ARRIVAL: Next stop in CLS

ICALP 2018

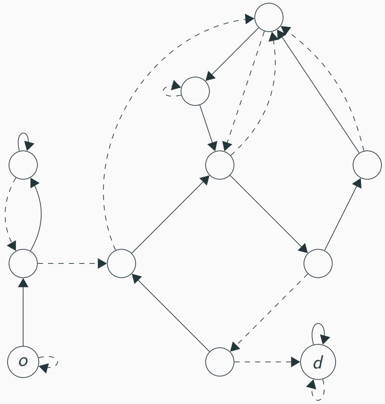
Bernd Gärtner, Thomas Dueholm Hansen, Pavel Hubáček, **Karel Král**, Hagar Mosaad, Veronika Slívová

July 11th, 2018

Definition of ARRIVAL

1. Each city... two out-going tracks *even* and *odd*
2. Each visit alternate outgoing edge

Question: If the train starts in origin *o* will it ever arrive to the destination *d*?



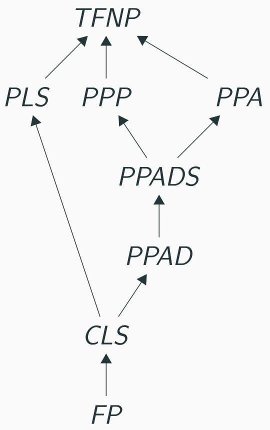
Known results

1. ARRIVAL Dohrau et al.'17:
    - Exponentially long route
    - Solvable in $NP \cap coNP$
2. Karthik C. S.'17:
    - Finding *NP* or *coNP* certificate is in *PLS*
    - Conjecture: search version of ARRIVAL is FPSPACE-complete
3. Fearnley et al.'17:
    - Decision version is *NL*-hard

Our results

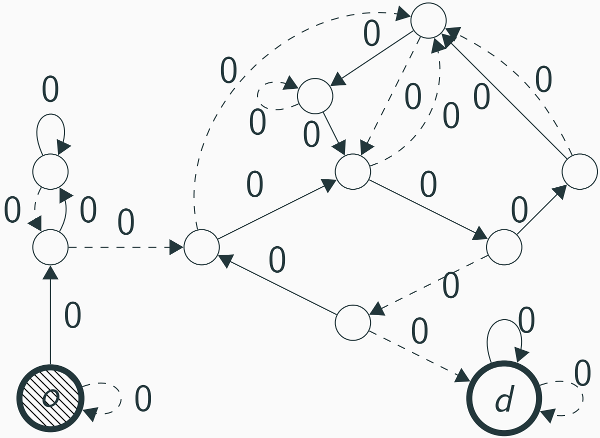
Our results:

