Fault Diagnosability Analysis of Multi-Mode Systems

Fatemeh Hashemniya*
Benoît Caillaud**
Erik Frisk*
Mattias Krysander*
Mathias Malandain**

*Linköping University, Sweden
**INRIA, France

Overview and Goal

Model-based fault diagnostics

Goal

Reuse design models to optimize system exploitation: detect component aging and failure early to minimize down time

$$\dot{x} = f(x,t)$$
$$y = g(x)$$
$$y = Y(t)$$

Model, faults, sensors & data

- System model : autonomous dynamical system, eg. ODEs
- Component fault : system behavior does not match one or several equations capturing the physics of the component
- Sensors : direct or indirect measurement of some of the system state variables
- Measurement data : additional equations rendering the model overdetermined

Structural method

- ODEs ➜ DAEs
- Exploit the structural redundancies
- Syndrome : minimal set of of structurally overdetermined equations
- Alarm : syndrome residuals can not be zero

Overview and Goal

Structural fault diagnostics for multi-mode systems

Li-ion batteries are keys for the sustainable electrification of vehicles.

Challenges

- Multi-mode systems (switched systems) with frequent mode changes
- Complex systems with many components
- Direct diagnosis --> Exponential explosion of analysis complexity

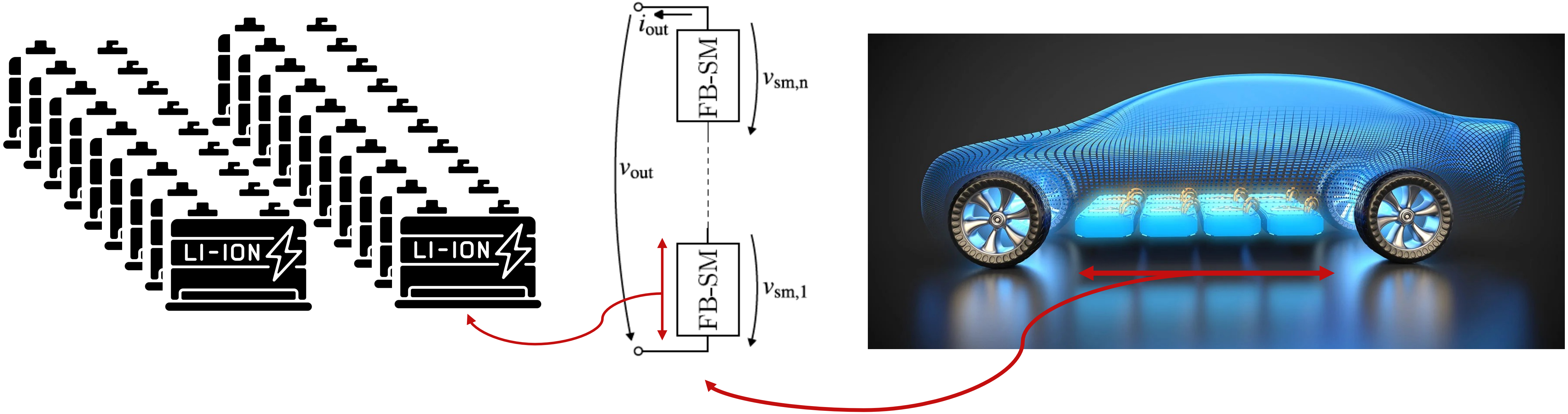
GOAL:

- ➢ Tools extension and development
- ➢ Overcome the complexity of diagnostics
- ➢ New fault modeling approach

Multi- Mode System

(motivation)

Reconfigurable battery system is an example of multi-mode systems.

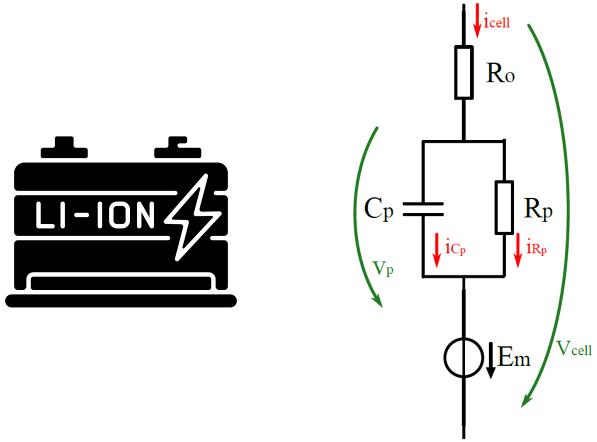


Multi- Mode Battery System

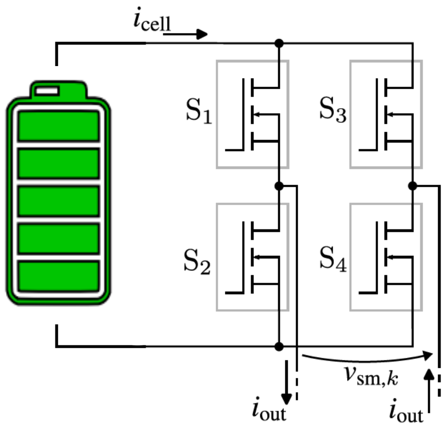
(simplified Battery example)

Based on the switch positions, there are 4 valid modes!

A Li-ion cell is modeled by an equivalent circuit model.



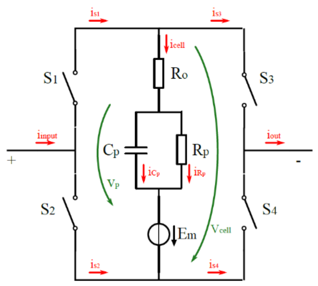
Full bridge modular multilevel converter (MMC) to produce AC.



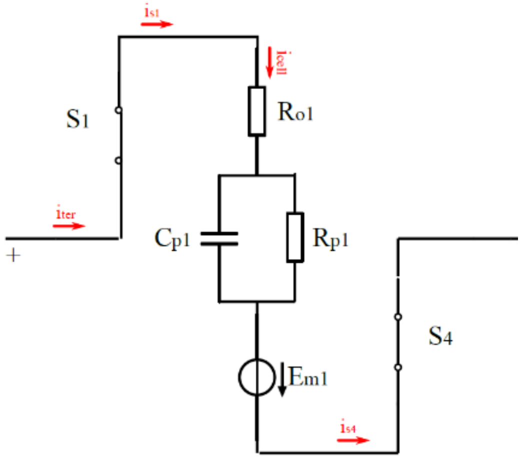
Multi- Mode Battery System

(Different modes)

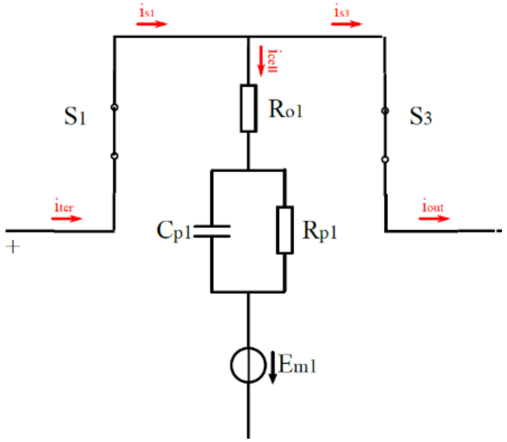
Based on the switch positions, the system works in different modes!



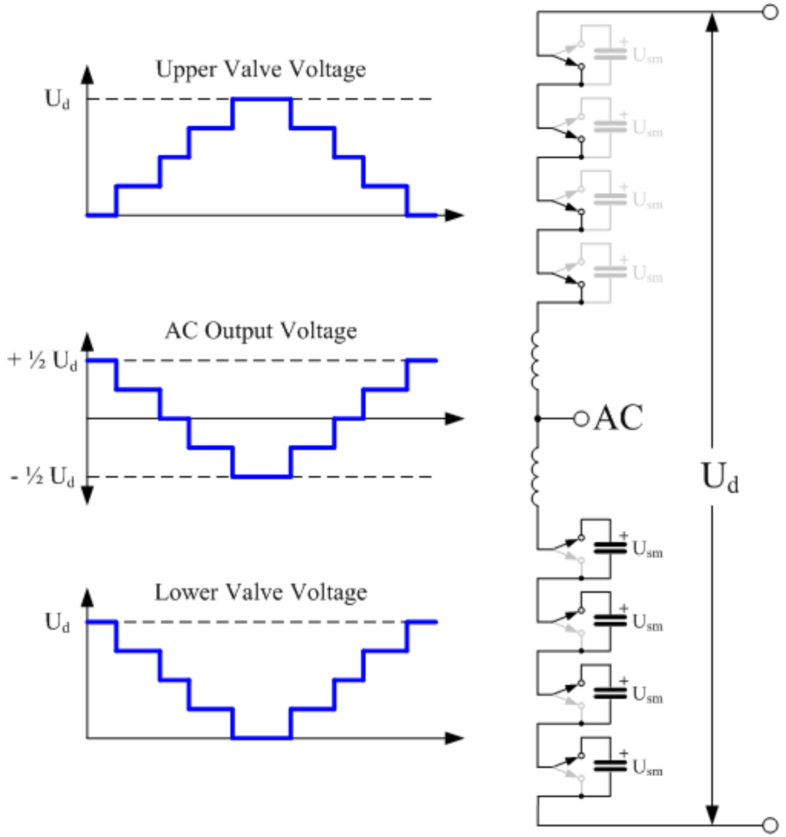
There are 16 possible modes, but only 4 modes are valid.
Others are not feasible.



