TECHNISCHE UNIVERSITÄT DRESDEN

Faculty of Traffic Sciences Chair of Traffic Flow Science

On Extracting Minimally Infeasible Periodic Event Networks

Peter Großmann, Jens Opitz

Rome, 07/04/2013

Outline

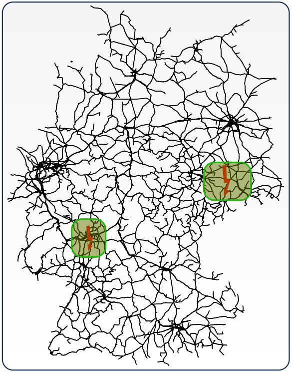
Motivation

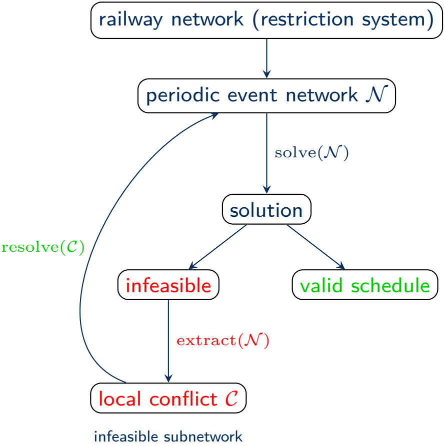
- many industrial problems encoded to SAT
- e. g., periodic event scheduling problem (PESP)
- used for timetabling in railway networks
- SAT solver's answer: formula $\mathcal{F}$ either satisfiable or unsatisfiable
- $\mathcal{F}$ resp. railway network often over-constrained in real-world instances
  $\rightarrow$ minimally unsatisfiable subformula extraction (MUS)

Motivation

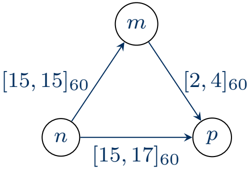
Public Railway Transport Networks

goal: compute valid timetable



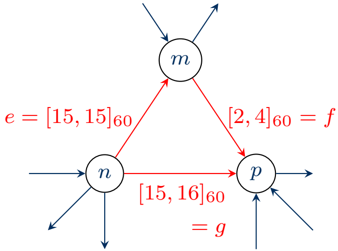


Periodic Event Scheduling Problem (PESP)



- period $T = 60$
- (periodic) events (nodes) n, m, $p \in \mathcal{V}$
  - schedule $\Pi : \mathcal{V} \to [0, T-1] \subseteq \mathbb{N}$
  - potential $\Pi(n)$, $\Pi(m)$, $\Pi(p)$
- constraints resp. activities (edges) $\mathcal{E}$
- e. g., (n, m) with $[15, 15]_{60}$
  - constraint holds iff $\Pi(m) - \Pi(n) \in [15, 15]_{60}$
  - e. g., $\Pi(n) = 2, \Pi(m) = 17$ or $\Pi(n) = 50, \Pi(m) = 5$
- schedule Π is valid iff all constraints hold
- tuple $\mathcal{N} = (\mathcal{V}, \mathcal{E}, T)$ is called periodic event network

Local Conflict



- infeasible periodic event network $\mathcal{N} = (\mathcal{V}, \mathcal{E}, T)$

Definition (Local Conflict)

Periodic event network $\mathcal{C} = (\mathcal{V}, \mathcal{Z}, T)$ with $\mathcal{Z} \subseteq \mathcal{E}$ is called **local conflict** iff