1. ARRIVAL is in $UP \cap coUP$
2. Search version in CLS
3. Randomized $\mathcal{O}(1.4143^n)$ algorithm for ARRIVAL

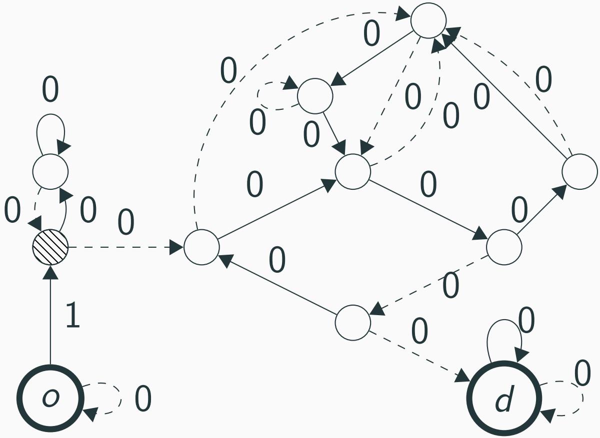


Daskalakis and Papadimitriou'11

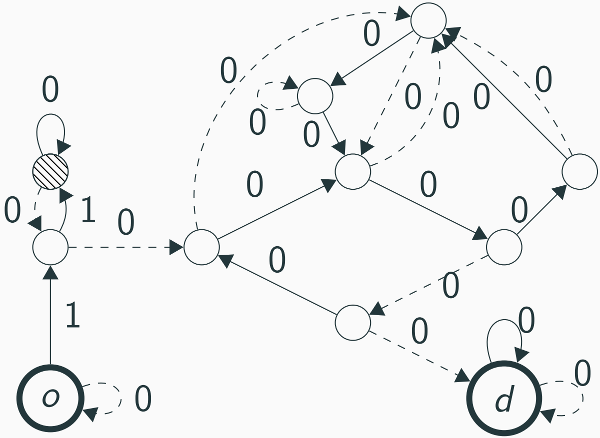
Switching flow and Run profile



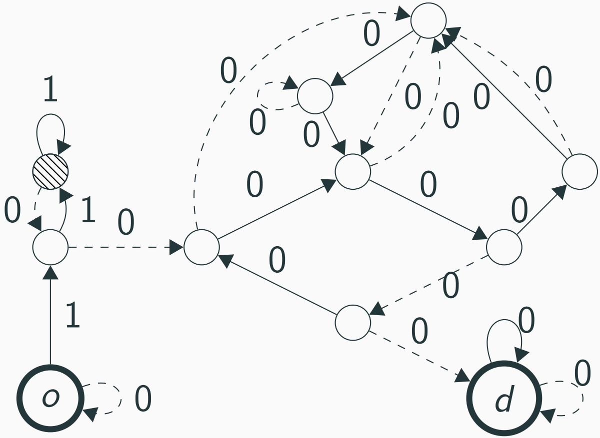
Switching flow and Run profile



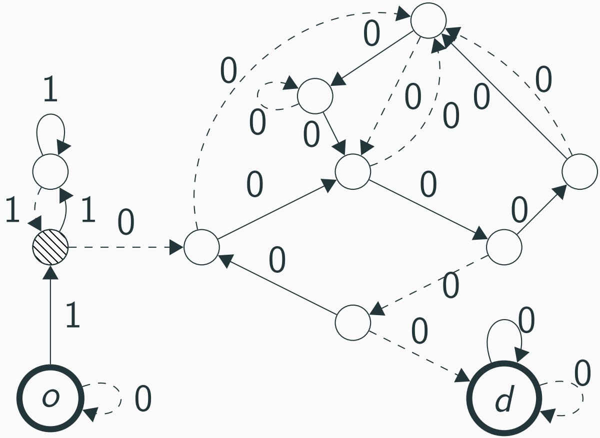
Switching flow and Run profile



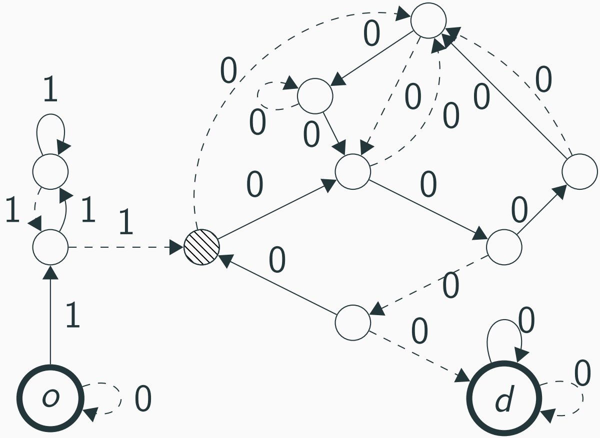
Switching flow and Run profile



Switching flow and Run profile



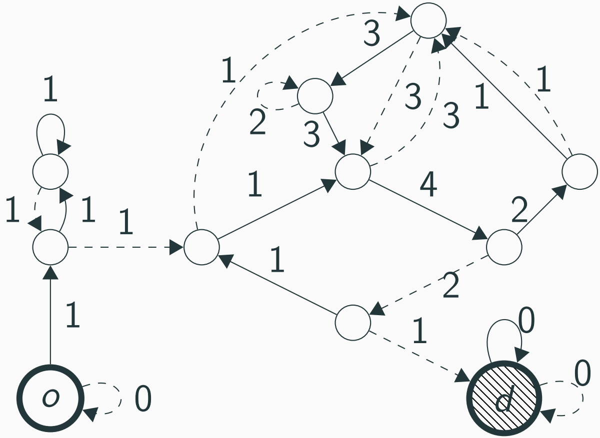
Switching flow and Run profile



Switching flow and Run profile

Many slides later...