Connected



Bypassed



Operating principle of Modular Multi-Level Converter (MMC).
Source: Wikipedia

Multi- Mode DAE*

Equations valid for all modes:

$$\dot{v}_p = \frac{i}{C_p} - \frac{v_p}{R_p C_p} \quad (1)$$

$$v_{cell} = v_p + R_0 i + E_m + f_{cell} \quad (2)$$

$$y_i = i + f_i \quad (3)$$

$$y_v = v_{cell} + f_v \quad (4)$$

Measured data

$$\dot{v}_p = \frac{dv_p}{dt} \quad (5)$$

Same

Switch dependent equations:

Forward/ Backward mode:

$$v_{sm} = \pm v_{cell} \quad (6)$$

$$i = \pm i_{pack} \quad (7)$$

Bypass 1&2 mode:

$$v_{sm} = 0 \quad (6)$$

$$i = 0 \quad (7)$$

Different

The system structure changes when the mode changes!

Equations valid for a special mode:

$$v_{sm} = if\ (Forward)\ then\ v, if\ (Backward)\ then - v, else\ 0 \quad (6)$$

$$i = if\ (Forward)\ then\ i_{pack}, if\ (Backward)\ then - i_{pack}, else\ 0 \quad (7)$$

*Benoît Caillaud, Mathias Malandain, Joan Thibault. Implicit structural analysis of multimode DAE systems. HSCC 2020 - 23rd ACM International Conference on Hybrid Systems: Computation and Control, Apr 2020, Sydney New South Wales Australia, France. pp.1-11, ⟨10.1145/3365365.3382201⟩.

Problem Formulation/Challenges

Even for a small system:

6 battery cells in one pack → Each cell has 4 modes → $4^6 = 4096$ combinations for switch configuration.

1. How to do diagnosis analysis for multi-mode systems with high number of components and configurations? (contribution 1)

A general solution

2. How to model faults? Pros and cons? (contribution 2)

F. Hashemniya, A. Balachandran, K. Mattias, and E. Frisk. Structural diagnosability analysis of switched and modular battery packs. In Prognostics and System Health Management Conference (PHM 2024), Stockholm, Sweden. IEEE,

Direct Fault Isolability Analysis

Classic Dulmage-Mendelsohn Decomposition

M: Set of equations
X: Set of unknown variables

M^- : underdetermined part
M^0 : just determined part
M^+ : overdetermined part

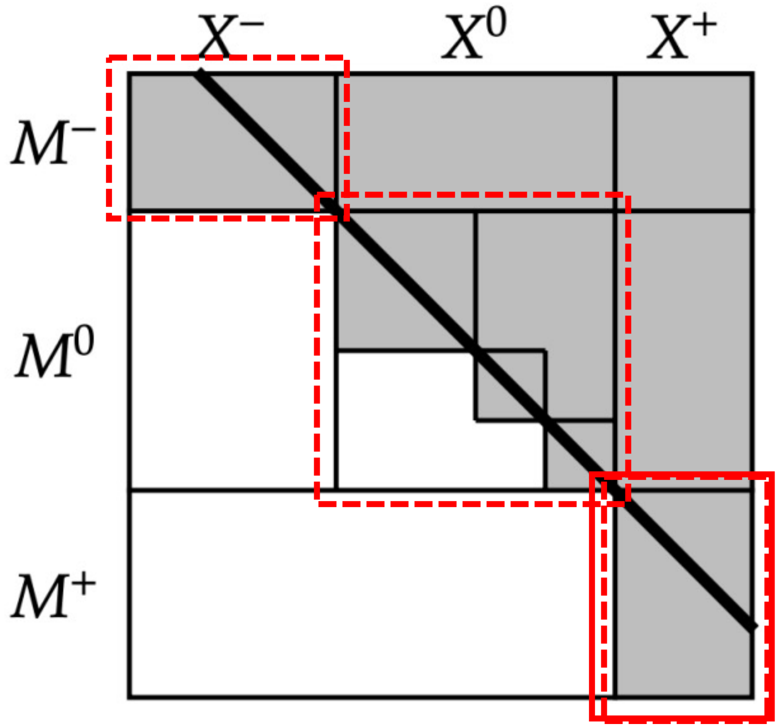
M^- : underdetermined part

M^+: Number of equations is more than unknown variables

M^0 : just determined part

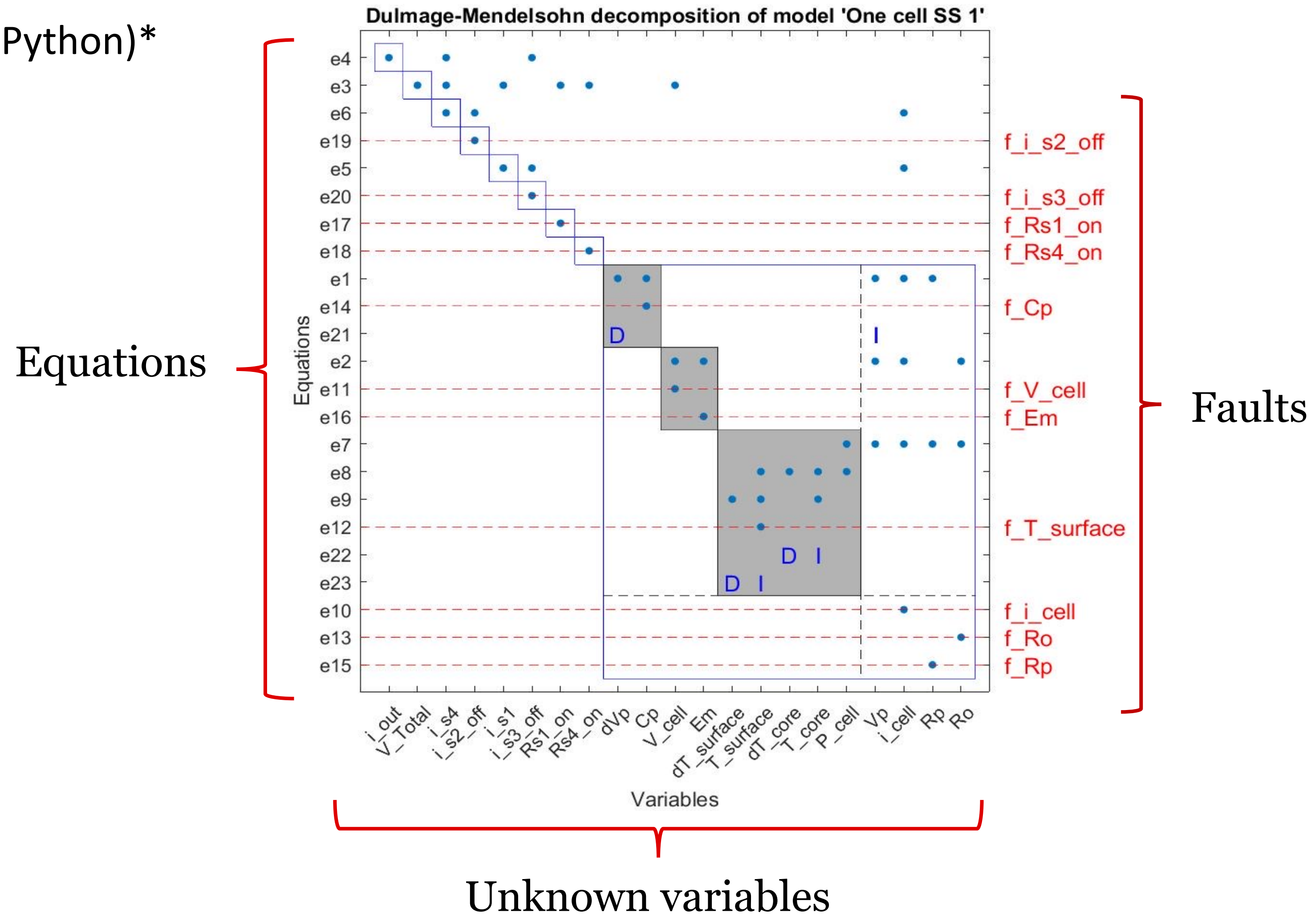
Here, there is redundancy that will be used for diagnosis and residual generation.

