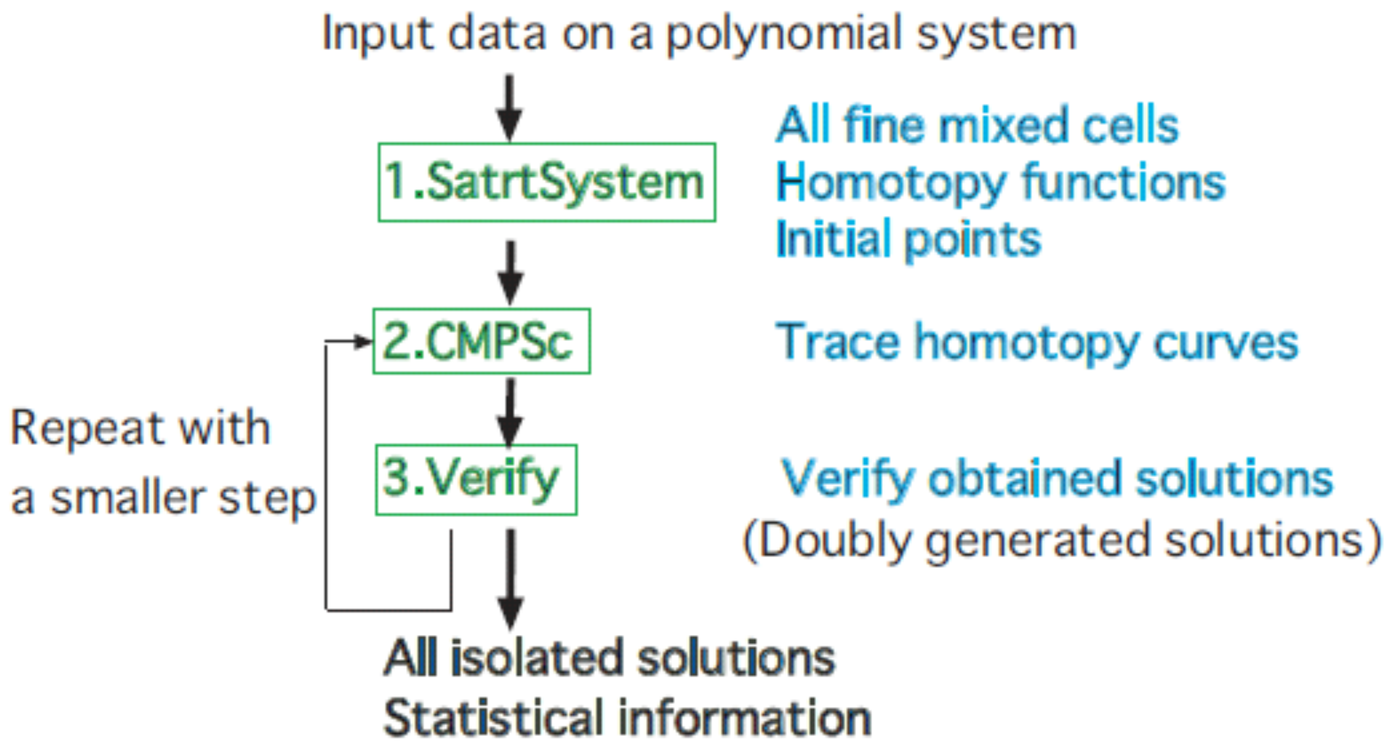

A master machine partitions a master problem into subproblems and distributes them to worker machines.

(b) Easy to use.

(c) Load balance — how to partition.

(d) Communication cost between master and worker machines.

