On the chordality of polynomial sets in triangular decomposition in top-down style

Chenqi Mou

joint work with **Yang Bai**

LMIB–School of Mathematics and Systems Science

Beihang University, China

ISSAC 2018, New York, USA

June 19, 2018

Chordal graph

$G = (V, E)$ *chordal* $\Longleftrightarrow$ for any cycle C contained in G of four or more vertexes, there is an edge $e \in E \setminus C$ connects two vertexes in C.

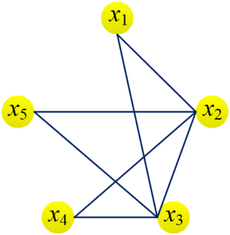


Figure: An illustrative chordal graph

Chordal graph

Perfect elimination ordering / chordal graph

$G = (V, E)$ a graph with $V = \{x_1, \ldots, x_n\}$:
An ordering $x_{i_1} < x_{i_2} < \cdots < x_{i_n}$ of the vertexes is called a *perfect elimination ordering* of G if for each $j = i_1, \ldots, i_n$, the restriction of G on

$$X_j = \{x_j\} \cup \{x_k : x_k < x_j \text{ and } (x_k, x_j) \in E\}$$

is a clique. A graph G is said to be *chordal* if there exists a perfect elimination ordering of it.

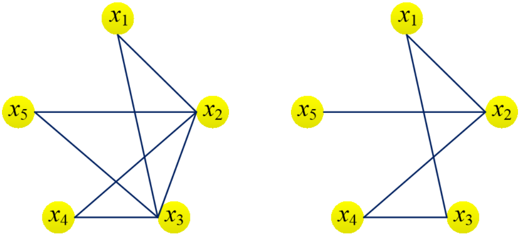


Figure: Chordal VS non-chordal graphs

Triangular set and decomposition

Triangular set in $\mathbb{K}[x_1, \ldots, x_n]$ with $x_1 < \cdots < x_n$

$$T_1(x_1, \ldots, \boxed{x_{s_1}})$$
$$T_2(x_1, \ldots, x_{s_1}, \ldots, \boxed{x_{s_2}})$$
$$T_3(x_1, \ldots, x_{s_1}, \ldots, x_{s_2}, \ldots, \boxed{x_{s_3}})$$
$$\vdots$$
$$T_r(x_1, \ldots, x_{s_1}, \ldots, x_{s_2}, \ldots, x_{s_3}, \ldots, \ldots, \boxed{x_{s_r}})$$

Triangular decomposition

Polynomial set $\mathcal{F} \subset \mathbb{K}[x_1, \ldots, x_n]$

$\Downarrow$

Triangular sets $\mathcal{T}_1, \ldots, \mathcal{T}_t$ s.t. $\mathsf{Z}(\mathcal{F}) = \bigcup_{i=1}^{t} \mathsf{Z}(\mathcal{T}_i / \operatorname{ini}(\mathcal{T}_i))$

- $\rightsquigarrow$ Solving $\mathcal{F} = 0 \Longrightarrow$ solving all $\mathcal{T}_i = 0$
- $\rightsquigarrow$ Multivariate generalization of Gaussian elimination

Inspired by the pioneering works of

D. Cifuentes

P.A. Parrilo (from MIT)

[Cifuentes and Parrilo 2017]: Connections between triangular sets and chordal graphs

- Experimental observations: algorithms for computing triangular sets due to Wang become more efficient when the input polynomial set is chordal ($\Longrightarrow$ Why?)

Associated graphs of polynomial sets

$\mathrm{supp}(\mathcal{F})$ for $F \subset \mathbb{K}[x_1, \ldots, x_n]$: the set of variables which appear in $\mathcal{F}$

Associated graphs

$\mathcal{F} \subset \mathbb{K}[x_1, \ldots, x_n]$, *associated graph* $G(\mathcal{F})$ of $\mathcal{F}$ is an undirected graph:

(a) vertexes of $G(\mathcal{F})$: the variables in $\mathrm{supp}(\mathcal{F})$

(b) edge (x_i, x_j) in $G(\mathcal{F})$: if there exists one polynomial $F \in \mathcal{F}$ with $x_i, x_j \in \mathrm{supp}(F)$

Chordal polynomial set

A polynomial set $\mathcal{F} \subset \mathbb{K}[x_1, \ldots, x_n]$ is said to be *chordal* if $G(\mathcal{F})$ is chordal.