M^+ : overdetermined part



Extended DM Decomposition

Extended DM:
Toolbox (Matlab and Python)*



* E. Frisk, M. Krysander, and D. Jung. A toolbox for analysis and design of model based diagnosis systems for large scale models. IFAC-PapersOnLine, 50(1):3287– 3293, 2017.

Extended DM Decomposition

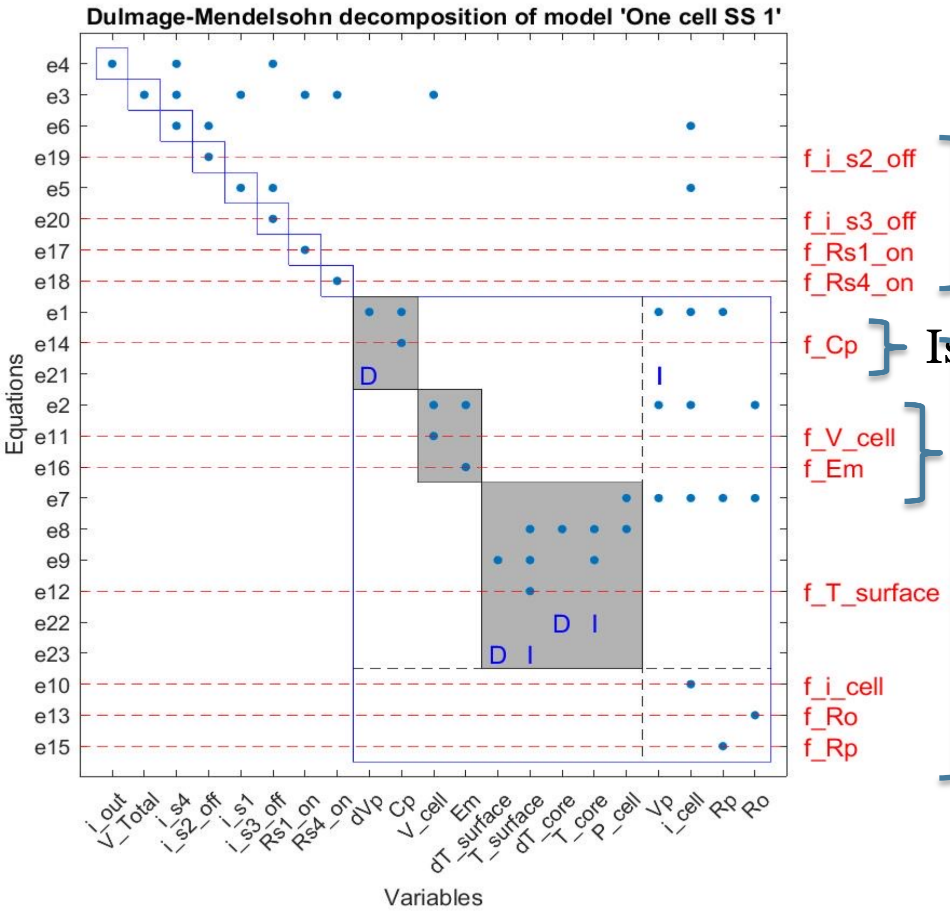
Extended DM: Diagnosability

Detectability:

Fault f is detectable if it is in the overdetermined part.

Isolability:

A fault f_i is structurally isolable from f_j in a model M if $e_{f_i} \in (\mathrm{M} \setminus \{e_{f_j}\})^+$.



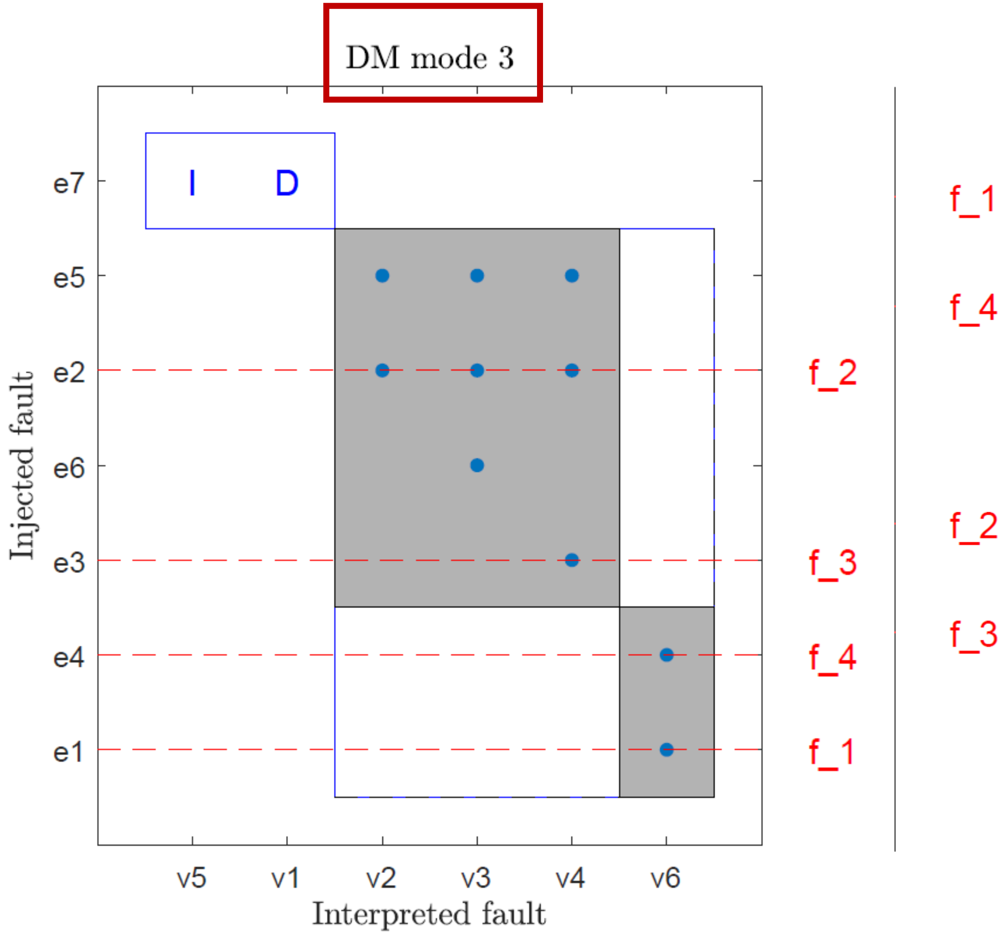
Not Detectable

Isolable from other faults

Not isolable from each other

Detectable

Multi-mode DM Decomposition

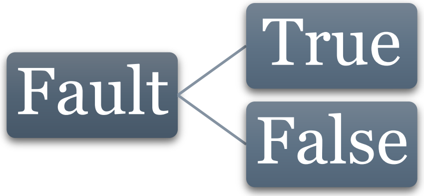



```
e1_under: mode2
e1_det:   mode1
e1_over:  mode3
```

```
e4_under: False
e4_det:   mode2
e4_over:  mode1 ^ mode3
```


Albert Benveniste, Benoît Caillaud, Mathias Malandain, Joan Thibault. Algorithms for the Structural Analysis of Multimode Modelica Models. Electronics, 2022, 11 (17), pp.1-63.