Structure of PHoMpara



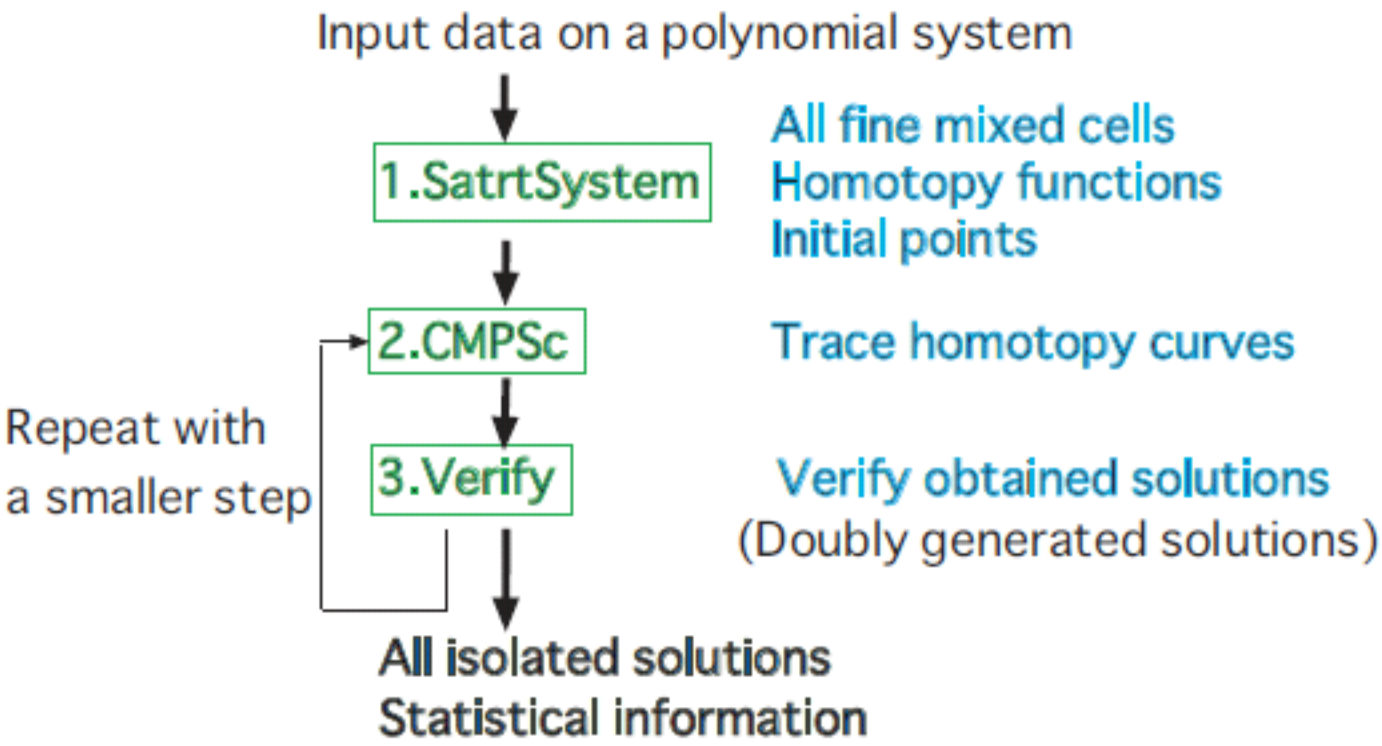

Parallel computation in 1. StartSystem

- Computation of all fine mixed cells — later.
- Balancing powers of the continuation parameter (Li-Verschelde 2000)

— an LP with a small # variables and a large # inequality constraints.

— a cutting plane (a column generation simplex) method.

Structure of PHoMpara

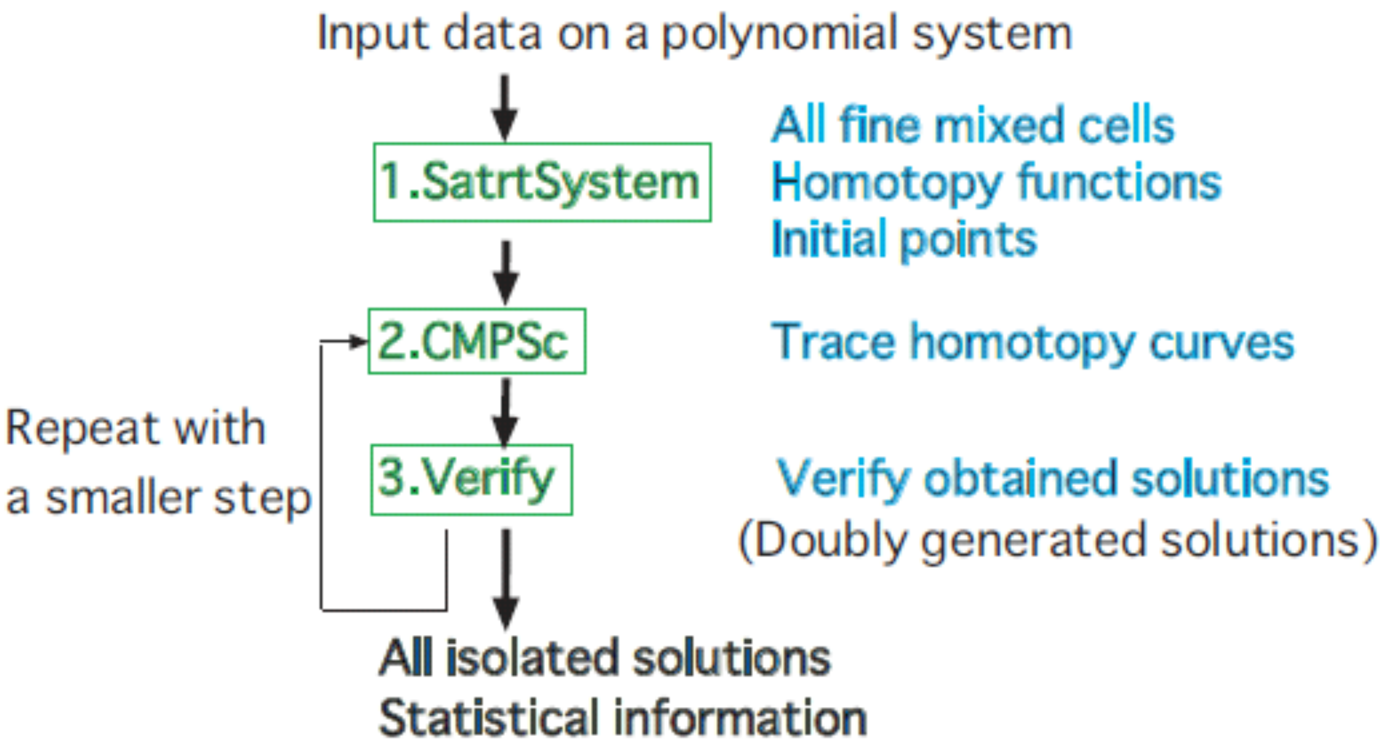



Parallel computation in 2. CMPSc

- Each homotopy curve can be traced by pred.corr. method independently.

— Easy to execute in parallel; divide the h.curves to be traced into (10× #workers) sets with almost equal size, and distribute each set to each worker.

Structure of PHoMpara




Parallel computation in 3. Verify

- Let $\{x^p\}$ be generated solutions. If $\|x^q - x^p\| < \epsilon$ then retrace.

— Sort $\{x^p\}$ according to their norms $\{\|x^p\|\}$ in parallel (quick sort).

Then the comparison is localized;

if $\|x^q\| - \|x^p\| > \epsilon$ then $\|x^q - x^p\| > \epsilon$.

