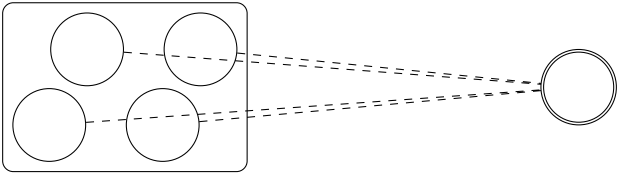
Time Unrolling Heuristics

Philipp Oldenburg

March 27, 2018

High-level idea

- Guided state space search

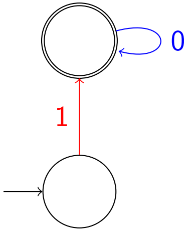


- Use cheapest network flow for estimating cheapest plan cost
- Estimation works better if we introduce time steps

What is a network flow

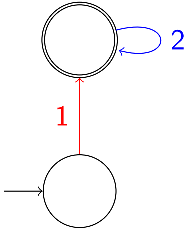
- Already introduced in 1956
  - *Network flow theory* by Jr. L. R. Ford
- Maps transitions to positive real numbers
- One unit gets moved from initial state to the goal states
- Non-goal flow only allowed in cycles
- Can be formalized as an LP

Flow example



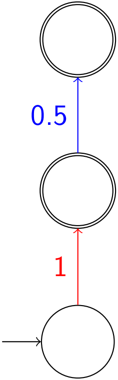
- Flow with cost of $1 \cdot cost(\rightarrow)$

Flow example (2)



- Flow with cost of $1 \cdot \mathit{cost}(\rightarrow) + 2 \cdot \mathit{cost}(\rightarrow)$

Flow example (3)

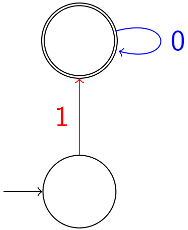


- Flow with cost of $1 \cdot cost(\rightarrow) + 0.5 \cdot cost(\rightarrow)$

Relationship flows and plans

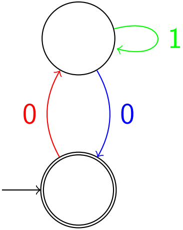
- For every flow a plan
- Other direction not guaranteed

Flow example - revisited



- Flow has corresponding plan: $\langle \rightarrow \rangle$

Flow example (3)

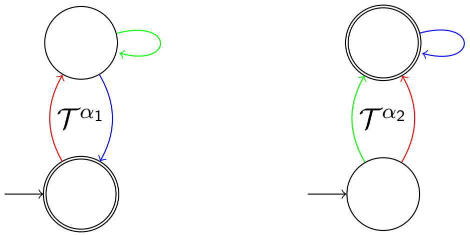


- Depicts flow with cost of *cost*($\rightarrow$) that uses an isolated cycle and has no corresponding plan
- Cycles might be bad for flow-plan relationship
- Here not of importance

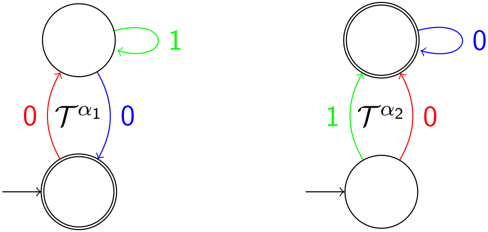
How flows are used

- Admissibly estimating plan costs
  - *An LP-based heuristic for optimal planning* CP 2007
- Combine information of multiple abstract transition systems
  - Synchronize flows for each operator
  - e.g. h^{SEQ} using atomic projections