Fault Modeling



Fault as Signal

```
e_1:    X_1 = X_2

e_f1:   X_3 = 5X_2 + f1
```

```
e_f1_under:  False
e_f1_det:    mode1 ^ mode2
e_f1_over:   ¬mode3
```

Fault as Boolean mode variables

```
                         e_1:   X_1 = X_2

If ¬F1    then           e_f1:  X_3 = 5X_2
```

```
e_f1_under:  False
e_f1_det:    mode1 ^ mode2 ^ ¬F1
e_f1_over:   ¬mode3 ^ ¬F1 ^ ¬F3
```

Multi-mode Isolability Matrix

Single – mode Fault Isolability Matrix

	f_{cell}	f_v	f_i
f_{cell}	0	0	0
f_v	0	0	0
f_i	0	0	0

Mode 1

	f_{cell}	f_v	f_i
f_{cell}	0	0	1
f_v	0	0	1
f_i	1	1	0

Mode 2

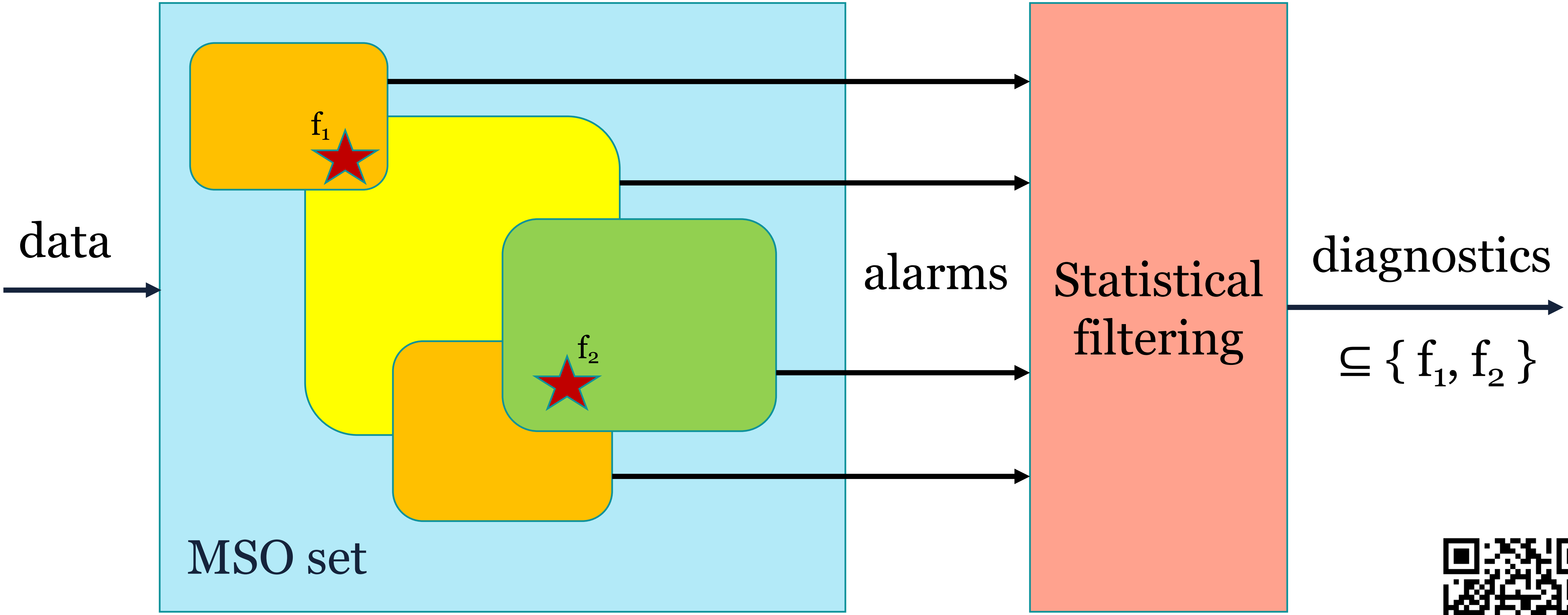
*Multi – mode ⇒ for All modes**

	NF	f_{cell}	f_v	f_i
f_{cell}	1	0	0	$m \in \{2,4\}$
f_v	1	0	0	$m \in \{2,4\}$
f_i	1	$m \in \{2,4\}$	$m \in \{2,4\}$	0

Table 3. Multi-mode fault diagnosability.

	NF	$f_{\text{cell},k}$	$f_{v_{\text{cell},k}}$	$f_{i_{\text{cell},k}}$	$f_{i_{\text{pack}}}$	$f_{v_{\text{pack}}}$
$f_{\text{cell},k}$	**T**	**F**	$\neg\text{bypass}_k$	**T**	**T**	**T**
$f_{v_{\text{cell},k}}$	**T**	$\neg\text{bypass}_k$	**F**	**T**	**T**	**T**
$f_{i_{\text{cell},k}}$	**T**	**T**	**T**	**F**	**T**	**T**
$f_{i_{\text{pack}}}$	$\neg\text{bypass}_{\text{all}}$	$\neg\text{bypass}_{\text{all}}$	$\neg\text{bypass}_{\text{all}}$	$\neg\text{bypass}_{\text{all}}$	**F**	$\neg\text{bypass}_{\text{all}}$
$f_{v_{\text{pack}}}$	**T**	**T**	**T**	**T**	**T**	**F**

Alarm filtering & correlation at runtime



Jung, Frisk, Krysander. *Quantitative Stochastic Fault Diagnosability Analysis*. Control Engineering Practice, 2018.

Conclusion and Future Work

- A methodology for conducting structural fault diagnostics for multi-mode systems
  - Using a multi-mode extension of the Dulmage-Mendelsohn decomposition
- 2 types of fault modeling with diagnosability definitions
  - Signal: better for large systems, but small number of faults
  - Boolean variable: relevant for higher number of faults and multiple simultaneous faults
- Complexity study on a battery pack with n SMs in series
  - Scales up to about 10 cells ; definitely does not scale up to a full pack

Future Work

- Generalizing our approach to more complex multi-mode models
  - Some faults can only occur in some modes
- Algorithmic improvements of mmDM
  - Improve computation time : modular algorithm, based on message passing principles †

† Benveniste, Caillaud, Malandain, Thibault. *Towards the separate compilation of Modelica: modularity and interfaces for the index reduction of incomplete DAE systems*. Modelica'23

Thank You!

Fatemeh.Hashemniya@liu.se

Benoit.Caillaud@inria.fr

Erik.Frisk@liu.se

Mattias.Krysander@liu.se

Mathias.Malandain@inria.fr

LINKÖPING UNIVERSITY

Inria

A Multimode Dulmage-Mendelsohn Decomposition Algorithm

Benoît Caillaud Mathias Malandain

November 18th, 2024

Implicit representation of the structure of a multimode DAE system

A multimode Dulmage-Mendelsohn decomposition algorithm

A compositional approach: the CoSTreD method

Focus on the ArgMax algorithm

Encoding a model (in a nutshell)

- Everything is encoded as functions of the mode variables
- BDDs (Binary Decision Diagrams) are an appropriate framework:
    - Compact and canonical representation of Boolean functions as DAGs
    - Efficient computations on such functions
    - Integer functions: variable-length little-endian binary encoding (list of BDDs)
- Negation $\neg$ and equality check in $\mathcal{O}(1)$, other operations include:

    Conjunction/disjunction: $\wedge / \vee$

    Existential quantification: $\exists a.\, f(a, b)$

    Universal quantification: $\forall a.\, f(a, b)$
- However: very sensitive to variable and computation ordering

Encoding a model (in a nutshell)

- Everything is encoded as functions of the mode variables
- BDDs (Binary Decision Diagrams) are an appropriate framework:
  - Compact and canonical representation of Boolean functions as DAGs
  - Efficient computations on such functions
  - Integer functions: variable-length little-endian binary encoding (list of BDDs)
- Negation $\neg$ and equality check in $\mathcal{O}(1)$, other operations include:
  - Conjunction/disjunction: $\wedge / \vee$
  - Existential quantification: $\exists a.\, f(a, b)$
  - Universal quantification: $\forall a.\, f(a, b)$
- However: very sensitive to variable and computation ordering

Encoding a model (in a nutshell)

- Everything is encoded as functions of the mode variables
- BDDs (Binary Decision Diagrams) are an appropriate framework:
  - Compact and canonical representation of Boolean functions as DAGs
  - Efficient computations on such functions
  - Integer functions: variable-length little-endian binary encoding (list of BDDs)
- Negation $\neg$ and equality check in $\mathcal{O}(1)$, other operations include:

  Conjunction/disjunction: $\wedge/\vee$

  Existential quantification: $\exists a.\, f(a, b)$

  Universal quantification: $\forall a.\, f(a, b)$

- However: very sensitive to variable and computation ordering

Encoding a model (in a nutshell)

- Everything is encoded as functions of the mode variables
- BDDs (Binary Decision Diagrams) are an appropriate framework:
  - Compact and canonical representation of Boolean functions as DAGs
  - Efficient computations on such functions
  - Integer functions: variable-length little-endian binary encoding (list of BDDs)
- Negation $\neg$ and equality check in $\mathcal{O}(1)$, other operations include:

  Conjunction/disjunction: $\wedge/\vee$

  Existential quantification: $\exists a.\, f(a, b)$

  Universal quantification: $\forall a.\, f(a, b)$
- However: very sensitive to variable and computation ordering

Useful sets

- M encodes the modes (model-dependent encoding)
- $I = \bigcup_{m \in M} I_m$ encodes the equations (unary representation)
- $J = \bigcup_{m \in M} J_m$ encodes the variables (unary representation)
- $E = \bigcup_{m \in M} E_m$ encodes the equation-variable edges (unary representation)

(Unary encoding of a set S provides access to $\mathcal{P}(S)$.)

