Efficient diagnosability assessment via ILP optimization: a railway benchmark

23rd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2018)

F. Basile[1], A. Boussif[2], Gianmaria De Tommasi[3], M. Ghazel[2], C. Sterle[3]

[1]Università degli Studi di Salerno, Italy

[2]IFSTTAR, University of Lille Nord, France

[3]Università degli Studi di Napoli Federico II, Italy

Torino - 6 September 2018

DIE
TI.

UNIVERSITA' DEGLI STUDI DI NAPOLI FEDERICO II

DIPARTIMENTO DI INGEGNERIA ELETTRICA
E DELLE TECNOLOGIE DELL'INFORMAZIONE

Outline

1. Preliminaries
   - Diagnosability in the Petri nets context
   - Motivation & Contribution
   - Notation & Definitions
2. The railway benchmark
3. Diagnosability approaches
   - Optimization-based approach
   - Graph-based approach
4. Numerical experiments
5. Conclusions

Diagnosability in the DES framework

- Fault detection and diagnosability have been studied in the Discrete Event Systems **(DES)** framework since early 90s
- The standard approach to check **diagnosability** is based on the **diagnoser** automata (see the seminal paper by *Sampath et al.*, IEEE TAC-1995)
- In the Petri nets **(PNs)** framework, a possible approach to fault diagnosis provides to associate the faults to unobservable transitions (unlabeled PNs) or events (labeled PNs)
- **A PN system is said to be *diagnosable* if every occurrence of an unobservable fault can be detected within a finite number of transition firings**

Diagnosability of Petri nets

- Different approaches for diagnosability have been proposed when DES are modelled with PNs
- A possible classification is the following
  - **graph-based** algorithms - analysis of reachability/coverability graphs or compact versions of them
    Jiroveanu and Boel, IEEE TAC-2010, *Cabasino et al.*, IEEE TAC-2012, *Boussif et al.*, VECoS-2015
  - **optimization-based** algorithms - the mathematical representation of PNs is exploited to assess diagnosability by solving Integer Linear Programming (ILP) problems)
    Basile et al., Automatica-2012, *Cong et al.*, IEEE TSMC-2017

Motivation

- Given the computational complexity of ILP problems, **the diagnosability conditions provided by *optimization-based* algorithms require the solution of NP-hard problems**
- **ILP programming is a standard optimization tool**
  - **it is possible to rely on efficient off-the-shelf optimization software tools**
  - CPLEX®
  - FICO™ Xpress
- **Despite their computational complexity, the *optimization-based* approaches can be practically more convenient when compared with the *graph-based* ones, which usually require ad hoc algorithms**

Contribution of this work

- **A comparison between a graph-based and an optimization-based algorithm is presented**

1. The optimization-based algorithm is taken from *Basile et al.*, Automatica-2012
2. The graph-based algorithm is taken from *Boussif et al.*, VECoS-2015

- The comparison is carried out using the *modular* railway benchmark presented in *Ghazel and Liu*, WODES-2016
- **Objective: efficiency assessment of the *optimization-based* algorithm 2 → The *graph-based* approach 1 was choosen since it outperforms other approaches on the considered benchmark (see *Boussif et al.*, DX-2017)**

PN notation

- $\mathcal{S} = \langle N, \boldsymbol{m}_0 \rangle$ is the net system, where $N = (P, T, \mathbf{Pre}, \mathbf{Post})$
- $T = T_o \cup T_{uo}$, and $T_f \subset T_{uo}$
- Given a **firing count vector** $\boldsymbol{\sigma} \in \mathbb{N}^n$, we would like to consider only firings of either observable or unobservable transitions. The following notation is introduced:

$$\boldsymbol{\sigma}_{|T_o} \in \mathbb{N}^n, \text{ with } \boldsymbol{\sigma}_{|T_o}(t) = \begin{cases} \boldsymbol{\sigma}(t) & \text{if } t \in T_o \\ 0 & \text{if} t \notin T_o \end{cases}$$

$$\boldsymbol{\sigma}_{|T_{uo}} \in \mathbb{N}^n, \text{ with } \boldsymbol{\sigma}_{|T_{uo}}(t) = \begin{cases} \boldsymbol{\sigma}(t) & \text{if } t \in T_{uo} \\ 0 & \text{if} t \notin T_{uo} \end{cases}$$