## Contents

1. Polyhedral homotopy method
2. PHoMpara
    - **Numerical results**
3. Enumeration of all mixed cells
    - Parallel implementation
    - Dynamic enumeration ---> Takeda’s Talk
4. Multivariate Hornor Scheme
    - Numerical results
5. Concluding remarks

**Numerical results — 1, Scalability**

**Hardware — PC cluster (AMD Athlon 2.0GHz)**

| | | cpu time in second | | | speedup |
|---|---|---|---|---|---|
| Problem | #workers | StSy | CMPSc | Total | ratio |
| katsura-11 | 1 | 637 | 3,923 | 4,550 | 1.0 |
| | 10 | 87 | 395 | 482 | 9.4 |
| | 20 | 68 | 211 | 279 | 16.3 |
| | 40 | 58 | 102 | 160 | 28.4 |
| noon-10 | 1 | 66 | 62,600 | 62,672 | 1.0 |
| | 10 | 24 | 6,211 | 6,235 | 10.0 |
| | 20 | 24 | 3,171 | 3,195 | 19.6 |
| | 40 | 27 | 1,770 | 1,797 | 34.9 |
| eco-14 | 1 | 13,620 | 9,033 | 22,653 | 1.0 |
| | 10 | 1,383 | 909 | 2,292 | 9.9 |
| | 20 | 718 | 460 | 1,178 | 19.2 |
| | 40 | 388 | 238 | 626 | 36.2 |

**Numerical results — 2, Large scale problems**

**Hardware — PC cluster (AMD Athlon 2.0GHz × 40 workers)**

| Problem | StSy | CMPSc + Verify | | | | Total | #sol |
|---|---|---|---|---|---|---|---|
| | | Tr.1 | Tr.2 | Tr.3 | Tr.4∼6 | | |
| | cpu | cpu<br>#curv | cpu<br>#curv | cpu<br>#curv | cpu<br>#curv | cpu | |
| eco-16 | 10,470 | 1,566<br>16,384 | 15<br>8 | | | 12,051 | 16,384 |
| noon-12 | 78 | 46,737<br>531,417 | 860<br>333 | 871<br>127 | 912<br>26 | 49,458 | 531,417 |
| katsura-15 | 13,638 | 5,224<br>32,768 | 45<br>61 | 57<br>25 | | 18,964 | 32,768 |
| RPS-10 | 638 | 256<br>1,024 | | | | 894 | 1024 |
| reimer-7* | 9 | 398<br>40,320 | 399<br>37,488 | 524<br>15,512 | 2,899<br>5,915 | 4,229 | 2,880 |

* Among 40,320 curves traced, only 2,880 converged isolated solutions. To distinguish divergent curves from convergent ones, some curves were traced 6 times — Our poor technique for detecting divergence.

## Contents

Notation

For $\forall a \in \mathbb{Z}^n_+ \equiv \{(a_1,\ldots,a_n) \geq 0 : a_j \text{ is integer}\}$, $\forall x \in \mathbb{C}^n$, let

$$x^{a} = x_1^{a_1} x_2^{a_2} \cdots x_n^{a_n}.$$

Write $\forall\ f_j(x)$ of a poly. system $f(x) = (f_1(x),\ldots,f_n(x))$ as

$$f_j(x) = \sum_{a \in \mathcal{A}_j} c_j(a) x^{a},$$

where $c_j(a) \in \mathbb{C}$ $(a \in \mathcal{A}_j)$ and $\mathcal{A}_j$ a finite subset of $\mathbb{Z}^n_+$ $(j = 1,\ldots,n)$.

**A family of homotopy systems in the polyhedral homotopy method.**

Each polyhedral system $h(x,t) = (h_1(x,t),\ldots,h_n(x,t))$:

$$(3)\ h_j(x,t) \equiv \sum_{a \in \mathcal{A}_j} c_j(a) x^{a} t^{\rho_j(a)} = 0,\ (x,t) \in \mathbb{C}^n \times [0,1]\ (j = 1,\ldots,n)$$

$h(x,1) \equiv f(x)$, $h(x,0)$ : a binomial system; $0^0 = 1$

$\Uparrow$ each solution $(\alpha, \beta, \rho)$ induces a homotpy system

Choose $\omega_j(a) \in \mathbb{R}$ (randomly) $(a \in \mathcal{A}_j,\ j = 1,2,\ldots,n)$. Li '99

Find all $(\alpha,\beta) \in \mathbb{R}^{2n}$ satisfying

(1) $\rho_j(a) = \langle a, \alpha \rangle + \omega_j(a) - \beta_j \geq 0$ $(a \in \mathcal{A}_j,\ j = 1,\ldots,n)$,

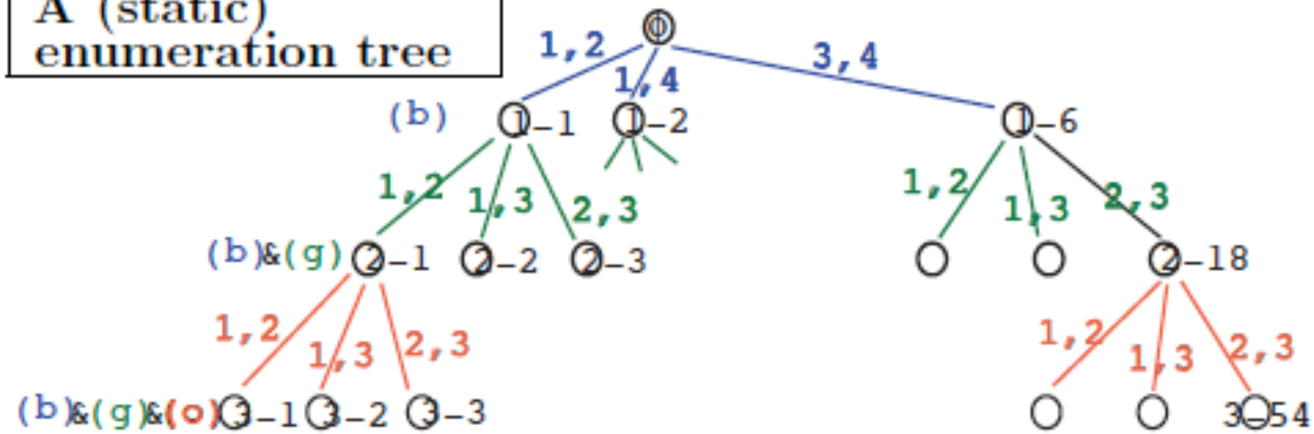
(2) for $\forall j$, exactly 2 of $\{\rho_j(a) : a \in \mathcal{A}_j\}$ are 0.