Associated graphs of polynomial sets

$$\mathbb{K}[x_1, \ldots, x_5]$$

$$\mathcal{P} = \{x_2 + x_1, x_3 + x_1, x_4^2 + x_2, x_4^3 + x_3, x_5 + x_2, x_5 + x_3 + x_2\}$$

$$\mathcal{Q} = \{x_2 + x_1, x_3 + x_1, x_3, x_4^2 + x_2, x_4^3 + x_3, x_5 + x_2\}$$

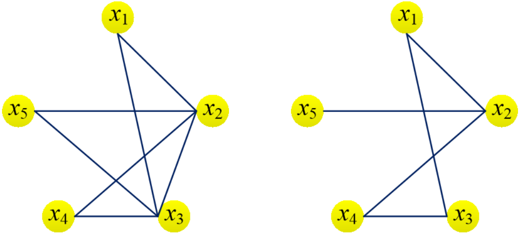
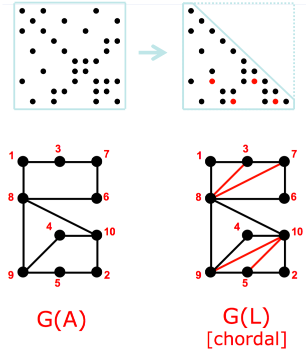


Figure: Associated graphs $G(\mathcal{P})$ (chordal) and $G(\mathcal{Q})$ (not chordal)

Chordal graphs in Gaussian elimination

- Tutorial by Chandrasekaran: Guassian elimination w.r.t. a perfect elimination ordering $\Longrightarrow$ no new fill-ins $\Longrightarrow$ sparse Gaussian elimination: sparse + chordal [Parter 61, Rose 70, Gilbert 94]
- Matrix with a chordal associated graph $\Longrightarrow$ Matrix in echolon-form with a subgraph



(credits to J. Gilbert)

Triangular decomposition in top-down style

The variables are handled in a strictly decreasing order: $x_n, x_{n-1}, \ldots, x_1$

- widely used strategy [Wang 1993, 1998, 2000], [Chai, Gao, Yuan 2008]
- the closest to Gaussian elimination
- algorithms due to Wang are mostly in top-down style (!!):

Efficient when chordal

$\Downarrow$

Chordal graphs in Gaussian elimination

$\Downarrow$

Triangular decomposition in top-down style
closest to Gaussian elimination

$\Downarrow$

Algorithms due to Wang are in top-down style

Triangular decomposition in top-down style

The variables are handled in a strictly decreasing order: $x_n, x_{n-1}, \ldots, x_1$

- widely used strategy [Wang 1993, 1998, 2000], [Chai, Gao, Yuan 2008]
- the closest to Gaussian elimination
- algorithms due to Wang are mostly in top-down style (!!):

Efficient when chordal

$\Downarrow$

Chordal graphs in Gaussian elimination

$\Downarrow$

Triangular decomposition in top-down style
closest to Gaussian elimination

$\Downarrow$

Algorithms due to Wang are in top-down style

Triangular decomposition in top-down style

The variables are handled in a strictly decreasing order: $x_n, x_{n-1}, \ldots, x_1$

- widely used strategy [Wang 1993, 1998, 2000], [Chai, Gao, Yuan 2008]
- the closest to Gaussian elimination
- algorithms due to Wang are mostly in top-down style (!!):

Efficient when chordal

$\Downarrow$

Chordal graphs in Gaussian elimination

$\Downarrow$

Triangular decomposition in top-down style
closest to Gaussian elimination

$\Downarrow$

Algorithms due to Wang are in top-down style

Problems

For a chordal polynomial set:

- Changes of graph structures of the polynomial sets in the process of triangular decomposition in top-down style
- Relationships (like inclusion) between associated graphs of computed triangular sets and the input polynomial set

Chordal graphs in Gaussian elimination $\Longrightarrow$ Chordal graphs in triangular decomposition in top-down style: multivariate generalization

Reduction w.r.t. one variable in triangular decomposition

$\mathcal{P} \subset \mathbb{K}[x_1, \ldots, x_n]$: $\mathcal{P}^{(i)} = \{P \in \mathcal{P} : \mathrm{lv}(P) = x_i\}$