Synchronized flow



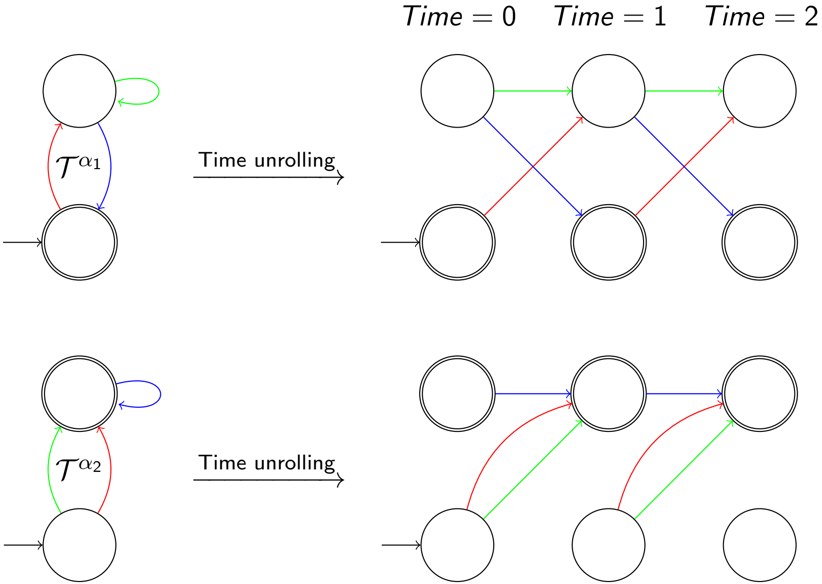
Synchronized flow example



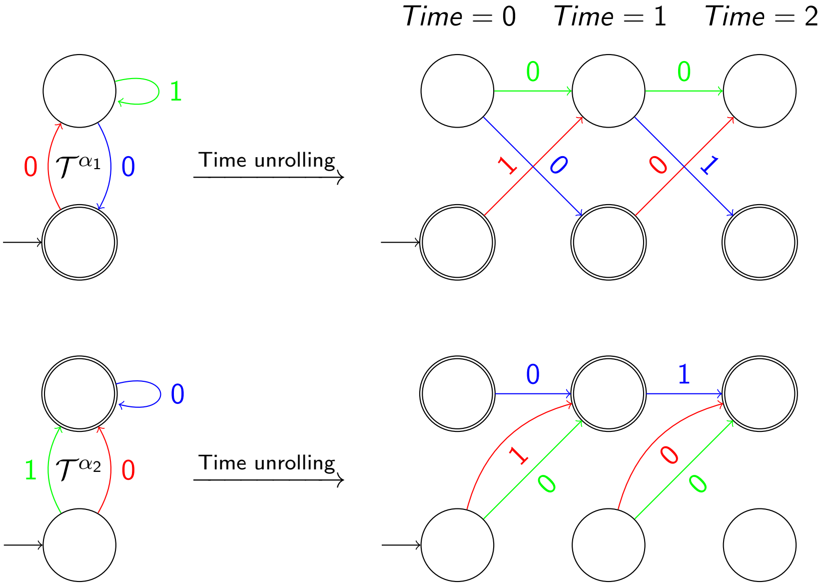
- Depicts cheapest synchronized flow with cost of *cost*($\rightarrow$) that uses an isolated cycle and has no corresponding plan
- This time cheapest synchronized flow
- Cycles might lead to low heuristic values $\rightarrow$ get rid of them

Time unrolling

Time unrolling example (2 time steps)



Time unrolling example (with cheapest synchronized flows)