(i) $\mathcal{C}$ is infeasible

(ii) $\mathcal{C}$ feasible for any removed constraint in $\mathcal{Z}$

- local conflict extraction: find $\mathcal{C} = (\mathcal{V}, \{e, f, g\}, T)$

Propositional Logic (SAT)

$$\mathcal{F} = \neg p \wedge (p \vee q)$$

- literal: $L = p$ or $L = \neg p$ (variable or its negation), $p \in \mathcal{R}$
- clause c: disjunction of literals
- formula $\mathcal{F}$ in conjunctive normal form (CNF): conjunction of clauses
- interpretation $J : \mathcal{R} \to \{F, T\}$
- $\mathcal{F}$ is satisfiable ($\mathcal{F}^J = T$) iff for all clauses at least one literal $L^J = T$
- e. g., $p^J = F$, $q^J = T$ implies $\mathcal{F}^J = T$
- $\mathcal{F}$ unsatisfiable if no such J exists

Minimally Unsatisfiable Subformula

$$\mathcal{F} = \neg p \wedge (p \vee q) \wedge \neg q \wedge \ldots$$

- split unsatisfiable $\mathcal{F}$ into disjoint sets of clauses $\{R\} \cup \mathcal{G}$ (domain-specific)
  $\rightarrow \mathcal{F} = R \wedge (\bigwedge_{K \in \mathcal{G}} K)$
- e. g., $\mathcal{G} = \{K_1, K_2, \ldots\}$, $R = \emptyset$

Definition

$\mathcal{M} = R \wedge (\bigwedge_{K \in \mathcal{G}'} K)$ with $\mathcal{G}' \subseteq \mathcal{G}$ is called **high-level minimally unsatisfiable subformula** (HLMUS) iff

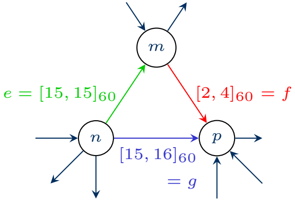
(i) $\mathcal{M}$ is unsatisfiable

(ii) for any $K \in \mathcal{G}'$: $\mathcal{M} \setminus K$ is satisfiable

- HLMUS extraction: find $\mathcal{G}' = \{K_1, K_2\}$

Local Conflict Extraction

Reduce Local Conflict to HLMUS



- let $\mathcal{N} = (\mathcal{V}, \mathcal{E}, T)$ infeasible periodic event network with $\mathcal{E} = \{e, f, g, \ldots\}$
- encode $\mathcal{N}$ as propositional formula $\mathcal{F}$ in CNF:

$$\begin{aligned}\mathcal{F} = \text{encode}(\mathcal{N}) = {} & \text{encodeNode}(n) \wedge \text{encodeNode}(m) \wedge \text{encodeNode}(p) \wedge \ldots \\ & \wedge \text{encodeEdge}(e) \wedge \text{encodeEdge}(f) \wedge \text{encodeEdge}(g) \wedge \ldots\end{aligned}$$

- split $\mathcal{F}$ in sets of clauses $\{R\} \cup \mathcal{G}$:

$$R = \text{encodeNode}(n) \wedge \text{encodeNode}(m) \wedge \text{encodeNode}(p) \wedge \ldots$$
$$\mathcal{G} = \{\text{encodeEdge}(e), \text{encodeEdge}(f), \text{encodeEdge}(g), \ldots\}$$

Results

- goal: extract local conflicts effectively/efficiently from periodic event network $\mathcal{N}$
- 4 solvers under test
  1. polynomial-based algorithm by Nachtigall, Opitz et al.
  2. `SAText`: comparison solver (SAT-based)
  3. `MoUsSaKa`: by Kottler (HLMUS extractor, destructive algorithm)
  4. `Haifa-MUC`: by Ryvchin et al. (HLMUS extractor, resolution-based)

Results

- periodic event networks $\mathcal{N} = (\mathcal{V}, \mathcal{E}, T)$: industrial scenarios
- generated by software system TAKT (TU Dresden)
- railway networks: south-east, south-west Germany and intercity network
- data from DB Netz AG
- runtime in seconds (2 days timeout)