Labeled PNs

- $G = \langle N, \boldsymbol{m}_0, \lambda \rangle$ is a *labeled* Petri net (LPN) system
- $\lambda : T \mapsto E \cup \{\varepsilon\}$ is the *labeling function*
  - $\lambda(\cdot)$ assigns to each transition $t \in T$ either an event in E or the *silent event* ε
  - $\lambda(t) = \varepsilon$ if $t \in T_{uo}$, while $\lambda(t) \neq \varepsilon$ otherwise
- We denote with

$$T^{\alpha} = \left\{ t \in T \mid \lambda(t) = \alpha \right\},$$

  the set of transitions associated with the same event $\alpha \in E$.
- w denotes a word of events associated with a sequence σ such that $w = \lambda(\sigma)$
- $|w|$ denotes the length of w, while $|w|_{\alpha}$ denotes the number of occurrences of the event α in w

Diagnosability - Definition 1/3

- $L/u = \{v \in T^* \text{ s.t. } uv \in L\}$, is the post-language of L after the sequence of transitions u.
- $Pr : T^* \mapsto T_o^*$ is the usual projection that erases the unobservable transitions in a sequence u.
- The inverse projection operator Pr_L^{-1} is defined as

$$Pr_L^{-1}(r) = \{u \in L \text{ s.t. } Pr(u) = r\}$$

- Let $\dot{u}$ be the final transition of sequence u and define

$$\Psi(\hat{t}) = \left\{u \in L \text{ s.t. } \dot{u} = \hat{t}\right\}$$

Diagnosability - Definition 2/3

Definition (Diagnosable fault)

A fault transition $t_f \in T_f$ is said to be diagnosable if

$$\exists\ h \in \mathbb{N} \text{ such that } \forall\ u \in \Psi(t_f) \text{ and } \forall\ v \in L/u \text{ with } |v| \geq h\,,$$

it is

$$r \in Pr_L^{-1}\big(Pr(uv)\big) \Rightarrow t_f \in r\,.$$

Diagnosability - Definition 3/3

Definition ($\mathcal{K}$–diagnosable fault)

Given $t_f \in T_f$ and $\mathcal{K} \in \mathbb{N}$ (i.e., the maximum length of the postfix is given), t_f is said to be $\mathcal{K}$–diagnosable if

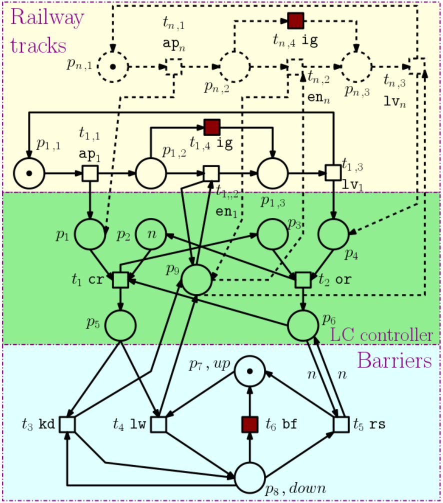
$$\forall\ u \in \Psi(t_f) \text{ and } \forall\ v \in L/u \text{ such that } |v| \geq \mathcal{K}\,,$$

then it is

$$r \in Pr_L^{-1}\big(Pr(uv)\big) \Rightarrow t_f \in r\,.$$

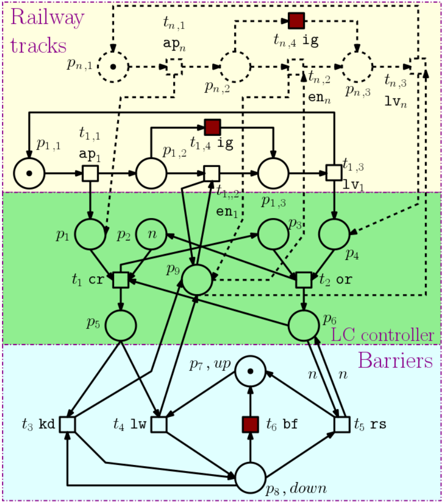
The railway benchmark - 1/2

- **Modular** PN model of a railway system that includes
  - *n* tracks
  - level crossing (LC) controller
  - the barriers
- Two *classes* of fault events are modeled by unobservable transitions
  - the *i*-th transition ($t_{i,4}$, ig) indicates that the *i*-th train enters the LC zone before the controller lowers the barriers;
  - the transition (t_6 , bf) indicates a defect in the barriers that results in a premature raising.