Encoding the problem data

The following data can be obtained by parsing the original model:

Name	Type	Meaning
χ_M	$M \to \mathbb{B}$	Set of possible modes
χ_I	$M \times I \to \mathbb{B}$	Mode dependency of equations
χ_J	$M \times J \to \mathbb{B}$	Mode dependency of variables
χ_E	$M \times E \to \mathbb{B}$	Mode dependency of edges
σ	$M \times E \to \mathbb{N}$	Values of the $\sigma_{m,i,j}$'s

Implicit representation of the structure of a multimode DAE system

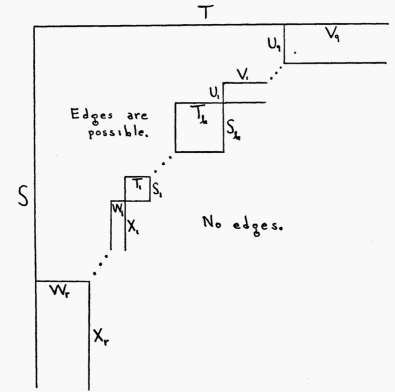
A multimode Dulmage-Mendelsohn decomposition algorithm

A compositional approach: the CoSTreD method

Focus on the ArgMax algorithm

The Dulmage-Mendelsohn decomposition

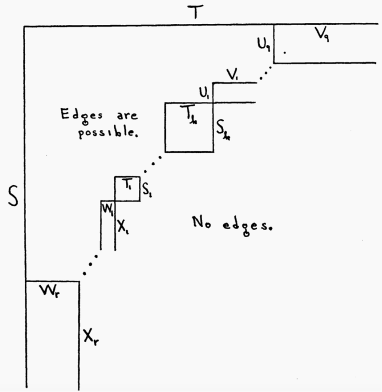
- Canonical decomposition of a bipartite graph $(S \cup T, E)$
- Applied to a system of algebraic equations, yields a partition into three subsystems:
  - I_u: underdetermined part (variables in J_u)
  - I_e: 'enabled' (square) part (variables in J_e)
  - I_o: overdetermined part (variables in J_o)
- Uses: over-/underdetermination diagnostics (model debugging), health monitoring



Matrix representation [*Dulmage & Mendelsohn, 'Two Algorithms for Bipartite Graphs', 1963]*

The Dulmage-Mendelsohn decomposition

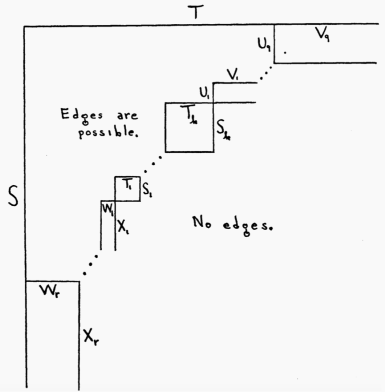
- Canonical decomposition of a bipartite graph $(S \cup T, E)$
- Applied to a system of algebraic equations, yields a partition into three subsystems:
  - I_u: underdetermined part (variables in J_u)
  - I_e: 'enabled' (square) part (variables in J_e)
  - I_o: overdetermined part (variables in J_o)
- Uses: over-/underdetermination diagnostics (model debugging), health monitoring



Matrix representation [*Dulmage & Mendelsohn, 'Two Algorithms for Bipartite Graphs', 1963]*

The Dulmage-Mendelsohn decomposition

- Canonical decomposition of a bipartite graph $(S \cup T, E)$
- Applied to a system of algebraic equations, yields a partition into three subsystems:
    - I_u: underdetermined part (variables in J_u)
    - I_e: 'enabled' (square) part (variables in J_e)
    - I_o: overdetermined part (variables in J_o)
- Uses: over-/underdetermination diagnostics (model debugging), health monitoring



Matrix representation [*Dulmage & Mendelsohn, 'Two Algorithms for Bipartite Graphs', 1963*]

A graph-based algorithm for DM

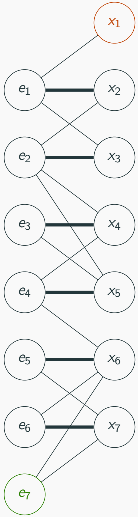
- Requires a maximum matching $\mathcal{M}$ of the system's adjacency graph
- Define I^u (resp. J^u): set of unmatched equations $i \in I$ (resp. variables $j \in J$); then:
  - $I_\mathfrak{o}$, $J_\mathfrak{o}$: reachable via an alternating path from I^u
  - $I_\mathfrak{u}$, $J_\mathfrak{u}$: reachable via an alternating path from J^u
  - $I_\mathfrak{e}$, $J_\mathfrak{e}$: remaining equations and variables
- The decomposition does not depend on the choice of a maximal matching.

A graph-based algorithm for DM

- Requires a maximum matching $\mathcal{M}$ of the system's adjacency graph
- Define I^u (resp. J^u): set of unmatched equations $i \in I$ (resp. variables $j \in J$); then:
  - $I_{\mathfrak{o}}, J_{\mathfrak{o}}$: reachable via an alternating path from I^u
  - $I_{\mathfrak{u}}, J_{\mathfrak{u}}$: reachable via an alternating path from J^u
  - $I_{\mathfrak{e}}, J_{\mathfrak{e}}$: remaining equations and variables
- The decomposition does not depend on the choice of a maximal matching.

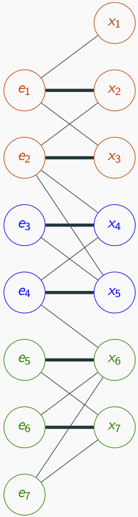
A graph-based algorithm for DM

- Requires a maximum matching $\mathcal{M}$ of the system's adjacency graph
- Define I^u (resp. J^u): set of unmatched equations $i \in I$ (resp. variables $j \in J$); then:
  - $I_{\mathfrak{o}}$, $J_{\mathfrak{o}}$: reachable via an alternating path from I^u
  - $I_{\mathfrak{u}}$, $J_{\mathfrak{u}}$: reachable via an alternating path from J^u
  - $I_{\mathfrak{e}}$, $J_{\mathfrak{e}}$: remaining equations and variables
- The decomposition does not depend on the choice of a maximal matching.



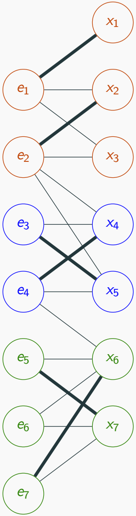
A graph-based algorithm for DM

- Requires a maximum matching $\mathcal{M}$ of the system's adjacency graph
- Define I^u (resp. J^u): set of unmatched equations $i \in I$ (resp. variables $j \in J$); then:
  - $I_{\mathfrak{o}}$, $J_{\mathfrak{o}}$: reachable via an alternating path from I^u
  - $I_{\mathfrak{u}}$, $J_{\mathfrak{u}}$: reachable via an alternating path from J^u
  - $I_{\mathfrak{e}}$, $J_{\mathfrak{e}}$: remaining equations and variables
- The decomposition does not depend on the choice of a maximal matching.



A graph-based algorithm for DM

- Requires a maximum matching $\mathcal{M}$ of the system's adjacency graph
- Define I^u (resp. J^u): set of unmatched equations $i \in I$ (resp. variables $j \in J$); then:
    - $I_{\mathfrak{o}}$, $J_{\mathfrak{o}}$: reachable via an alternating path from I^u
    - $I_{\mathfrak{u}}$, $J_{\mathfrak{u}}$: reachable via an alternating path from J^u
    - $I_{\mathfrak{e}}$, $J_{\mathfrak{e}}$: remaining equations and variables
- The decomposition does not depend on the choice of a maximal matching.