Illustration of (1) and (2): $n = 3$, a variable vector $(\alpha, \beta) \in \mathbb{R}^6$

$$(1) \quad \begin{cases} \langle a, \alpha \rangle + \omega_1(a) - \beta_1 \geq 0 \ (a \in \mathcal{A}_1 = \{a_1^1, a_2^1, a_3^1, a_4^1\}) \text{ — (b)}, \\ \langle a, \alpha \rangle + \omega_2(a) - \beta_2 \geq 0 \ (a \in \mathcal{A}_2 = \{a_1^2, a_2^2, a_3^2\}) \text{ — (g)}, \\ \langle a, \alpha \rangle + \omega_3(a) - \beta_3 \geq 0 \ (a \in \mathcal{A}_3 = \{a_1^3, a_2^3, a_3^3\}) \text{ — (o)}. \end{cases}$$

(2) requires that exactly 2 equalities hold in each group $\mathcal{A}_1$, $\mathcal{A}_2$, $\mathcal{A}_3$.

A (static) enumeration tree




- A subsystem of (1), (2) is attached to each node.
- Each edge specifies two equalities in (2).

Choose $\omega_j(a) \in \mathbb{R}$ (randomly) $(a \in \mathcal{A}_j, \ j = 1, 2, \ldots, n)$. Li '99

Find all $(\alpha, \beta) \in \mathbb{R}^{2n}$ satisfying

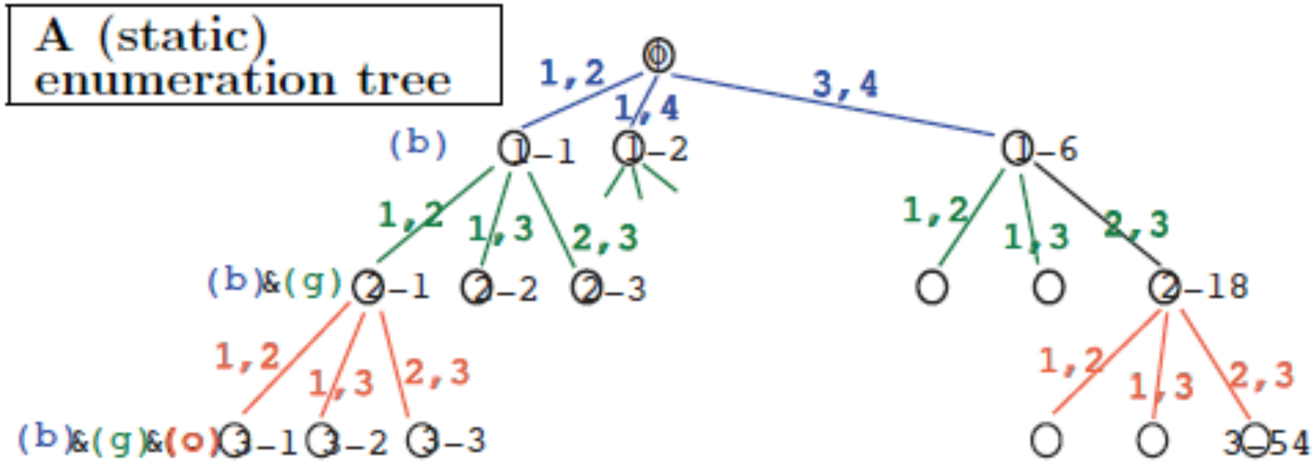
(1) $\rho_j(a) = \langle a, \alpha \rangle + \omega_j(a) - \beta_j \geq 0 \ (a \in \mathcal{A}_j, \ j = 1, \ldots, n)$,

(2) for $\forall j$, exactly 2 of $\{\rho_j(a) : a \in \mathcal{A}_j\}$ are 0.

Illustration of (1) and (2): $n = 3$, a variable vector $(\alpha, \beta) \in \mathbb{R}^6$

$$(1) \quad \begin{cases} \langle a, \alpha \rangle + \omega_1(a) - \beta_1 \geq 0 \ (a \in \mathcal{A}_1 = \{a_1^1, a_2^1, a_3^1, a_4^1\}) \text{ — (b)}, \\ \langle a, \alpha \rangle + \omega_2(a) - \beta_2 \geq 0 \ (a \in \mathcal{A}_2 = \{a_1^2, a_2^2, a_3^2\}) \text{ — (g)}, \\ \langle a, \alpha \rangle + \omega_3(a) - \beta_3 \geq 0 \ (a \in \mathcal{A}_3 = \{a_1^3, a_2^3, a_3^3\}) \text{ — (o)}. \end{cases}$$

(2) requires that exactly 2 equalities hold in each group $\mathcal{A}_1$, $\mathcal{A}_2$, $\mathcal{A}_3$.