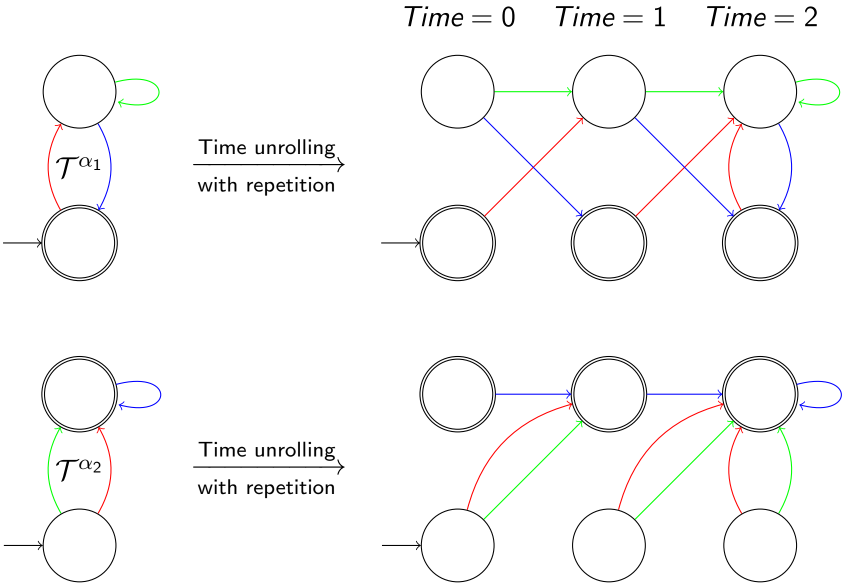
Time unrolling disadvantages

- $\#\textit{time steps} \cdot |\mathcal{S}|$ new states for every abstract transition system $\rightarrow$ bigger LP/IP
- Plans of abstractions might no longer be preserved
  - Only plans with length at most $\#\textit{time steps}$ are preserved

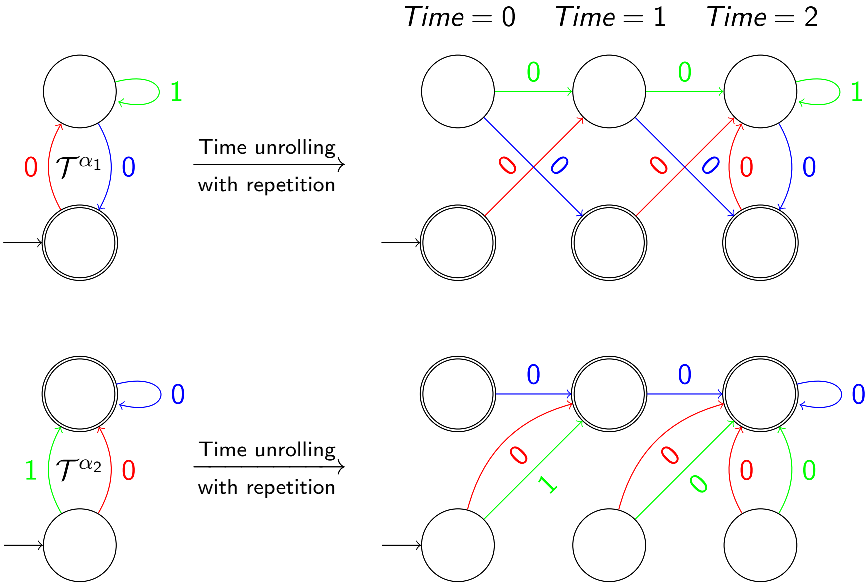
  This can make the heuristic inadmissible

  $\rightarrow$ New type of time unrolling

Time unrolling with repetition

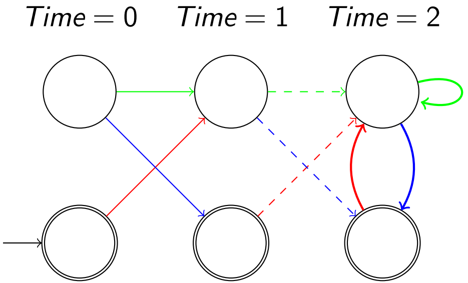


Time unrolling with repetition (cheapest synchronized flows)

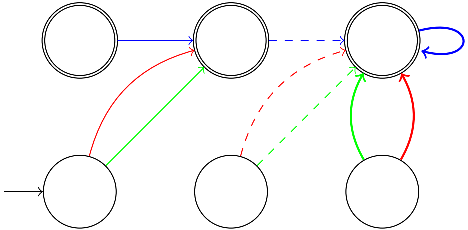


Time synchronization

Time unrolled $\mathcal{T}^{\alpha_1}$ with rep.