Theorem

$\mathcal{P} \subset \mathbb{K}[x_1, \ldots, x_n]$ chordal, $x_1 < \cdots < x_n$ perfect elimination ordering: Let $T \in \mathbb{K}[x_1, \ldots, x_n]$ with $\mathrm{lv}(T) = x_n$ and $\mathrm{supp}(T) \subset \mathrm{supp}(\mathcal{P}^{(n)})$, and $\mathcal{R} \subset \mathbb{K}[x_1, \ldots, x_n]$ with $\mathrm{supp}(\mathcal{R}) \subset \mathrm{supp}(\mathcal{P}^{(n)}) \setminus \{x_n\}$. Then for

$$\tilde{\mathcal{P}} = \{\tilde{\mathcal{P}}^{(1)}, \ldots, \tilde{\mathcal{P}}^{(n-1)}, T\},$$

where $\tilde{\mathcal{P}}^{(k)} = \mathcal{P}^{(k)} \cup \mathcal{R}^{(k)}$ for $k = 1, \ldots, n-1$, we have $G(\tilde{\mathcal{P}}) \subset G(\mathcal{P})$

$$\begin{array}{ccccccc}
\mathcal{P} = \{\mathcal{P}^{(1)}, & \mathcal{P}^{(2)}, & \ldots, & \mathcal{P}^{(n)}\}: & & G(\mathcal{P}) \text{ chordal} \\
\Downarrow & \Downarrow & & \Downarrow & & \cup \\
\tilde{\mathcal{P}} = \{\tilde{\mathcal{P}}^{(1)}, & \tilde{\mathcal{P}}^{(2)}, & \ldots, & T\}: & & G(\tilde{\mathcal{P}}) \\
/\!/ & \| & & s.t. & & \\
\mathcal{P}^{(1)} \cup \mathcal{R}^{(1)}, & \mathcal{P}^{(2)} \cup \mathcal{R}^{(2)}, & \ldots, & \mathrm{supp}(T) \subset \mathrm{supp}(\mathcal{P}^{(n)}) & &
\end{array}$$

$\rightsquigarrow$ In particular, $\mathrm{supp}(T) = \mathrm{supp}(\mathcal{P}^{(n)}) \Longrightarrow G(\tilde{\mathcal{P}}) = G(\mathcal{P})$

Some notations

mapping f_i

$$f_i : 2^{\mathbb{K}[\boldsymbol{x}_i]\setminus\mathbb{K}[\boldsymbol{x}_{i-1}]} \to (\mathbb{K}[\boldsymbol{x}_i] \setminus \mathbb{K}[\boldsymbol{x}_{i-1}]) \times 2^{\mathbb{K}[\boldsymbol{x}_{i-1}]}$$
$$\mathcal{P} \mapsto (T, \mathcal{R})$$

s.t $\mathrm{supp}(T) \subset \mathrm{supp}(\mathcal{P})$ and $\mathrm{supp}(\mathcal{R}) \subset \mathrm{supp}(\mathcal{P})$ (where $\mathbb{K}[\boldsymbol{x}_0] = \mathbb{K}$).

$\mathcal{P} \subset \mathbb{K}[x_1, \ldots, x_n]$ and a fixed integer i $(1 \leq i \leq n)$, suppose that $(T_i, \mathcal{R}_i) = f_i(\mathcal{P}^{(i)})$ for some f_i. For $j = 1, \ldots, n$, define

$$\mathrm{red}_i(\mathcal{P}^{(j)}) := \begin{cases} \mathcal{P}^{(j)}, & \text{if } j > i \\ \{T_i\}, & \text{if } j = i \\ \mathcal{P}^{(j)} \cup \mathcal{R}_i^{(j)}, & \text{if } j < i \end{cases}$$

and $\mathrm{red}_i(\mathcal{P}) := \cup_{j=1}^{n} \mathrm{red}_i(\mathcal{P}^{(j)})$. In particular, write

$$\boxed{\overline{\mathrm{red}}_i(\mathcal{P}) := \mathrm{red}_i(\mathrm{red}_{i+1}(\cdots(\mathrm{red}_n(\mathcal{P}))\cdots))}$$

The above theorem becomes

$G(\mathrm{red}_n(\mathcal{P})) \subset G(\mathcal{P})$, and the equality holds if $\mathrm{supp}(T_n) = \mathrm{supp}(\mathcal{P}^{(n)})$.