node 1-1— (b) & 2 equalities $\langle a, \alpha \rangle + \omega_1(a) - \beta_1 = 0$ $(a = a_1^1, a_2^1)$

node 2-1 — node 1-1 & (g) & $\langle a, \alpha \rangle + \omega_1(a) - \beta_1 = 0$ $(a = a_1^2, a_2^2)$

node 3-3 — nodes 1-1, 2-1 & (o) & $\langle a, \alpha \rangle + \omega_1(a) - \beta_1 = 0$ $(a = a_2^3, a_3^3)$

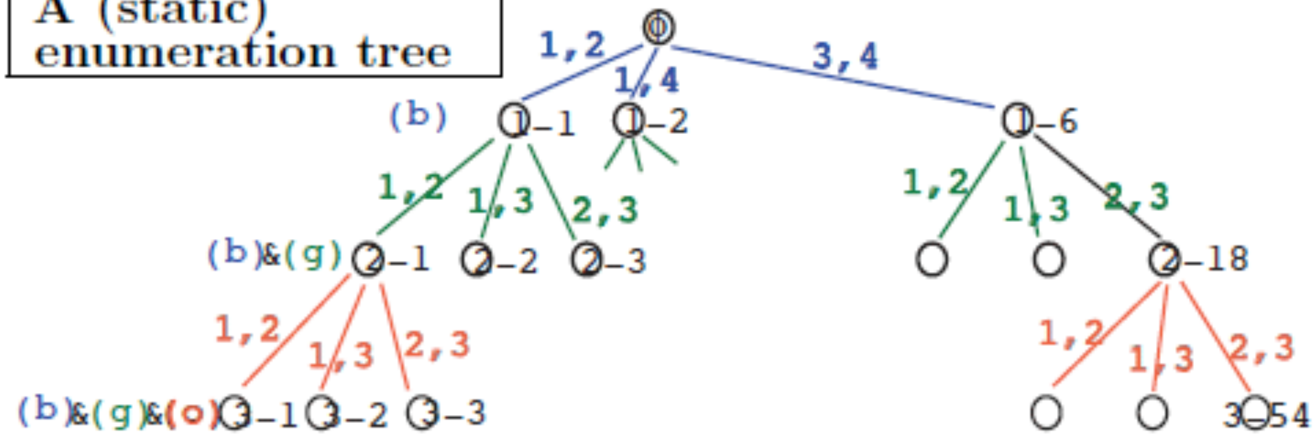
- a sol. $(\alpha, \beta)$ of (1) & (2) $\Leftrightarrow$ a feasible leaf node among 3-1, ..., 3-54.
- If a node $\ell$-$k$ is infeasible then so are its child nodes.
  $\Rightarrow$ No sol. of (1) & (2) in the subtree with the root node $\ell$-$k$.
- The feasibility of each node is checked by an LP simplex method.

Illustration of (1) and (2): $n = 3$, a variable vector $(\alpha, \beta) \in \mathbb{R}^6$

$$(1) \quad \begin{cases} \langle a, \alpha \rangle + \omega_1(a) - \beta_1 \geq 0 \ (a \in \mathcal{A}_1 = \{a_1^1, a_2^1, a_3^1, a_4^1\}) \text{ --- (b)}, \\ \langle a, \alpha \rangle + \omega_2(a) - \beta_2 \geq 0 \ (a \in \mathcal{A}_2 = \{a_1^2, a_2^2, a_3^2\}) \text{ --- (g)}, \\ \langle a, \alpha \rangle + \omega_3(a) - \beta_3 \geq 0 \ (a \in \mathcal{A}_3 = \{a_1^3, a_2^3, a_3^3\}) \text{ --- (o)}. \end{cases}$$

(2) requires that exactly 2 equalities hold in each group $\mathcal{A}_1$, $\mathcal{A}_2$, $\mathcal{A}_3$.

A (static) enumeration tree




- A single cpu implementation — the depth first search to the tree. Some techniques proposed by Gao-Li- '00, Li-Li '01.
- A parallel implementation — Assign a subtree to each worker; node 1-1 to worker 1, node 1-2 to worker 2 , . . . Additional techniques to improve the load balance among workers.

## Contents

1. Polyhedral homotopy method
2. PHoMpara
    - Numerical results
3. Enumeration of all mixed cells
    - Parallel implementation
    - Dynamic enumeration ---> Takeda’s Talk
4. **Multivariate Hornor Scheme**
    - Numerical results
5. Concluding remarks

**Minimize $\#\times$ in evaluating**

| |
|---|
| a single polynomial $f(x_1, \ldots, x_n)$. |

| |
|---|
| a single polynomial $f(x_1, \ldots, x_n)$ & its p.derivatives. |

| |
|---|
| a sys. of polynomials $f_i(x_1, \ldots, x_n)$ $(i = 1, \ldots, n)$ & their p.derivatives. |

When $n = 1$, apply the Hornor scheme and the idea of Aut. Diff.

Minimize $\#\times$ in evaluating

a single polynomial $f(x_1, \ldots, x_n)$.

a single polynomial $f(x_1, \ldots, x_n)$ & its p.derivatives.

a sys. of polynomials $f_i(x_1, \ldots, x_n)$ $(i = 1, \ldots, n)$ & their p.derivatives.

When $n \geq 2$, the situation is much more complicated.

- The Hornor scheme is not unique. Example: $n = 4$

$$f_1(x) = c_1 x_1^3 + c_2 x_1^5 x_2^3 + c_3 x_1^4 x_2^4 + c_4 x_2^2 + c_5, \ f_2(x), \ f_3(x), \ f_4(x).$$

In this case, some different "Hornor factorizations" are:

$$\begin{aligned} f_1(x) &= x_1^3(c_1 + c_2 x_1^2 x_2^3) + x_2^2(c_3 x_1^4 x_2^2 + c_4) + c_5 \text{ —(a), } 16\times \\ &= c_1 x_1^3 + x_2^2(x_1^4 x_2(c_2 x_1 + c_3 x_2) + c_4) + c_5 \text{ —(b), } 12\times \\ &= x_1^3(c_1 + x_1 x_2^3(c_2 x_1 + c_3 x_2)) + c_4 x_2^2 + c_5 \text{ —(c), } 11\times \end{aligned}$$

$f_2(x)$, $f_3(x)$, $f_4(x)$ have some different Hornor factorizations too.