$\mathcal{N}$	$\lvert\mathcal{E}\rvert$	polyn.	SAText	MoUsSaKa	Haifa-MUC
1	128	1	2	1	1
2	211	1	5	1	1
3	327	1	9	1	1
4	533	2	12	1	1
5	689	2	7	1	1
6	702	3	82	10	4
7	788	-	102 203	383	out of memory
8	5 571	-	7 269	72	84
9	6 305	-	timeout	timeout	out of memory
10	9 821	-	36	2	5
11	11 082	-	48	2	4
12	14 589	-	23	3	5

TECHNISCHE UNIVERSITÄT DRESDEN

Conclusion

- automatic extraction of local conflicts has decisive importance
- can be encoded into a SAT-based domain (HLMUS)
- instances can be solved efficiently by state-of-the-art solvers
- outlook:
  - better encodings for difficult/complex local conflicts
  - use solvers in parallel

Sources

Peter Großmann.
Extracting and Resolving Local Conflicts in Periodic Event Networks.
Diploma thesis, TU Dresden, Faculty of Computer Science, 2012.

Peter Großmann, Steffen Hölldobler, Norbert Manthey, Karl Nachtigall, Jens Opitz, and Peter Steinke.
Solving periodic event scheduling problems with SAT.
In IEA/AIE, volume 7345 of LNAI, pages 166–175. Springer, 2012.

Karl Nachtigall and Jens Opitz.
A modulo network simplex method for solving periodic timetable optimisation problems.
In Operations Research, pages 461–466. Springer, 2008.

Christos H. Papadimitriou and David Wolfe.
The complexity of facets resolved.
J. Comput. Syst. Sci., 37(1):2–13, 1988.

Vadim Ryvchin and Ofer Strichman.
Faster extraction of high-level minimal unsatisfiable cores.
In Karem A. Sakallah and Laurent Simon, editors, SAT, volume 6695 of LNCS, pages 174–187. Springer, 2011.

Modulo Interval resp. Periodic Extension

- interval $[a, b] := \{x \in \mathbb{Z} \mid a \leq x \leq b\} \subset \mathbb{Z}$, $a, b \in \mathbb{Z}$
- e. g., $[2, 4] = \{2, 3, 4\}$
- interval modulo $T \in \mathbb{N}$:

$$[a, b]_T := \bigcup_{z \in \mathbb{Z}} [a + z \cdot T,\ b + z \cdot T] \subseteq \mathbb{Z}$$

- e. g., $[2, 4]_{10} = \ldots \cup [-8, -6] \cup [2, 4] \cup [12, 14] \cup [22, 24] \cup \ldots$

Complexity Classification

- which complexity class does Local Conflicts (LC) belong to?
- complexity class $coNP = \{\bar{L} \mid L \in NP\}$
- e. g., $\mathrm{UNSAT} = \{\mathcal{F} \mid \mathcal{F}$ unsatisfiable$\}$ is $coNP$-complete
- complexity class $DP = \{L_1 \cap L_2 \mid L_1 \in NP, L_2 \in coNP\}$

Theorem

LC is DP-complete

Complexity Classification

Lemma

$\text{PESP} = \{\mathcal{N} \mid \mathcal{N} \text{ is feasible}\}$ is NP-complete

Lemma

$\overline{\text{PESP}} = \{\mathcal{N} \mid \mathcal{N} \text{ is infeasible}\}$ is $coNP$-complete

Proof. (LC is DP-complete)

to show:

1. LC $\in DP$
2. LC is DP-hard

1.:

- find L_1, L_2, such that $LC = L_1 \cap L_2$ and $L_1 \in NP, L_2 \in coNP$.
- let $\mathcal{N} = (\mathcal{V}, \mathcal{Z}, T)$ be a PEN, $\mathcal{Z} = S \cup \bigcup_{E \in \mathcal{D}} E$
- $L_1 = \{\mathcal{N} \mid \mathcal{N} \text{ is infeasible}\} \in coNP$
- $L_2 = \{\mathcal{N} \mid \forall E \in \mathcal{D} : (\mathcal{V}, \mathcal{Z} \setminus E, T) \text{ is feasible}\} \in NP$, because $|\mathcal{D}| < \infty$

MUS zu LC

Definition (MUS)

$\mathcal{F}$ is **minimally unsatisfiable subformula** iff $\mathcal{F}$ is unsatisfiable and for any clause $c \in \mathcal{F}$: $\mathcal{F} \setminus \{c\}$ is unsatisfiable.