Reduction w.r.t. all variables in triangular decomposition

$$
\begin{array}{rllllll}
\mathcal{P} = \{\mathcal{P}^{(1)}, & \mathcal{P}^{(2)}, & \ldots, & \mathcal{P}^{(n-1)}, & \mathcal{P}^{(n)}\}: & G(\mathcal{P}) \text{ chordal} \\
\Downarrow & \Downarrow & & \Downarrow & \Downarrow & \cup \\
\mathrm{red}_n(\mathcal{P}) = \{\tilde{\mathcal{P}}^{(1)}, & \tilde{\mathcal{P}}^{(2)}, & \ldots, & \tilde{\mathcal{P}}^{(n-1)}, & T_n\}: & G(\mathrm{red}_n(\mathcal{P})) \\
\Downarrow & \Downarrow & & \Downarrow & \Downarrow & \boxed{??} \\
\overline{\mathrm{red}}_{n-1}(\mathcal{P}) = \{\tilde{\tilde{\mathcal{P}}}^{(1)}, & \tilde{\tilde{\mathcal{P}}}^{(2)}, & \ldots, & T_{n-1}, & T_n\}: & G(\overline{\mathrm{red}}_{n-1}(\mathcal{P})) \\
 & & \vdots & & & \boxed{??} \\
\overline{\mathrm{red}}_1(\mathcal{P}) = \{T_1, & T_2, & \ldots, & T_{n-1}, & T_n\}: & G(\overline{\mathrm{red}}_1(\mathcal{P}))
\end{array}
$$

Proposition

$\mathcal{P} \subset \mathbb{K}[x_1, \ldots, x_n]$ chordal, $x_1 < \cdots < x_n$ perfect elimination ordering:

For each i $(1 \leq i \leq n)$, suppose that $(T_i, \mathcal{R}_i) = f_i(\overline{\mathrm{red}}_{i+1}(\mathcal{P})^{(i)})$ for some f_i and $\mathrm{supp}(T_i) = \mathrm{supp}(\overline{\mathrm{red}}_{i+1}(\mathcal{P})^{(i)})$. Then

$$G(\overline{\mathrm{red}}_1(\mathcal{P})) = \cdots = G(\overline{\mathrm{red}}_{n-1}(\mathcal{P})) = G(\mathrm{red}_n(\mathcal{P})) = G(\mathcal{P}).$$

Counter example for successive inclusions

$\mathrm{supp}(T_i) \subset \mathrm{supp}(\overline{\mathrm{red}}_{i+1}(\mathcal{P})^{(i)})$: then in general we will NOT have

$$G(\overline{\mathrm{red}}_1(\mathcal{P})) \subset \cdots \subset G(\overline{\mathrm{red}}_{n-1}(\mathcal{P})) \subset G(\mathrm{red}_n(\mathcal{P})) \subset G(\mathcal{P})$$

Example

$$\mathcal{P} = \{x_2 + x_1, x_3 + x_1, x_4^2 + x_2, x_4^3 + x_3, x_5 + x_2, x_5 + x_3 + x_2\}$$

$$\mathcal{Q} = \mathrm{red}_5(\mathcal{P}) = \{x_2 + x_1, x_3 + x_1, x_3, x_4^2 + x_2, x_4^3 + x_3, x_5 + x_2\}$$

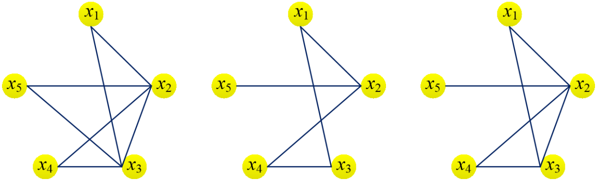
$$\Downarrow$$

$$T_4 = \mathrm{prem}(x_4^3 + x_3, x_4^2 + x_2) = -x_2x_4 + x_3,$$

$$\mathcal{R}_4 = \{\mathrm{prem}(x_4^2 + x_2, -x_2x_4 + x_3)\} = \{x_3^2 - x_2^3\},$$

$$\Downarrow$$

$$\mathcal{Q}' := \overline{\mathrm{red}}_4(\mathcal{P}) = \{x_2 + x_1, x_3 + x_1, x_3^2 - x_2^3, x_3, -x_2x_4 + x_3, x_5 + x_2\}.$$



Subgraphs of the input chordal graph

Theorem

$\mathcal{P} \subset \mathbb{K}[x_1, \ldots, x_n]$ chordal, $x_1 < \cdots < x_n$ perfect elimination ordering:

For each $i = n, \ldots, 1$, $\boxed{G(\overline{\mathrm{red}}_i(\mathcal{P})) \subset G(\mathcal{P})}$.