- If monomials involved in the Hornor factorizations such as $x_1^3$ and $x_1^4 x_2$ were evaluated independently, we could just choose "the min. Hornor factorization" for each $f_j(x)$

Minimize $\#\times$ in evaluating

a single polynomial $f(x_1, \ldots, x_n)$.

a single polynomial $f(x_1, \ldots, x_n)$ & its p.derivatives.

a sys. of polynomials $f_i(x_1, \ldots, x_n)$ $(i = 1, \ldots, n)$ & their p.derivatives.

When $n \geq 2$, the situation is much more complicated.

- The Hornor scheme is not unique. Example: $n = 4$

$$f_1(x) = c_1 x_1^3 + c_2 x_1^5 x_2^3 + c_3 x_1^4 x_2^4 + c_4 x_2^2 + c_5, \;\; f_2(x), \;\; f_3(x), \;\; f_4(x).$$

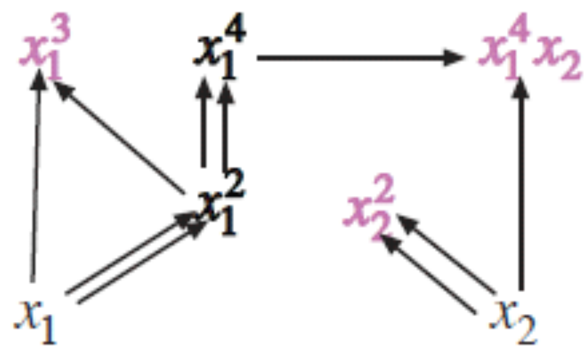
In this case, some different "Hornor factorizations" are:

$$\begin{aligned} f_1(x) &= x_1^3(c_1 + c_2 x_1^2 x_2^3) + x_2^2(c_3 x_1^4 x_2^2 + c_4) + c_5 \text{ —(a), } 16\times \\ &= c_1 x_1^3 + x_2^2(x_1^4 x_2 (c_2 x_1 + c_3 x_2) + c_4) + c_5 \text{ —(b), } 12\times \qquad \Rightarrow 10\times \\ &= x_1^3(c_1 + x_1 x_2^3 (c_2 x_1 + c_3 x_2)) + c_4 x_2^2 + c_5 \text{ —(c), } 11\times \end{aligned}$$

$f_2(x)$, $f_3(x)$, $f_4(x)$ have some different Hornor factorizations too.

- But we can save $\times$ by evaluating all monomials together.

To compute monomials with degree$\geq 2$ in (b), we need $5\times$; while $7\times$ if computed separately. Hence $f_1(x)$ can be evaluated $10\times$.



Minimize $\#\times$ in evaluating

| |
|---|
| a single polynomial $f(x_1, \ldots, x_n)$. |
| a single polynomial $f(x_1, \ldots, x_n)$ & its p.derivatives. |
| a sys. of polynomials $f_i(x_1, \ldots, x_n)$ $(i = 1, \ldots, n)$ & their p.derivatives. |

• Therefore "minimizing $\#\times$" in evaluating [ ] is a very complicated combinatorial optimization problem.

Two step methods for evaluating [ ]:

1. A min. Hornor factorization for each $f_i(x)$, assuming that the monomials involved are computed independently — next.
2. Efficient computation of all monomials involved in the Hornor factorizations — later.

Minimize $\#\times$ in evaluating

a single polynomial $f(x_1, \dots, x_n)$.

Here we assume that the monomials involved are computed independently.

**Notation and Definition**

$$(P, \alpha_p) = \sum_{p \in P} c_p x^{\alpha_p} \text{ (the coefficients } c_p \ (p \in P) \text{ are not relevant).}$$

$$(Q, \alpha_p) \text{ is factorizable iff } \exists \gamma \neq 0; (Q, \alpha_p) = x^{\gamma} \left( \sum_{p \in Q} c_p x^{\alpha_p - \gamma} \right).$$

$$\mathcal{S}((P, \alpha_p)) = \{Q \subseteq P : \#Q \geq 2, \ (Q, \alpha_p) \text{ is factorizable}\}.$$

$$(P, \alpha_p) : \begin{cases} \text{non-factorizable} & \text{if } \mathcal{S}((P, \alpha_p)) = \emptyset, \\ \text{partially-factorizable} & \text{if } \mathcal{S}((P, \alpha_p)) \neq \emptyset. \end{cases}$$

$\nu((P, \alpha_p))$ : the min. $\#\times$ to evaluate $(P, \alpha_p)$.

$$g(x) = 4x_1^2 x_2 + 3x_1 x_2^2 + 2x_2 = x_2(4x_1^2 + 3x_1^2 x_2 + 2) \quad \text{factorizable.}$$

In this case,

$$\nu(g(x)) = \deg(x_2) + \nu(4x_1^2 + 3x_1^2 x_2 + 2).$$

Minimize $\#\times$ in evaluating

a single polynomial $f(x_1, \ldots, x_n)$.

Here we assume that the monomials involved are computed independently.