Switching flow and Run profile



Switching flow and Run profile

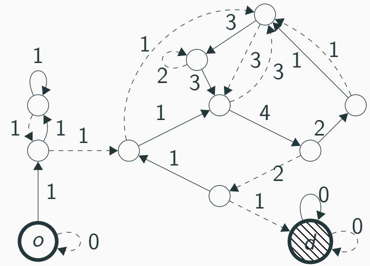
Def: $\boldsymbol{f} \in \mathbb{N}^{2n}$ is a *switching flow* if it satisfies:

1. Kirchhoff's Law (flow conservation)
2. Parity Condition

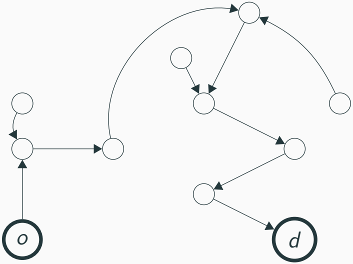
- A *Run profile*. . . route of the train
- (Dohrau et al.'17) *Switching flow* $\Rightarrow$ *run profile*
- *NP* certificate = switching flow
- *coNP* certificate = switching flow to dead-end destination $\overline{d}$

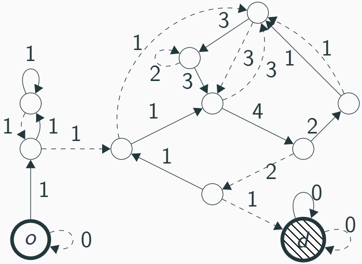
From a switching flow we get:

1. Position of the train
2. Last used edge for each vertex
3. Previous run profile



Graph of last used edges $G_{\boldsymbol{f}}^*$ and $UP \cap coUP$





Run profile verification

Theorem: A switching flow $\boldsymbol{f}$ is a run profile iff the train never left d or $\bar{d}$ and either:

1. There is no cycle in $G^*_{\boldsymbol{f}}$.
2. Or there is exactly one cycle in $G^*_{\boldsymbol{f}}$ which contains the end-vertex of $\boldsymbol{f}$.

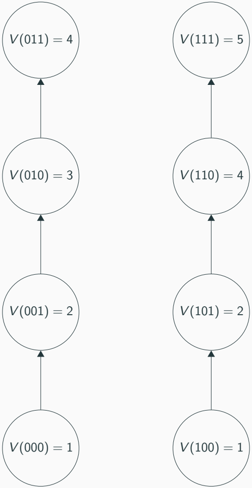
Run profile is unique and we can verify for given vector
$\Rightarrow$ $\textsc{Arrival}$ is in UP.

Continuous Local Search

CLS (Daskalakis and Papadimitriou'11)

- Finding approximate fixed points of contraction maps
- PLCP
- Finding min-max strategy in simple stochastic games
- ...

End of Metered Line

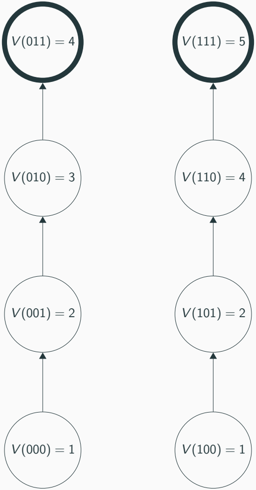


Types of solutions:

1. End of some line
2. Non-trivial beginning of some line
3. Broken valuation:
    - non-trivial $V(x) = 1$
    - or increase by > 1

Theorem (Hubáček and Yogev'17)
EOML is in CLS.

End of Metered Line

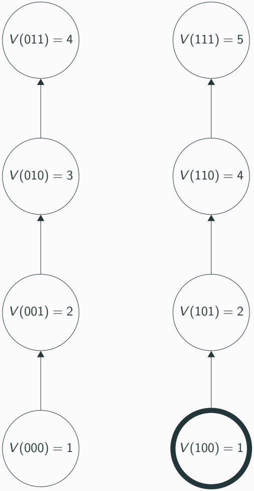


Types of solutions:

1. **End of some line**
2. Non-trivial beginning of some line
3. Broken valuation:
    - non-trivial $V(x) = 1$
    - or increase by > 1

Theorem (Hubáček and Yogev'17)
EOML is in CLS.

End of Metered Line

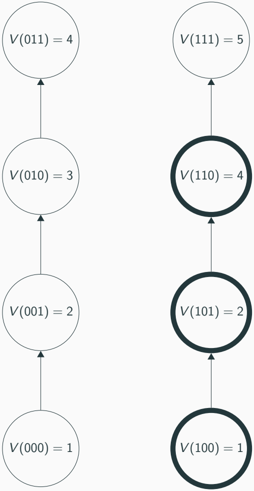


Types of solutions:

1. End of some line
2. **Non-trivial beginning of some line**
3. Broken valuation:
    - non-trivial $V(x) = 1$
    - or increase by > 1

Theorem (Hubáček and Yogev'17)
EOML is in CLS.