Corollary

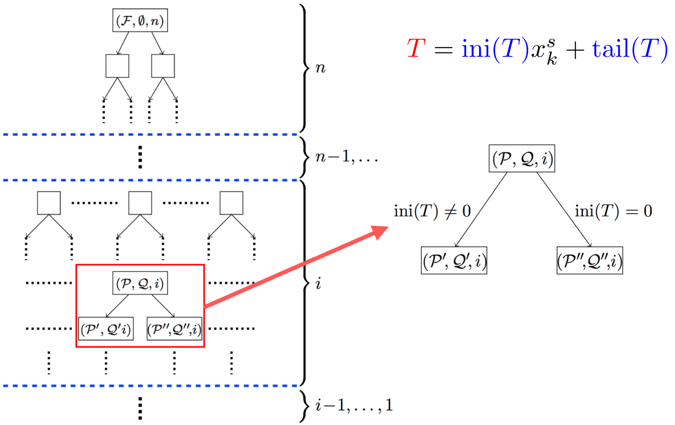
$\mathcal{P} \subset \mathbb{K}[x_1, \ldots, x_n]$ chordal, $x_1 < \cdots < x_n$ perfect elimination ordering:

If $\mathcal{T} := \overline{\mathrm{red}}_1(\mathcal{P})$ does not contain any nonzero constant, then $\mathcal{T}$ forms a triangular set such that $G(\mathcal{T}) \subset G(\mathcal{P})$.

- $\mathcal{T}$ above: the main component in the triangular decomposition
- Valid for ANY algorithms for triangular decomposition in top-down style
- Problem: what about the other triangular sets? (splitting strategies)

Wang's method: binary decomposition tree

[Wang 93]: Wang's method, simply-structured algorithm for triangular decomposition in top-down style



$$\mathcal{P}' := \mathcal{P} \setminus \mathcal{P}^{(i)} \cup \{T\} \cup \{\mathrm{prem}(P, T) : P \in \mathcal{P}\}, \quad \mathcal{Q}' := \mathcal{Q} \cup \{\mathrm{ini}(T)\},$$

$$\mathcal{P}'' := \mathcal{P} \setminus \{T\} \cup \{\mathrm{ini}(T), \mathrm{tail}(T)\}, \quad \mathcal{Q}'' := \mathcal{Q},$$

Wang's method: left child

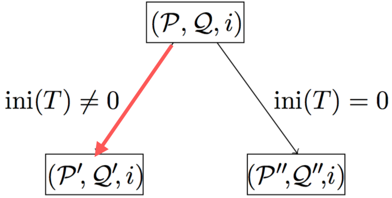
Proposition: Wang's method applied to $\mathcal{F} \subset \mathbb{K}[x_1, \ldots, x_n]$, chordal

$\mathcal{F} \subset \mathbb{K}[x_1, \ldots, x_n]$ chordal, $x_1 < \cdots < x_n$ perfect elimination ordering:

$(\mathcal{P}, \mathcal{Q}, i)$ arbitrary node in the binary decomposition tree such that $G(\mathcal{P}) \subset G(\mathcal{F})$, $T \in \mathcal{P}$ with minimal degree in x_i. Denote

$$\mathcal{P}' = \mathcal{P} \setminus \mathcal{P}^{(i)} \cup \{T\} \cup \{\mathrm{prem}(P, T) : \ P \in \mathcal{P}^{(i)}\}.$$

Then $G(\mathcal{P}') \subset G(\mathcal{F})$.



$G(\mathcal{P}') \subset G(\mathcal{F})$ on the conditions that $G(\mathcal{F})$ is chordal and $G(\mathcal{P}) \subset G(\mathcal{F})$

Wang's method: right child

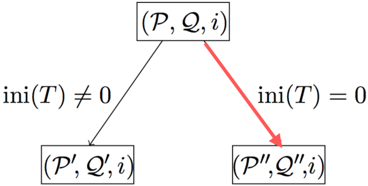
Proposition

$(\mathcal{P}, \mathcal{Q}, i)$ arbitrary node in the binary decomposition tree, $T \in \mathcal{P}^{(i)}$ with minimal degree in x_i. Denote

$$\mathcal{P}'' = \mathcal{P} \setminus \{T\} \cup \{\mathrm{ini}(T), \mathrm{tail}(T)\}.$$

Then $G(\mathcal{P}'') \subset G(P)$.