Lemma

MUS is DP-hard

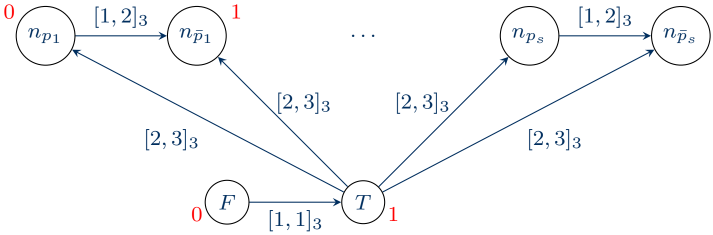
- encode propositional variables as events
- encode disjunction $(p \vee q)$ as periodic event network
- connect them, in order to encode $c \in \mathcal{F}$

Complexity Classification

Encode Propositional Variables

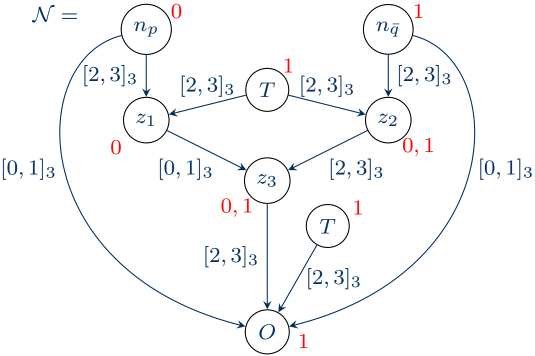
- let p_i $(i \in \{1, \ldots, s\})$ be propositional variables
- periodic event network $\mathcal{N} = (\mathcal{V}, \mathcal{E}, 3)$
- WLOG $\Pi(F) = 0$, $\Pi(T) = 1$
- encode literals p and $\neg p$
- $\Pi(n_p), \Pi(n_{\bar{p}}) \in \{0, 1\}$ semantically means $p^J, (\neg p)^J \in \{T, F\}$

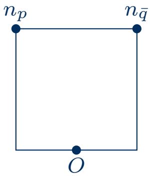
$\mathcal{N} =$



Disjunction

- encode disjunction, e. g., $(p \lor \neg q)$
- encode literals p, $\neg q$ as before
- $(p \lor \neg q)^J$ equivalent to $\Pi(O)$

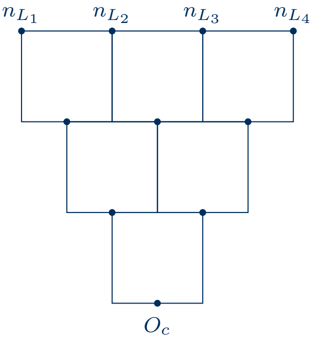




Complexity Classification

Clause

- encode clause as layered encoded disjunctions
- e. g., $(p \vee q \vee r) \equiv (p \vee q) \vee (q \vee r)$
- e. g., network $\mathcal{N}$ for each $c = (L_1 \vee \ldots \vee L_4)$
- assign n_{L_i} $(i \in \{1, \ldots, 4\})$ corresponding to J
- then, $\Pi(O_c)$ is equivalent to c^J



Complexity Classification

MUS to LC

Lemma

MUS is DP-hard

Lemma

MUS $\leq_p$ LC

Lemma

LC $\in DP$

Theorem

LC is DP-complete

Instance Names

- periodic event networks $\mathcal{N} = (\mathcal{V}, \mathcal{E}, T)$: industrial scenarios
- generated by software system TAKT (TU Dresden)
- railway networks: south-east, south-west Germany and intercity network
- data from DB Netz AG
- runtime in seconds (2 days timeout)