Multimode adaptation: computing all partial matchings

Already existing: description of all perfect matchings

Modified: description of all matchings

$\mu : M \times \mathcal{P}(E) \to \mathbb{B}$ (each equation is connected to one and only one variable)

$\nu : M \times \mathcal{P}(E) \to \mathbb{B}$ (each variable is connected to one and only one equation)

$\Upsilon : M \times \mathcal{P}(E) \to \mathbb{B}$ (an edge must be active to be part of a matching)

$X := \mu \wedge \nu \wedge \Upsilon$ (all perfect matchings)

Multimode adaptation: computing all partial matchings

Already existing: description of all perfect matchings

Modified: description of all matchings

$\mu : M \times \mathcal{P}(E) \to \mathbb{B}$ (each equation is connected to at most one variable)

$\nu : M \times \mathcal{P}(E) \to \mathbb{B}$ (each variable is connected to at most one equation)

$\Upsilon : M \times \mathcal{P}(E) \to \mathbb{B}$ (an edge must be active to be part of a matching)

$X := \mu \wedge \nu \wedge \Upsilon$ (all partial matchings)

Multimode adaptation: picking one maximum matching per mode

- ArgMax and choice algorithms already implemented (by induction on the BDD structure)
  - ArgMax needs a weight function $\omega : M \times \mathcal{P}(E) \to \mathbb{N}$ as input
- Create ω such that every edge has weight 1
  - Results in the choice of one maximum matching per mode
- Finally: compute one characteristic function $T_e : M \to \mathbb{B}$ per edge (in which modes is this edge in the chosen matching?)
  - Already implemented specialized inductive algorithm

Multimode adaptation: picking one maximum matching per mode

- ArgMax and choice algorithms already implemented (by induction on the BDD structure)
  - ArgMax needs a weight function $\omega : M \times \mathcal{P}(E) \to \mathbb{N}$ as input
- Create ω such that every edge has weight 1
  - Results in the choice of one maximum matching per mode
- Finally: compute one characteristic function $T_e : M \to \mathbb{B}$ per edge (in which modes is this edge in the chosen matching?)
  - Already implemented specialized inductive algorithm

Multimode adaptation: picking one maximum matching per mode

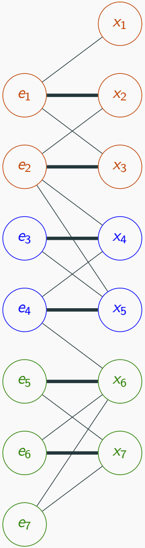
- ArgMax and choice algorithms already implemented (by induction on the BDD structure)
  - ArgMax needs a weight function $\omega : M \times \mathcal{P}(E) \to \mathbb{N}$ as input
- Create ω such that every edge has weight 1
  - Results in the choice of one maximum matching per mode
- Finally: compute one characteristic function $T_e : M \to \mathbb{B}$ per edge (in which modes is this edge in the chosen matching?)
  - Already implemented specialized inductive algorithm

Multimode adaptation: the Dulmage-Mendelsohn algorithm

- Important observation: in a given 'propagation step', either only edges in $\mathcal{M}$ are followed, or only edges outside $\mathcal{M}$
- Easy to translate in the multimode setting:
  - Three functions $o_i, u_i, e_i : M \to \mathbb{B}$ for every equation i; for instance, initial value of o_i (unmatched eqs):

$$\neg\left(\bigvee_{e \in \mathcal{I}^{-1}(i)} T_e\right) \wedge \chi_I(\cdot, i)$$

  - Same idea for every variable j
  - Simple iterations until the over- and under- determined parts are known (fixpoint)

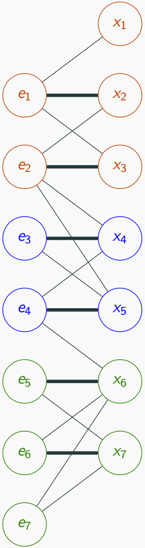


Multimode adaptation: the Dulmage-Mendelsohn algorithm

- Important observation: in a given 'propagation step', either only edges in $\mathcal{M}$ are followed, or only edges outside $\mathcal{M}$
- Easy to translate in the multimode setting:
  - Three functions $\mathfrak{o}_i$, $\mathfrak{u}_i$, $\mathfrak{e}_i : M \to \mathbb{B}$ for every equation i; for instance, initial value of $\mathfrak{o}_i$ (unmatched eqs):

$$\neg\left(\bigvee_{e\in\mathcal{I}^{-1}(i)} T_e\right) \wedge \chi_I(\cdot, i)$$

  - Same idea for every variable j
  - Simple iterations until the over- and under- determined parts are known (fixpoint)

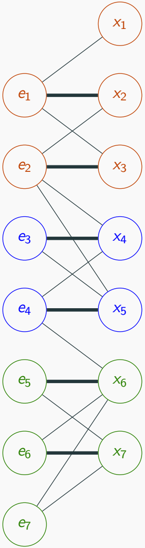


Multimode adaptation: the Dulmage-Mendelsohn algorithm

- Important observation: in a given 'propagation step', either only edges in $\mathcal{M}$ are followed, or only edges outside $\mathcal{M}$
- Easy to translate in the multimode setting:
  - Three functions $\mathfrak{o}_i$, $\mathfrak{u}_i$, $\mathfrak{e}_i : M \to \mathbb{B}$ for every equation i; for instance, initial value of $\mathfrak{o}_i$ (unmatched eqs):

$$\neg \left(\bigvee_{e \in \mathcal{I}^{-1}(i)} T_e \right) \wedge \chi_I(\cdot, i)$$

  - Same idea for every variable j
  - Simple iterations until the over- and under- determined parts are known (fixpoint)

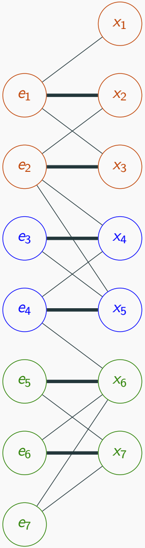


Multimode adaptation: the Dulmage-Mendelsohn algorithm

- Important observation: in a given 'propagation step', either only edges in $\mathcal{M}$ are followed, or only edges outside $\mathcal{M}$
- Easy to translate in the multimode setting:
  - Three functions $\mathfrak{o}_i$, $\mathfrak{u}_i$, $\mathfrak{e}_i : M \to \mathbb{B}$ for every equation i; for instance, initial value of $\mathfrak{o}_i$ (unmatched eqs):

$$\neg \left(\bigvee_{e \in \mathcal{I}^{-1}(i)} T_e \right) \wedge \chi_I(\cdot, i)$$

  - Same idea for every variable j
  - Simple iterations until the over- and under- determined parts are known (fixpoint)

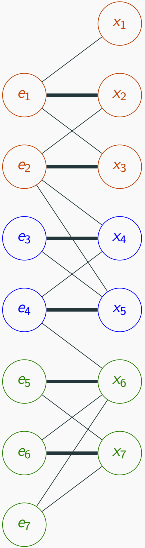


Multimode adaptation: the Dulmage-Mendelsohn algorithm

- Important observation: in a given ‘propagation step’, either only edges in $\mathcal{M}$ are followed, or only edges outside $\mathcal{M}$
- Easy to translate in the multimode setting:
  - Three functions $\mathfrak{o}_i$, $\mathfrak{u}_i$, $\mathfrak{e}_i : M \to \mathbb{B}$ for every equation i; for instance, initial value of $\mathfrak{o}_i$ (unmatched eqs):

$$\neg \left(\bigvee_{e \in \mathcal{I}^{-1}(i)} T_e \right) \wedge \chi_I(\cdot, i)$$

  - Same idea for every variable j
  - Simple iterations until the over- and under- determined parts are known (fixpoint)



Implicit representation of the structure of a multimode DAE system

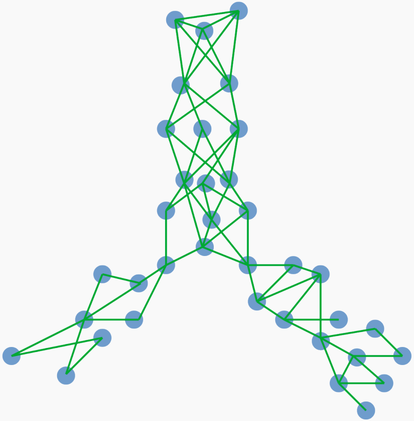
A multimode Dulmage-Mendelsohn decomposition algorithm