$\rightsquigarrow$In particular, $\mathrm{supp}(\mathrm{tail}(T)) = \mathrm{supp}(T) \Longrightarrow G(\mathcal{P}'') = G(\mathcal{P})$.

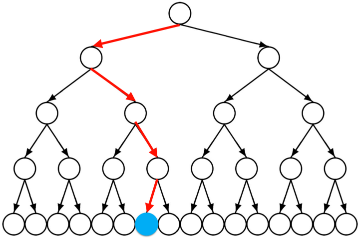


$G(\mathcal{P}'') \subset G(P)$ under no conditions

Wang's method: any node

Theorem: Wang's method applied to $\mathcal{F} \subset \mathbb{K}[x_1, \ldots, x_n]$, chordal

$\mathcal{F} \subset \mathbb{K}[x_1, \ldots, x_n]$ chordal, $x_1 < \cdots < x_n$ perfect elimination ordering:
For any node $(\mathcal{P}, \mathcal{Q}, i)$ in the binary decomposition tree, $G(\mathcal{P}) \subset G(\mathcal{F})$



Corollary: Wang's method applied to $\mathcal{F} \subset \mathbb{K}[x_1, \ldots, x_n]$, chordal

$\mathcal{F} \subset \mathbb{K}[x_1, \ldots, x_n]$ chordal, $x_1 < \cdots < x_n$ perfect elimination ordering:
For any triangular set $\mathcal{T}$ computed by Wang's method, $G(\mathcal{T}) \subset G(\mathcal{F})$

Problems solved

Variable sparsity of polynomial sets

Variable sparsity

$G(\mathcal{F}) = (V, E)$ associated graph of $\mathcal{F} = \{F_1, \ldots, F_r\} \subset \mathbb{K}[x_1, \ldots, x_n]$. Define the *variable sparsity* $s_v(\mathcal{F})$ of $\mathcal{F}$ as

$$s_v(\mathcal{F}) = |E| / \binom{2}{|V|},$$

denominator: edge number of a complete graph of $|V|$ vertexes

Sparse triangular decomposition

Sparse Gaussian elimination $\Longrightarrow$ sparse triangular decomposition in top-down style: multivariate generalization, on-going work

- sparse Gröbner bases [Faugère, Spaenlehauer, Svartz 2014]
- sparse FGLM algorithms [Faugère, Mou 2011, 2017]

Complexity analysis for triangular decomposition in top-down style

Chordal completion

For a graph G, another graph G' is called a *chordal completion* of $\mathcal{G}$ if G' is chordal with G as its subgraph.

The *treewidth* of a graph G is defined to be the minimum of the sizes of the largest cliques in all the possible chordal completions of G.

- many NP-complete problems related to graphs can be solved efficiently for graphs of bounded treewidth [Arnborg, Proskurowski 1989]
- Complexities for computing Gröbner bases for polynomial sets with small treewidth [Cifuents and Parrilo 2016]

Reminding you of the inclusion of graphs for Wang's method

The input chordal associated graph: upper bound

- Complexities for triangular decomposition: first for polynomial sets with chordal graphs / small treewidth

Future works

- Chordality in regular decomposition in top-down style: the most popular triangular decomposition
  $\Longrightarrow$ A dynamic multi-branch decomposition tree
- More other graph structures? perfect, linear, directed graphs...
  $\Longrightarrow$ Suggestion welcome!

Thanks!

A refined algorithm for regular decomposition

Input: a polynomial set $\mathcal{F} \subset \mathbb{K}[\boldsymbol{x}]$
Output: a variable ordering $\overline{\boldsymbol{x}}$ and a regular decomposition Φ of $\mathcal{F}$ with respect to $\overline{\boldsymbol{x}}$

1. Compute the variable sparsity s_v of $\mathcal{F}$
2. If s_v is smaller than some sparsity threshold s_0 ($\mathcal{F}$ is sparse), then
    1. If $G(\mathcal{F})$ is chordal, then compute its perfect elimination ordering $\overline{\boldsymbol{x}}$ [1]
    2. Else compute its chordal completion $\overline{G}(\mathcal{F})$ [2] and a perfect elimination ordering $\overline{\boldsymbol{x}}$ of $\overline{G}(\mathcal{F})$
3. Compute the regular decomposition of $\mathcal{F}$ with respect to $\overline{\boldsymbol{x}}$ with a top-down algorithm [3]