The railway benchmark 2/2

- The proposed *optimization-based* approach cannot be used to assess non-diagnosability
- The fault $(t_{i,4}, \texttt{ig})$ is not diagnosable when $n > 1$.
- **Only $(t_6, \texttt{bf})$ will be considered for the comparison**



$\mathcal{K}$-diagnosability via solution of ILP problems 1/3

- Originally proposed in *Basile et al.*, Automatica-2012
- Gives a necessary and sufficient condition to check $\mathcal{K}$-diagnosability in **bounded and live** labeled net systems
- **Cannot be used to assess non-diagnosability**

$\mathcal{K}$-diagnosability via solution of ILP problems 2/3

- A labeled bounded and live net system $G = \langle N, \boldsymbol{m}_0, \lambda \rangle$
- A fault transition t_f
- A positive integer $\mathcal{J}$ such that inequalities (1) (denoted with $\mathcal{F}(\boldsymbol{m}_0, \hat{t}, \mathcal{J}, \mathcal{K})$) describe the set

$$\mathcal{M}(t_f) = \left\{ \boldsymbol{m} \in \mathbb{N}^m \mid \left(\boldsymbol{m}_0 [u\rangle \boldsymbol{m} \right) \bigwedge \left(t_f \notin u \right) \bigwedge \left(\boldsymbol{m} [t_f\rangle \right) \right\}$$

$$\left\{ \begin{array}{lr}
\boldsymbol{m}_0 \geq \mathbf{Pre} \cdot \boldsymbol{u}_1 & \\
\boldsymbol{m}_0 + \boldsymbol{C} \cdot \boldsymbol{u}_1 \geq \mathbf{Pre} \cdot \boldsymbol{u}_2 & \\
\ldots & \text{(1a)} \\
\boldsymbol{m}_0 + \boldsymbol{C} \cdot \sum_{i=1}^{\mathcal{J}-1} \boldsymbol{u}_i \geq \mathbf{Pre} \cdot \boldsymbol{u}_{\mathcal{J}} & \\
\boldsymbol{m}_0 + \boldsymbol{C} \cdot \sum_{i=1}^{\mathcal{J}} \boldsymbol{u}_i \geq \mathbf{Pre}(\cdot, \hat{t}) & \text{(1b)} \\
\boldsymbol{m}_0 + \boldsymbol{C} \cdot \sum_{i=1}^{\mathcal{J}} \boldsymbol{u}_i + \boldsymbol{C}(\cdot, \hat{t}) \geq \mathbf{Pre} \cdot \boldsymbol{v}_1 & \\
\mathbf{Pre} \cdot \boldsymbol{v}_2 & \\
\ldots & \text{(1c)} \\
\boldsymbol{m}_0 + \boldsymbol{C} \cdot \sum_{i=1}^{\mathcal{J}} \boldsymbol{u}_i + \boldsymbol{C}(\cdot, \hat{t}) + \boldsymbol{C} \cdot \sum_{j=1}^{\mathcal{K}-1} \boldsymbol{v}_j \geq \mathbf{Pre} \cdot \boldsymbol{v}_{\mathcal{K}} & \\
\sum_{i=1}^{\mathcal{J}} \boldsymbol{u}(\hat{t}) = 0 & \text{(1d)} \\
\left\| \sum_{j=1}^{\mathcal{K}} \boldsymbol{v}_j \right\|_1 \geq \mathcal{K} & \text{(1e)}
\end{array} \right.$$