A compositional approach: the CoSTreD method

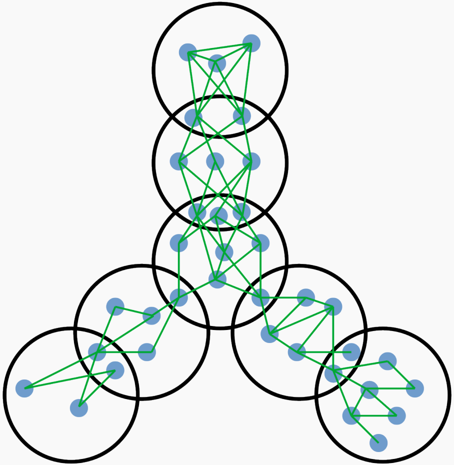
Focus on the ArgMax algorithm

Principle of CoSTreD: Decomposing a constraint problem

Primal graph of the constraint system

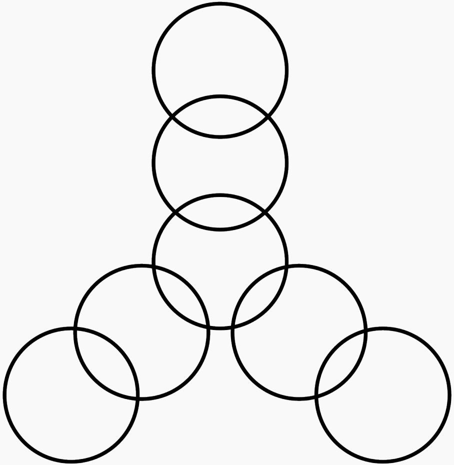


Use sparsity and low tree width to compute a decomposition



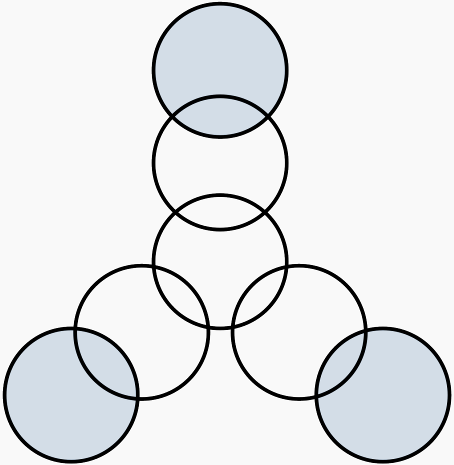
Principle of CoSTreD: Decomposing a constraint problem

Use sparsity and low tree width to compute de decomposition



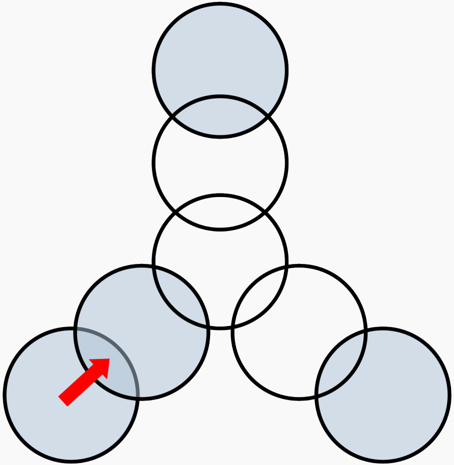
Principle of CoSTreD: Forward Reduction Process

project local constraints on shared variables



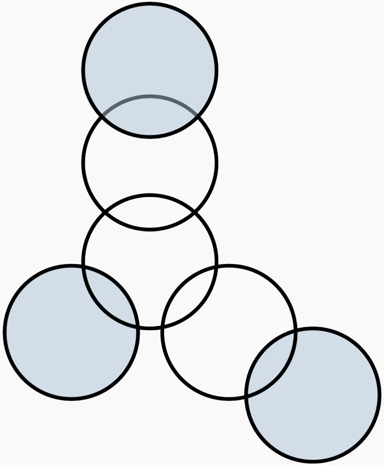
Principle of CoSTreD: Forward Reduction Process

Pass to parent and combine with local constraints



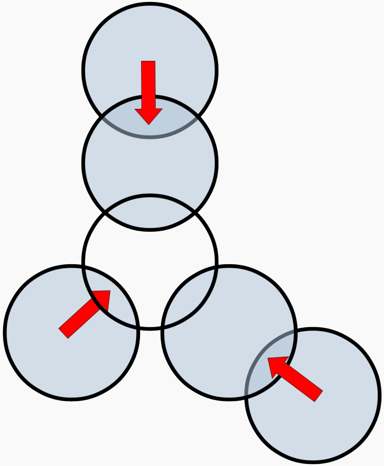
Principle of CoSTreD: Forward Reduction Process

project resulting constraints on shared variables



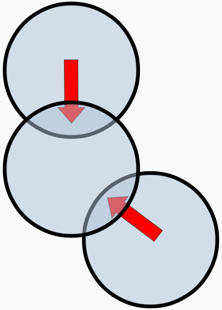
Principle of CoSTreD: Forward Reduction Process

Pass to parent and combine with local constraints



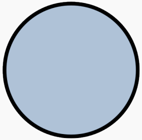
Principle of CoSTreD: Forward Reduction Process

... and so on, up to the tree root



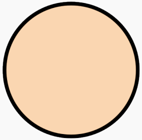
Principle of CoSTreD: Forward Reduction Process

... and so on, up to the tree root



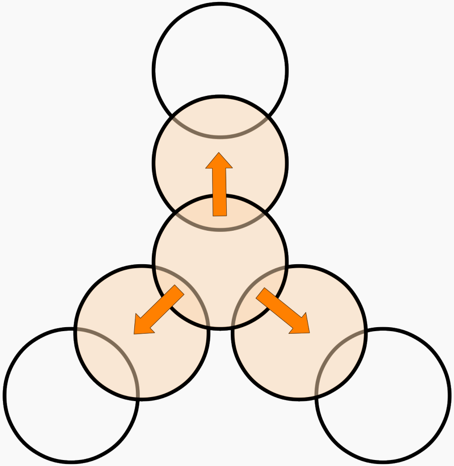
Principle of CoSTreD: Backward Propagation Process

Compute partial solution at the root



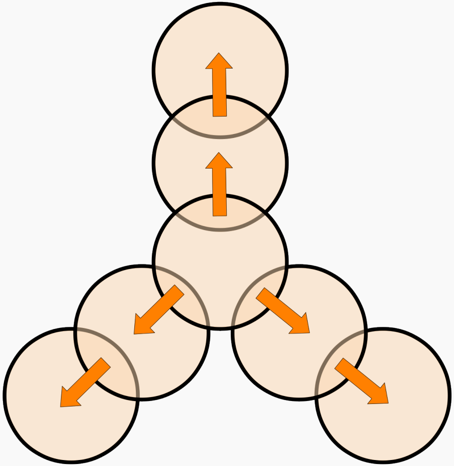
Principle of CoSTreD: Backward Propagation Process

Pass solution down the tree, combine with local constraints and compute partial solution

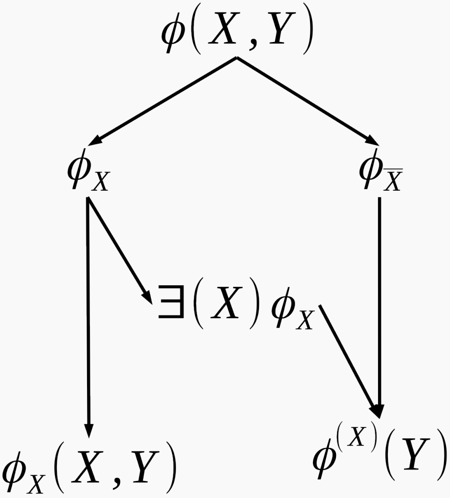


Principle of CoSTreD: Backward Propagation Process

... and so on, down to the leaves

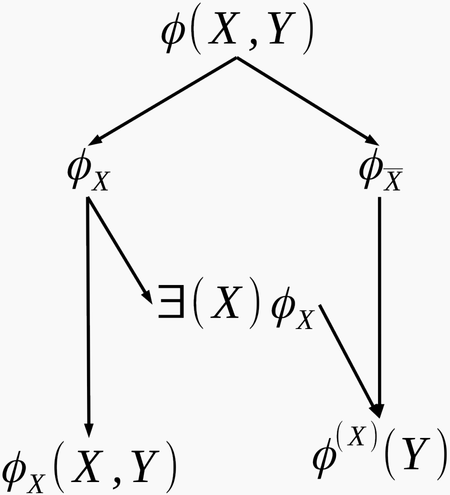


Application to mmDM: ArgMax(Boolean constraints *wedge* weight function)



- CoSTreD exploits the sparsity and low tree width of physical models
- Cost model for computation time and memory consumption
  - ⇒ graphical optimization problem WAP
  - ⇒ strongly related to treewidth (NP-complete)
- Naive but effective solving: greedy heuristics (min-degree)
- CoSTreD can also be used for MaxSAT and QBF problems