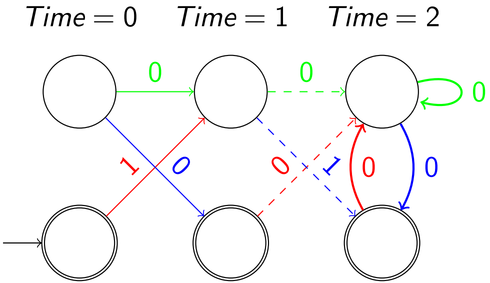


Time unrolled $\mathcal{T}^{\alpha_2}$ with rep.

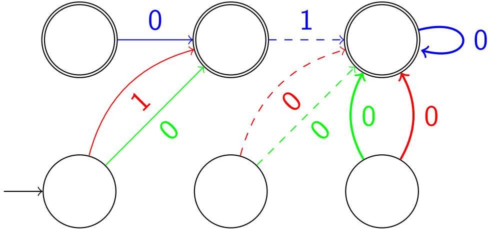


Time synchronization (with cheap. time synchronized flow)

Time unrolled $\mathcal{T}^{\alpha_1}$ with rep.



Time unrolled $\mathcal{T}^{\alpha_2}$ with rep.



Time synchronization properties

- Introduces new synchronization constraints
  - Before: Constraints for every operator
  - After: Constraints for every operator for every time step
- Can prevent cycle exploitation as seen

Time unrolling heuristics

h_n^{ATUR}

- Abstractions: atomic projections
- Uses time unrolling with repetition and n time steps
- Is defined as the cost of cheapest time synchronized flow
- n can be chosen depending on size of atomic projections
- IP version: $h_{n,\mathbb{N}}^{ATUR}$

h^{MATU}

- Abstractions: atomic projections
- Uses time unrolling (without repetition) and lowest number of time steps possible
- Is defined as the cost of cheapest time synchronized flow
- Adjusts the number of time steps dynamically
- IP version: $h_{\mathbb{N}}^{MATU}$

Properties of h_n^{ATUR} and h^{MATU}

Dominance relationships

- Given any integers n, m with $n \leq m$
  - $h^{SEQ} \leq h_n^{ATUR} \leq h_m^{ATUR} \leq h^{MATU}$
  - IP versions dominate their counterparts
  - $h_{\mathbb{N}}^{MATU} = h^*$

Admissibility

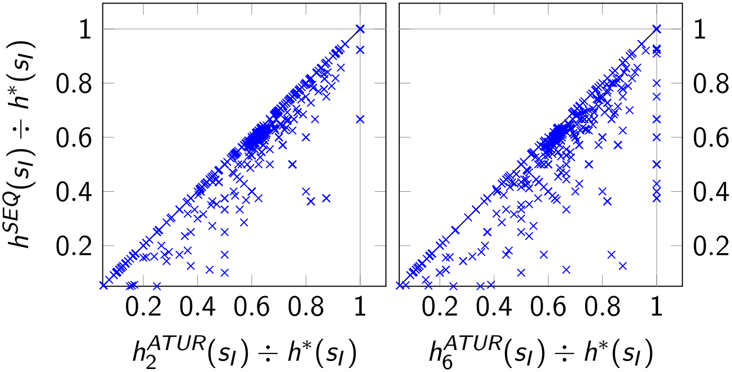
- h_n^{ATUR} admissible
- h^{MATU} only admissible with unit costs
- Analogously for IP versions

Experiments

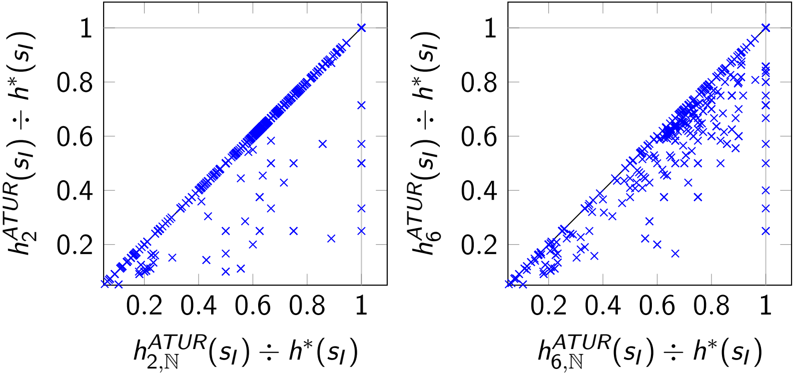
Experiment settings

- sciCORE
- Limits: 30 minutes and 2 GB
- Optimal tracks of IPC (1998-2014)
- h^{SEQ} and $h^{LM\text{-}cut}$ as comparison