$\mathcal{K}$-diagnosability via solution of ILP problems 3/3

Theorem

Given a positive integer $\mathcal{K}$, t_f is $\mathcal{K}$-diagnosable if and only if there exist $3\left(\mathcal{J}+\mathcal{K}\right)$ vectors $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_{\mathcal{J}}, \boldsymbol{v}_1, \ldots, \boldsymbol{v}_{\mathcal{K}}$, $\boldsymbol{\epsilon}_1, \ldots, \boldsymbol{\epsilon}_{\mathcal{J}+\mathcal{K}}, \boldsymbol{s}_1, \ldots, \boldsymbol{s}_{\mathcal{J}+\mathcal{K}} \in \mathbb{N}^n$ such that

$$\min_{s.t.\ \mathcal{LD}\left(\mathbf{m}_0, t_f, \mathcal{J}, \mathcal{K}\right)} \sum_{r=1}^{\mathcal{J}+\mathcal{K}} \epsilon_r(t_f) \neq 0\,,$$

where the set $\mathcal{LD}\left(\mathbf{m}_0, t_f, \mathcal{J}, \mathcal{K}\right)$ includes $\mathcal{F}\left(\mathbf{m}_0, \hat{t}, \mathcal{J}, \mathcal{K}\right)$ and other similar linear constraints.

Semi-symbolic diagnoser (SSD)

- All *graph-based* approaches use a deterministic graph (called *diagnoser*) whose nodes contain a set of reachable (normal and/or faulty) markings and whose arcs are the observed events
- A diagnoser can be used both to check diagnosability and to perform the online diagnosis (in the case of diagnosable systems)
- The SSD approach was originally proposed in *Boussif*, PhD thesis and *Boussif et al.*, VECoS-2015
- It is based on the computation of a *semi-symbolic diagnoser*
- The SSD technique shows three interesting features compared to other approaches
  - it adopts a structure that explicitly separates normal (non-faulty) and the faulty markings in each node of the diagnoser
  - it uses a compact representation of the node markings using *binary decision diagrams*

Check diagnosability with SSD

Theorem

An LPN is said to be diagnosable, w.r.t. T_f, if and only if for each $F-$uncertain cycle $c\ell$ in its SSD $\mathcal{D}$, if $\rho^{c\ell} = \mathcal{S}_1, \mathcal{S}_2, \ldots$ is its indicating sequence, then $\exists\ i \in \mathbb{N}^* \ :\ \mathcal{S}_i = \emptyset$.

- *An F-uncertain cycle is a cycle in the SSD in which all nodes contains both normal and faulty marking*
- *Given an F-uncertain cycle, the associated $c\ell$-indicating sequence $\rho^{c\ell} = \mathcal{S}_1, \mathcal{S}_2, \ldots,$ is an infinite sequence of sets of markings, such that:*
  - $\mathcal{S}_1 = a_1.\mathcal{M}_F$
  - $\forall\ i > 1 :\ \mathcal{S}_i = Reach_{T_{uo}}(Img(\mathcal{S}_{i-1}, T_{\alpha_{(i-1)_{mod_n}}}))$

Experimental setup

- In order to apply the chosen *optimization-based* approach, a Matlab® script that calls the FICO™ Xpress API to solve the ILP problem was used **(off-the-shelf software)**
- The SSD approach is implemented by the DPN-SOG tool **(ad hoc software tool)**
- The hardware platform was a 64-bit PC equipped with CPU Intel® Core™ i3-6100U, at 2.30 GHz with 4GB of RAM

Preliminary comments

- The current implementation of the SSD approach within DPN-SOG permits to assess diagnosability but not $\mathcal{K}$-diagnosability
- The considered *ILP-based* approach cannot be used to assess non-diagnosability
- The comparison is made only on fault (t_6 , $\texttt{bf}$)