End of Metered Line



Types of solutions:

1. End of some line
2. Non-trivial beginning of some line
3. **Broken valuation:**
   - **non-trivial** $V(x) = 1$
   - **or increase by** > 1

Theorem (Hubáček and Yogev'17)
EOML is in CLS.

Reduction ARRIVAL to EOML

Reduction Arrival to EOML:

- Vertices – integer vectors of length $2n$
- If not run profile selfloop and $V(x) = 0$
- Successor/predecessor – next/previous step of the train
- Valuation = sum of the vector

Very special instance of EOML – just one line.

Aldous' Algorithm

EOML algorithm (Aldous'83):

Given EOML instance over strings $\{0,1\}^m$:

1. Let x_m be maximizing $V(x)$ among $2^{m/2}$ random strings.
2. Start from x_m, move to the successor until a solution is found.

Expected number of steps is $\mathcal{O}(m2^{m/2})$.

Aldous' Algorithm

EOML algorithm (Aldous'83):

Given EOML instance over strings $\{0,1\}^m$:

1. Let x_m be maximizing $V(x)$ among $2^{m/2}$ random strings.
2. Start from x_m, move to the successor until a solution is found.

Expected number of steps is $\mathcal{O}(m2^{m/2})$.

The obvious encoding of $\boldsymbol{f}$ (for each edge remember a number that is at most 2^n takes $\Theta(n^2)$ bits) is slower than the trivial simulation.

Aldous' Algorithm

EOML algorithm (Aldous'83):

Given EOML instance over strings $\{0,1\}^m$:

1. Let x_m be maximizing $V(x)$ among $2^{m/2}$ random strings.
2. Start from x_m, move to the successor until a solution is found.

Expected number of steps is $\mathcal{O}(m2^{m/2})$.

Our tool: The switching flow $\boldsymbol{f}$ can be efficiently decoded from the parities of visits of vertices and the position of the train.

$m = n + \log n$ gives us $\mathcal{O}\left(\mathrm{poly}(n)2^{\frac{n+\log n}{2}}\right) \in \mathcal{O}(1.4143^n)$.

Open questions

1. Is ARRIVAL polynomial time solvable?
2. Is there any evidence for not being polynomial time solvable?

Open questions

1. Is ARRIVAL polynomial time solvable?
2. Is there any evidence for not being polynomial time solvable?

Thank you for your attention!

Formal definitions.

Continuous Local Search

Definition (Daskalakis and Papadimitriou'11)
CLS is the class of total search problems reducible to the following problem called *CLOpt*.

Given two arithmetic circuits $f: [0,1]^3 \to [0,1]^3$ and $p: [0,1]^3 \to [0,1]$, and two real constants $\varepsilon, \lambda > 0$, find either a point $x \in [0,1]^3$ such that $p(f(x)) \leq p(x) + \varepsilon$ or a pair of points $x, x' \in [0,1]^3$ certifying that either p or f is not λ-Lipschitz.

Reminder: Arithmetic circuits are like polynomials but we can reuse intermediate results.

End of Metered Line

Definition (EOML)

Given circuits $S, P\colon \{0,1\}^m \to \{0,1\}^m$, and $V\colon \{0,1\}^m \to [2^m] \cup \{0\}$ such that $P(0^m) = 0^m \neq S(0^m)$ and $V(0^m) = 1$, find a string $x \in \{0,1\}^m$ satisfying one of the following:

1. either $P(S(x)) \neq x$ or $S(P(x)) \neq x \neq 0^m$,
2. $x \neq 0^m$ and $V(x) = 1$,
3. either $V(x) > 0$ and $V(S(x)) - V(x) \neq 1$ or $V(x) > 1$ and $V(x) - V(P(x)) \neq 1$.

The circuits P and S implicitly represent a graph.

The circuit V is a counter of distance from the all zeroes string.

Switching flow and Run profile

We say that $\boldsymbol{f} \in \mathbb{N}^{2n}$ is a *switching flow* if the following holds:

1. Kirchhoff's Law (flow conservation):

$$\forall v \in V\colon \sum_{e=(u,v)\in E} \boldsymbol{f}_e - \sum_{e=(v,w)\in E} \boldsymbol{f}_e = [v = d] - [v = o]\ ,$$

   where $[\cdot]$ is the indicator variable of the event in brackets.

2. Parity Condition:

$$\forall v \in V\colon \boldsymbol{f}_{s_1(v)} \le \boldsymbol{f}_{s_0(v)} \le \boldsymbol{f}_{s_1(v)} + 1\ .$$