Notation and Definition

$$(P, \alpha_p) = \sum_{p \in P} c_p x^{\alpha_p} \text{ (the coefficients } c_p \ (p \in P) \text{ are not relevant).}$$

$$(Q, \alpha_p) \text{ is factorizable iff } \exists \gamma \neq 0; (Q, \alpha_p) = x^{\gamma} \left( \sum_{p \in Q} c_p x^{\alpha_p - \gamma} \right).$$

$$\mathcal{S}((P, \alpha_p)) = \{Q \subseteq P : \#Q \geq 2,\ (Q, \alpha_p) \text{ is factorizable}\}.$$

$$(P, \alpha_p) : \begin{cases} \text{non-factorizable} & \text{if } \mathcal{S}((P, \alpha_p)) = \emptyset, \\ \text{partially-factorizable} & \text{if } \mathcal{S}((P, \alpha_p)) \neq \emptyset. \end{cases}$$

$\nu((P, \alpha_p))$ : the min. $\#\times$ to evaluate $(P, \alpha_p)$.

$$4x_1^2 + 2x_2^2 + 2 \quad \text{non-factorizable.}$$

In this case,

$$\nu(g(x)) = \text{the sum of degrees of all terms} = 2 + 2 + 0 = 4.$$

Minimize $\#\times$ in evaluating

a single polynomial $f(x_1, \ldots, x_n)$.

Here we assume that the monomials involved are computed independently.

**Notation and Definition**

$$(P, \alpha_p) = \sum_{p \in P} c_p x^{\alpha_p} \text{ (the coefficients } c_p \ (p \in P) \text{ are not relevant).}$$

$$(Q, \alpha_p) \text{ is factorizable iff } \exists \gamma \neq 0; (Q, \alpha_p) = x^{\gamma} \left( \sum_{p \in Q} c_p x^{\alpha_p - \gamma} \right).$$

$$\mathcal{S}((P, \alpha_p)) = \{ Q \subseteq P : \#Q \geq 2, \ (Q, \alpha_p) \text{ is factorizable} \}.$$

$$(P, \alpha_p) : \begin{cases} \text{non-factorizable} & \text{if } \mathcal{S}((P, \alpha_p)) = \emptyset, \\ \text{partially-factorizable} & \text{if } \mathcal{S}((P, \alpha_p)) \neq \emptyset. \end{cases}$$

$\nu((P, \alpha_p))$ : the min. $\#\times$ to evaluate $(P, \alpha_p)$.

$$\begin{aligned} 4x_1^2 + 3x_1x_2^2 + 2x_2 &= x_1(4x_1 + 3x_2^2) + 2x_2 \\ &= 4x_1^2 + x_2(3x_1x_2 + 2) \quad \text{partially-factorizable.} \end{aligned}$$

In this case,

$$\nu(g(x)) = \min \left\{ \nu(4x_1^2 + 3x_1x_2^2) + \nu(2x_2), \nu(4x_1^2) + \nu(3x_1x_2^2 + 2x_2) \right\}$$

Minimize $\#\times$ in evaluating

a single polynomial $f(x_1,\ldots,x_n)$.

Here we assume that the monomials involved are computed independently.

**Notation and Definition**

$$(P,\alpha_p) = \sum_{p\in P} c_p x^{\alpha_p} \text{ (the coefficients } c_p \ (p\in P) \text{ are not relevant).}$$

$$(Q,\alpha_p) \text{ is factorizable iff } \exists\gamma \neq 0; (Q,\alpha_p) = x^{\gamma}\left(\sum_{p\in Q} c_p x^{\alpha_p-\gamma}\right).$$

$$\mathcal{S}((P,\alpha_p)) = \{Q \subseteq P : \#Q \geq 2,\ (Q,\alpha_p) \text{ is factorizable}\}.$$

$$(P,\alpha_p) : \begin{cases} \text{non-factorizable} & \text{if } \mathcal{S}((P,\alpha_p)) = \emptyset, \\ \text{partially-factorizable} & \text{if } \mathcal{S}((P,\alpha_p)) \neq \emptyset. \end{cases}$$

$\nu((P,\alpha_p))$ : the min. $\#\times$ to evaluate $(P,\alpha_p)$.

$$\nu((P,\alpha_p)) = \begin{cases} \deg(\gamma) + \nu((P,\alpha_p-\gamma)) & \text{if factorizable,} \\ \sum_{p\in P}\deg(\alpha_p) & \text{if non-factorizable,} \\ \min\{\nu((Q,\alpha_p)) + \nu((P\backslash Q,\alpha_p)) : Q \in \mathcal{S}((P,\alpha_p))\} & \\ & \text{if partially-factorizable} \end{cases}$$

- The recursive formula $\nu$ + a lbd technique to compute min. $\#\times$.
- $\nu$ is too expensive for larger size poly. sys. $\Rightarrow$ Heuristic methods.

## Contents

Numerical results: $\#\times$, where all monomials are computed independently.

| | $\#\times$ (cpu time to compute $\#\times$) | | |
|---|---|---|---|
| poly. system (#eq,deg,#terms) | "$\nu$+lbd" exact meth. | Heuristic method1 | Heuristic method2 |
| game4two (4,2,8) | 28 (0.5) | 28 (0.1) | 32 (0.1) |
| butcher (7,4,9) | 70 (0.7) | 70 (0.2) | 81 (0.2) |
| pole34sys (12,3,73) | - ($\geq$ 3600) | 864 (3.1) | 1008 (6.4) |
| pltp34sys (12,4,96) | - ($\geq$ 3600) | 1212 (1212) | 1560 (9.6) |
| sparse5 (5,10,8) | 95 (0.2) | 110 (0.1) | 100 (0.1) |
| rose (3,9,29) | - ($\geq$ 3600) | 63 (0.1) | 61 (0.1) |
| cyclic-8 (8,8,8) | 128 (134.7) | 150 (0.3) | 128 (0.3) |
| cyclic-10 (10,10,10) | - ($\geq$ 3600) | 281 (0.6) | 228 (0.6) |
| cyclic-24 (24,24,24) | - ($\geq$ 3600) | 3443 (30.1) | 1932 (11.9) |

#eq : the number of equations = the number of variables,
deg = $\max_i \deg(f_i(x))$, #terms = $\max_i$ the number of terms of $f_i(x)$

"$\nu$+lbd" exact meth. — "the recursive formula $\nu$+lbd" technique".