---

[1] [Rose, Tarjan, and Lueker 1976]
[2] [Bodlaender and Koster 2008]
[3] Say, [Wang 2000]

Sparse triangular decomposition

A sparse polynomial system arising from the lattice reachability problem [Cifuentes and Parrilo 2017], [Diaconis, Eisenbud, Sturmfels 1998]

$$\mathcal{F}_i := \{x_k x_{k+3} - x_{k+1} x_{k+2} : k = 1, 2, \ldots, i\}, \quad i \in \mathbb{Z}_{>0}$$

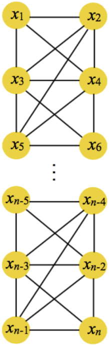


Figure: Associated graph of $\mathcal{F}_i$

Sparse triangular decomposition

Comparisons of timings for computing regular decomposition of one class of chordal and variable sparse polynomials [Cifuentes and Parrilo 2017]

$$\mathcal{F}_i := \{x_k x_{k+3} - x_{k+1} x_{k+2} : k = 1, 2, \ldots, i\}, \quad i \in \mathbb{Z}_{>0}$$

Table: Regular decomposition with RegSer in Epsilon: top-down

n	s_v	t_p	t_r					$\bar{t}_r$	$\bar{t}_r/t_p$
10	0.53	0.19	0.14	0.21	0.22	0.11	0.21	0.18	0.95
20	0.28	1.44	4.24	4.45	3.15	4.41	4.65	4.18	2.90
25	0.23	4.25	50.62	20.29	15.55	25.01	35.10	29.31	6.90
30	0.19	11.94	177.37	185.94	130.54	142.97	103.42	148.05	12.40
35	0.17	42.33	560.56	291.35	633.43	320.98	938.45	548.95	12.97
40	0.15	161.11	1883.64	3618.04	4289.13	4013.99	2996.37	3360.23	20.86

Table: Regular decomposition with RegularChains in Maple: not top-down

n	s_v	t_p	t_r					$\bar{t}_r$	$\bar{t}_r/t_p$
15	0.37	45.90	17.29	21.41	13.62	32.50	19.63	20.89	0.46
17	0.33	216.69	87.29	197.35	104.86	68.28	130.83	117.72	0.54
19	0.30	1303.08	415.90	308.37	780.75	221.75	831.15	511.58	0.39
21	0.27	8787.32	1823.29	2064.55	2431.49	1926.02	1593.36	1967.74	0.22

Sparse triangular decomposition

Comparisons of timings for computing regular decomposition of one class of chordal and variable sparse polynomials [Cifuentes and Parrilo 2017]

$$\mathcal{F}_i := \{x_k x_{k+3} - x_{k+1} x_{k+2} : k = 1, 2, \ldots, i\}, \quad i \in \mathbb{Z}_{>0}$$

Table: Regular decomposition with RegSer in Epsilon: top-down

n	s_v	t_p	t_r					$\bar{t}_r$	$\bar{t}_r/t_p$
10	0.53	0.19	0.14	0.21	0.22	0.11	0.21	0.18	0.95
20	0.28	1.44	4.24	4.45	3.15	4.41	4.65	4.18	2.90
25	0.23	4.25	50.62	20.29	15.55	25.01	35.10	29.31	6.90
30	0.19	11.94	177.37	185.94	130.54	142.97	103.42	148.05	12.40
35	0.17	42.33	560.56	291.35	633.43	320.98	938.45	548.95	12.97
40	0.15	161.11	1883.64	3618.04	4289.13	4013.99	2996.37	3360.23	20.86

Table: Regular decomposition with RegularChains in Maple: not top-down

n	s_v	t_p	t_r					$\bar{t}_r$	$\bar{t}_r/t_p$
15	0.37	45.90	17.29	21.41	13.62	32.50	19.63	20.89	0.46
17	0.33	216.69	87.29	197.35	104.86	68.28	130.83	117.72	0.54
19	0.30	1303.08	415.90	308.37	780.75	221.75	831.15	511.58	0.39
21	0.27	8787.32	1823.29	2064.55	2431.49	1926.02	1593.36	1967.74	0.22