Application to mmDM: ArgMax(Boolean constraints *wedge* weight function)



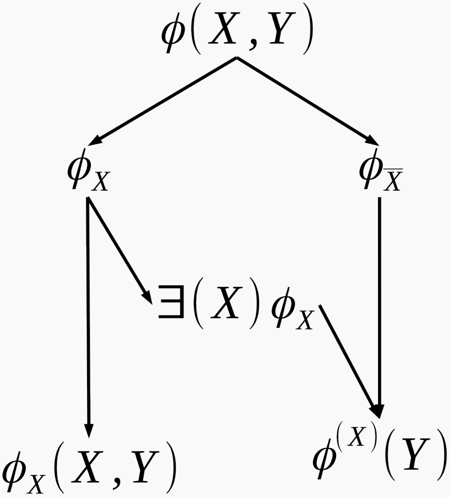
- CoSTreD exploits the sparsity and low tree width of physical models
- Cost model for computation time and memory consumption
  - $\Rightarrow$ graphical optimization problem WAP
  - $\Rightarrow$ strongly related to treewidth (NP-complete)
- Naive but effective solving: greedy heuristics (min-degree)
- CoSTreD can also be used for MaxSAT and QBF problems

Application to mmDM: ArgMax(Boolean constraints *wedge* weight function)

$$
\begin{tikzcd}
 & \phi(X,Y) \arrow[dl] \arrow[dr] & \\
\phi_X \arrow[dd] \arrow[dr] & & \phi_{\overline{X}} \arrow[dd] \\
 & \exists(X)\,\phi_X \arrow[dr] & \\
\phi_X(X,Y) & & \phi^{(X)}(Y)
\end{tikzcd}
$$

- CoSTreD exploits the sparsity and low tree width of physical models
- Cost model for computation time and memory consumption
  - $\Rightarrow$ graphical optimization problem WAP
  - $\Rightarrow$ strongly related to treewidth (NP-complete)
- Naive but effective solving: greedy heuristics (min-degree)
- CoSTreD can also be used for MaxSAT and QBF problems

Application to mmDM: ArgMax(Boolean constraints *wedge* weight function)



- CoSTreD exploits the sparsity and low tree width of physical models
- Cost model for computation time and memory consumption
  - $\Rightarrow$ graphical optimization problem WAP
  - $\Rightarrow$ strongly related to treewidth (NP-complete)
- Naive but effective solving: greedy heuristics (min-degree)
- CoSTreD can also be used for MaxSAT and QBF problems

Implicit representation of the structure of a multimode DAE system

A multimode Dulmage-Mendelsohn decomposition algorithm

A compositional approach: the CoSTreD method

Focus on the ArgMax algorithm

The ArgMax algorithm for maximum-weight matchings

- Input:
  - $X : M \times E \to \mathbb{B}$ (all perfect matchings)
  - $(\omega_k : M \times E \to \mathbb{B})_{k=0\ldots N-1}$ (matching weights in little-endian representation)
- Let $S_N \equiv X$; for $k = N - 1$ down to 0:
  - $\omega_k^{\max} \equiv \exists \vec{E}.(S_{k+1} \wedge \omega_k)$ $\quad (\omega_k^{\max} : M \to \mathbb{B})$
  - $S_k \equiv S_{k+1} \wedge (\omega_k \Leftrightarrow \omega_k^{\max})$
- Output:
  - $S_0 : M \times E \to \mathbb{B}$ represents all maximum-weight perfect matchings
- ArgMax may also be used for selecting one matching in every mode from S_0: replace $(\omega_k)_{k=0\ldots N-1}$ with $(\chi_E(e))_{e \in E}$ to get a lexicographical order

The ArgMax algorithm for maximum-weight matchings

- Input:
  - $X : M \times E \to \mathbb{B}$ (all perfect matchings)
  - $(\omega_k : M \times E \to \mathbb{B})_{k=0\ldots N-1}$ (matching weights in little-endian representation)
- Let $S_N \equiv X$; for $k = N-1$ down to 0:
  - $\omega_k^{\max} \equiv \exists \vec{E}.(S_{k+1} \wedge \omega_k)$ $\quad (\omega_k^{\max} : M \to \mathbb{B})$
  - $S_k \equiv S_{k+1} \wedge (\omega_k \Leftrightarrow \omega_k^{\max})$
- Output:
  - $S_0 : M \times E \to \mathbb{B}$ represents all maximum-weight perfect matchings
- ArgMax may also be used for selecting one matching in every mode from S_0: replace $(\omega_k)_{k=0\ldots N-1}$ with $(\chi_E(e))_{e\in E}$ to get a lexicographical order

The ArgMax algorithm for maximum-weight matchings

- Input:
  - $X : M \times E \to \mathbb{B}$ (all perfect matchings)
  - $(\omega_k : M \times E \to \mathbb{B})_{k=0\ldots N-1}$ (matching weights in little-endian representation)
- Let $S_N \equiv X$; for $k = N - 1$ down to 0:
  - $\omega_k^{\max} \equiv \exists \vec{E}.(S_{k+1} \wedge \omega_k)$ $\qquad$ $(\omega_k^{\max} : M \to \mathbb{B})$
  - $S_k \equiv S_{k+1} \wedge (\omega_k \Leftrightarrow \omega_k^{\max})$
- Output:
  - $S_0 : M \times E \to \mathbb{B}$ represents all maximum-weight perfect matchings
- ArgMax may also be used for selecting one matching in every mode from S_0: replace $(\omega_k)_{k=0\ldots N-1}$ with $(\chi_E(e))_{e \in E}$ to get a lexicographical order

The ArgMax algorithm for maximum-weight matchings

- Input:
  - $X : M \times E \to \mathbb{B}$ (all perfect matchings)
  - $(\omega_k : M \times E \to \mathbb{B})_{k=0\ldots N-1}$ (matching weights in little-endian representation)
- Let $S_N \equiv X$; for $k = N - 1$ down to 0:
  - $\omega_k^{\max} \equiv \exists \vec{E}.(S_{k+1} \wedge \omega_k)$ $\qquad (\omega_k^{\max} : M \to \mathbb{B})$
  - $S_k \equiv S_{k+1} \wedge (\omega_k \Leftrightarrow \omega_k^{\max})$
- Output:
  - $S_0 : M \times E \to \mathbb{B}$ represents all maximum-weight perfect matchings
- ArgMax may also be used for selecting one matching in every mode from S_0: replace $(\omega_k)_{k=0\ldots N-1}$ with $(\chi_E(e))_{e\in E}$ to get a lexicographical order

The ArgMax algorithm for maximum-weight matchings

- Input:
  - $X : M \times E \to \mathbb{B}$ (all perfect matchings)
  - $(\omega_k : M \times E \to \mathbb{B})_{k=0\ldots N-1}$ (matching weights in little-endian representation)
- Let $S_N \equiv X$; for $k = N - 1$ down to 0:
  - $\omega_k^{\max} \equiv \exists \vec{E}.(S_{k+1} \wedge \omega_k)$ $\qquad (\omega_k^{\max} : M \to \mathbb{B})$
  - $S_k \equiv S_{k+1} \wedge (\omega_k \Leftrightarrow \omega_k^{\max})$
- Output:
  - $S_0 : M \times E \to \mathbb{B}$ represents all maximum-weight perfect matchings
- ArgMax may also be used for selecting one matching in every mode from S_0: replace $(\omega_k)_{k=0\ldots N-1}$ with $(\chi_E(e))_{e \in E}$ to get a lexicographical order

The ArgMax algorithm for maximum-weight matchings

- Input:
  - $X : M \times E \to \mathbb{B}$ (all perfect matchings)
  - $(\omega_k : M \times E \to \mathbb{B})_{k=0\ldots N-1}$ (matching weights in little-endian representation)
- Let $S_N \equiv X$; for $k = N - 1$ down to 0:
  - $\omega_k^{\max} \equiv \exists \vec{E}.(S_{k+1} \wedge \omega_k)$ $\quad (\omega_k^{\max} : M \to \mathbb{B})$
  - $S_k \equiv S_{k+1} \wedge (\omega_k \Leftrightarrow \omega_k^{\max})$
- Output:
  - $S_0 : M \times E \to \mathbb{B}$ represents all maximum-weight perfect matchings
- ArgMax may also be used for selecting one matching in every mode from S_0: replace $(\omega_k)_{k=0\ldots N-1}$ with $(\chi_E(e))_{e\in E}$ to get a lexicographical order