H-1 — similar to Ceberio & Kreinovich 2004.

H-2 — gathering similar monomials.

Numerical results: $\#\times$, where all monomials are computed independently.

| | $\#\times$ (cpu time to compute $\#\times$) | | |
|---|---|---|---|
| poly. system (#eq,deg,#terms) | "$\nu$+lbd" exact meth. | Heuristic method1 | Heuristic method2 |
| game4two (4,2,8) | 28 (0.5) | 28 (0.1) | 32 (0.1) |
| butcher (7,4,9) | 70 (0.7) | 70 (0.2) | 81 (0.2) |
| pole34sys (12,3,73) | - ($\geq$ 3600) | 864 (3.1) | 1008 (6.4) |
| pltp34sys (12,4,96) | - ($\geq$ 3600) | 1212 (1212) | 1560 (9.6) |
| sparse5 (5,10,8) | 95 (0.2) | 110 (0.1) | 100 (0.1) |
| rose (3,9,29) | - ($\geq$ 3600) | 63 (0.1) | 61 (0.1) |
| cyclic-8 (8,8,8) | 128 (134.7) | 150 (0.3) | 128 (0.3) |
| cyclic-10 (10,10,10) | - ($\geq$ 3600) | 281 (0.6) | 228 (0.6) |
| cyclic-24 (24,24,24) | - ($\geq$ 3600) | 3443 (30.1) | 1932 (11.9) |

#eq : the number of equations = the number of variables,
deg = $\max_i \deg(f_i(x))$, #terms = $\max_i$ the number of terms of $f_i(x)$

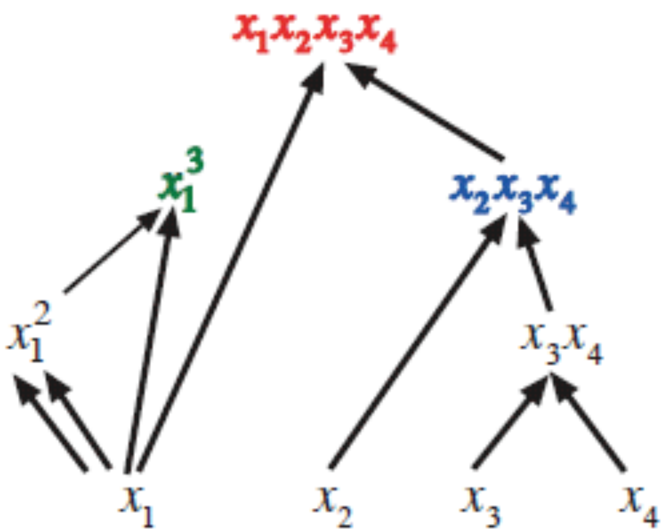
- "$\nu$+lbd" is too expensive for larger deg and/or #terms cases.
- H-1 performs better in some cases, and H-2 does in some other cases.
- In practice, try some heuristic methods and choose the best one.

2. Efficient computation of all monomials in the Hornor factorizations.

$\Rightarrow$ Except $x_1, \ldots, x_n$, $x^\alpha$ is computed as the product of two lower degree monomials; $x^\alpha = x^\beta x^\gamma$ for some $\beta,\ \gamma \in \mathbb{Z}_+^n$.

Suppose monomials $x_1x_2x_3x_4$, $x_1x_2x_3$ and $x_1^3$ are to be computed.

If we compute these monomials independently, we need 7 $\times$.




In this case, 5 $\times$ to compute all the monomials.

- A heuristic method for constructing this kind of graph.

## Contents

1. Polyhedral homotopy method
2. PHoMpara
    - Numerical results
3. Enumeration of all mixed cells
    - Parallel implementation
    - Dynamic enumeration ---> Takeda's Talk
4. Multivariate Hornor Scheme
    - Numerical results
5. Concluding remarks

Our goal:

| |
|---|
| Numerically stable and fast implementation of the polyhedral homotopy method for computing all (isolated) solutions of a large scale polynomial system |

| | Research fields |
|---|---|
| (a) Mathematical foundations on the polyhedral homotopy method | — Algebraic geometry |
| + | |
| (b) Accurate and fast homotopy curve tracing techniques | — Numerical analysis |
| + | |
| (c) Some techniques from optimization; LP, Implicit enumeration, etc. | — Optimization |
| + | |
| (d) Parallel computation | — Computer science |

Our goal:

Numerically stable and fast implementation of the polyhedral homotopy method for computing all (isolated) solutions of a large scale polynomial system

Future plan for PHoM and PHoMpara

**(i) Dynamic enumeration of all mixed cells (Mizutani-Takeda-Kojima '06) into PHoM $\Rightarrow$ speedup**

**(ii) Methods for efficient evaluation of polynomials and their prartial derivatives (Kojima '06) into PHoM $\Rightarrow$ speedup & accuracy**

**(iii) "Polyhedral end games" (Huber-Verschelde '98) to detect divergence and degenerate solutions into PHoM $\Rightarrow$ speedup & accuracy**

**(iv) Update PHoMpara, the parallel version of PHoM taking account of (i), (ii) and (iii) $\Rightarrow$ speedup & accuracy**

# Thank you!

This material is obtained at http://www.is.titech.ac.jp/~kojima/talk.html.