$\mathcal{N}$	instance
1	p_1
2	p_2
3	p_3
4	p_4
5	p_5
6	p_6
7	k
8	fern
9	b
10	dp_sta_1
11	dp_sta_2
12	dp_sta_3

Order Encoding (Finite Ordered Domains)

$$F = p_{x,6} \wedge \neg p_{x,2} \wedge (\neg p_{x,2} \vee p_{x,3}) \\ \wedge (\neg p_{x,3} \vee p_{x,4}) \wedge (\neg p_{x,4} \vee p_{x,5}) \wedge (\neg p_{x,5} \vee p_{x,6})$$

- encoding of $x \in I$ with $I = [l, u] \subset \mathbb{Z}$
- e. g., $x \in [3, 6]$
- propositional variable $p_{x,i}$ ($i \in [l, u]$) with meaning: $x \leq i$
- $\Rightarrow$ two cases:
  1. $p_{x,i} = T$, then $x \leq i$
  2. $p_{x,i} = F$, then $x \not\leq i$, hence $x \geq i + 1$
- since $x \in [3, 6]$, it must hold:
  - $x \leq 6$, implies $p_{x,6}$ must be T
  - $x \geq 3 \Leftrightarrow x \not\leq 2$, hence $\neg p_{x,2}$ must be T
- $\forall i \in \{3, \ldots, 6\} : (x \leq i - 1) \Rightarrow (x \leq i)$
- that is $p_{x,i-1} \rightarrow p_{x,i}$
- which is semantically equivalent to $(\neg p_{x,i-1} \vee p_{x,i})$ $(\forall i \in \{3, \ldots, 6\})$

Encoding PESP as SAT

Order Encoding (Extract Value of x)

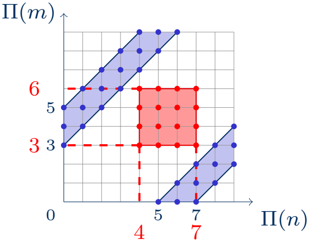
- SAT solver solves SAT instance and returns assignment for $p_{x,i}$ ($\forall i \in \{2, \ldots, 6\}$), if satisfiable
- e. g.,

$p_{x,2}$	$p_{x,3}$	$p_{x,4}$	$p_{x,5}$	$p_{x,6}$
F	F	F	T	T

- hence, $x \leq 6$ and $x \leq 5$
- but $x \not\leq 4$ and $x \not\leq 3$, ...
- $\Rightarrow x \geq 5$
- $x \leq 5$ and $x \geq 5$ implies $x = 5$
- for each variable assignment the exact value for x can be extracted

Encoding PESP as SAT

Constraint



- let be $T = 10$, $e = (n, m)$ with constraint $[3, 5]_{10}$
- exclude not feasible regions wrt. $[3, 5]_{10}$
- e. g., no pair $(\Pi(n), \Pi(m)) : \Pi(n) \leq 7, \Pi(n) \geq 4, \Pi(m) \leq 6, \Pi(m) \geq 3$
- iff $\neg((\Pi(n) \leq 7) \wedge (\Pi(n) \geq 4) \wedge (\Pi(m) \leq 6) \wedge (\Pi(m) \geq 3))$
- iff $\neg((\Pi(n) \leq 7) \wedge \neg(\Pi(n) \leq 3) \wedge (\Pi(m) \leq 6) \wedge \neg(\Pi(m) \leq 2))$
- iff $\neg(p_{n,7} \wedge \neg p_{n,3} \wedge p_{m,6} \wedge \neg p_{m,2})$
- iff $(\neg p_{n,7} \vee p_{n,3} \vee \neg p_{m,6} \vee p_{m,2})$
- connect $[\neg p_{n,7}, p_{n,3}, \neg p_{m,6}, p_{m,2}]$ to formula conjunctively