Experimental results

n	Petri net features				Diagnosability via SSD				$\mathcal{K}$-diagnosability via ILP				
	$\lvert P\rvert$	$\lvert T\rvert$	$\lvert\mathcal{N}\rvert$	$\lvert\mathcal{A}\rvert$	$\lvert\mathcal{D}_S\rvert$	$\lvert\mathcal{D}_T\rvert$	$\mathcal{D}_e$ (s)	$\mathcal{D}_m$ (kB)	$\mathcal{K}$	Last_ILP$_e$ (s)	Total_ILP$_e$ (s)	#constr. (origin / Xpress)	#unkow. (origin / Xpress)
1	12	10	20	43	10	14	**0**	44	7	0.3	**9**	721 / 225	228 / 180
2	15	14	142	500	83	205	**0**	1056	13	0.6	**26**	1171 / 467	425 / 380
3	18	18	832	4085	483	1745	**1**	8696	19	0.7	**56**	1729 / 798	682 / 639
4	21	22	4314	27142	2434	11774	**2**	80400	25	1.1	**108**	2395 / 1237	999 / 923
5	24	26	20556	157551	11304	69112	**30**	430456	31	4	**194**	3169 / 1764	1376 / 1294
6	27	30	92070	831384	56136	414299	**458**	2155100	37	2.7	**326**	4051 / 2386	1813 / 1725
7	30	34	393336	4086585	261262	2282890	**7836**	10167015	43	5.6	**507**	5041 / 3110	2310 / 2197
8	33	38	1618866	19013130	*	*	o.t.	*	49	6.4	**767**	6139 / 3940	2867 / 2743
9	36	42	*	*	*	*	o.t.	*	55	8.8	**1079**	7345 / 5006	3484 / 3351
10	39	46	*	*	*	*	o.t.	*	61	11.8	**1514**	8659 / 6013	5822 / 4017
11	42	50	*	*	*	*	o.t.	*	64	20	**1874**	12519 / 6686	11400 / 4555
12	45	54	*	*	*	*	o.t.	*	67	32	**3630**	13962 / 7688	12798 / 5125

*: No result obtained in 4 hours. o.t.: Out of time (more than 4 hours).

- n: the number of tracks;
- $|P|$ and $|T|$: the number of places and transitions in the PN models, respectively;
- $|\mathcal{N}|$ and $|\mathcal{A}|$: the number of nodes and arcs in the reachability graph, respectively;
- $|\mathcal{D}_S|$ and $|\mathcal{D}_T|$: the numbers of nodes and arcs in the SSD, respectively;
- $\mathcal{D}_e$ and $\mathcal{D}_m$: the required time and memory to generate perform the verification respectively;
- $\mathcal{K}$: number of events needed to detect the fault;
- Last_ILP$_e$: the time taken by Xpress to solve the ILP problem that satisfies Theorem 1;
- Total_ILP$_e$: the time taken by Xpress to solve the $\mathcal{K}$ ILP problems needed to assess $\mathcal{K}$-diagnosability;
- #const.: the number of constraints in the ILP problem that satisfies Theorem 1 before and after Xpress presolver, respectively;
- #unkow.: the number of unknowns in the ILP problem that satisfies Theorem 1 before and after Xpress presolver, respectively.

Conclusions - 1/2

- Although the proposed *optimization-based* approach requires to solve a number of ILP problems equal to $\mathcal{K}$ to assess $\mathcal{K}$-diagnosability, as soon as the size of the model becomes relatively large (in our case, as soon as $n > 6$), the time needed to perform the analysis becomes way lower than the one required by the *graph-based* SSD approach
  - It should be noticed the SSD algorithm has been directly implemented in C++, the *ILP-based* approach has been deployed in the Matlab® environment and relies on the FICO™ Xpress API. Hence, from the implementation point-of-view, there is a time overhead for the latter approach that is bigger than for the former, and this fact may have a non negligible impact when the size of the problem is relatively small
- Given the exponential explosion of the state space, the *graph-based* approach becomes practically unfeasible for $n > 7$, not terminating within the 4 hours timeout on the considered platform

Conclusions - 2/2

- Since it does not require the explicit computation of the reachability set, the *ILP-based* is particularly well suited for LPN models with a high level of *parallelism*
  - An additional track has a significant impact on the size of the model state space, but it does not affect too much the efficiency of *ILP-based* approach
  - This result is achieved thanks to the fact that the algebraic formulation enables to exploit the parallelism in the dynamic evolution of each track, and that the tracks evolve in parallel.
- The *ILP-based* approach exploits commercial tools for the solution of the ILP problems
  - This permits to takes advantage of all the preprocessing processes of these commercial tools
  - In the considered case, the number of constraints and unknowns after the run of the Xpress presolver is always smaller than the one of the original ILP problem, and this has a positive impact on the time needed to solve the problem

Questions?