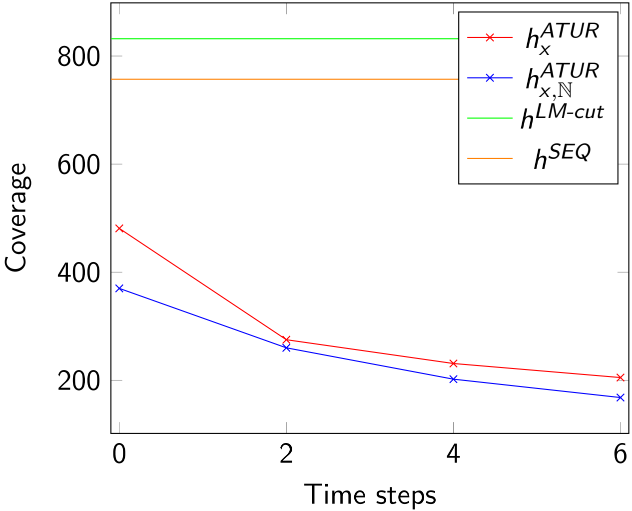
Experimental results - $h_n^{ATUR}(s_I)$



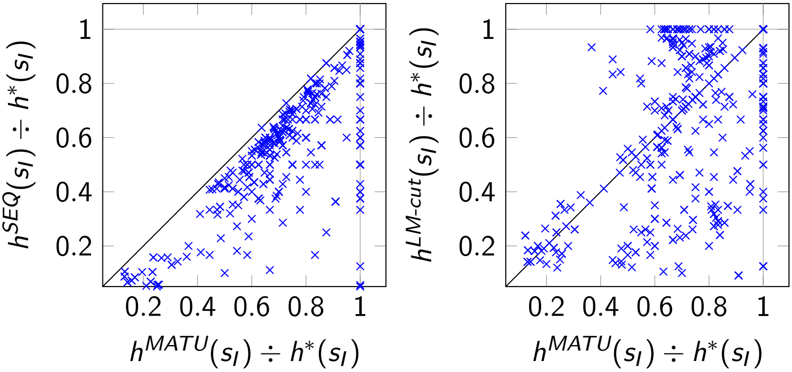
Experimental results - $h^{ATUR}_{n,\mathbb{N}}(s_I)$



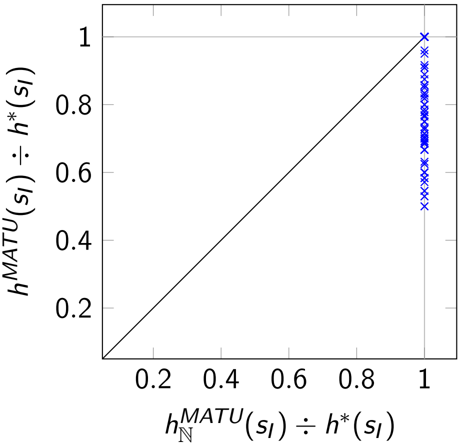
Experimental results - Coverage of $h_{n,\mathbb{N}}^{ATUR}$ and h_n^{ATUR}



Experimental results - $h^{MATU}(s_I)$



Experimental results - $h_{\mathbb{N}}^{MATU}(s_I)$



Experimental results - Coverage of h^{MATU} and $h_{\mathbb{N}}^{MATU}$

	Coverage	Coverage ÷ Coverage of h^{MATU}
$h^{LM\text{-}cut}$	832	4.8
h^{SEQ}	757	4.4
h^{MATU}	174	1
$h_{\mathbb{N}}^{MATU}$	127	0.7

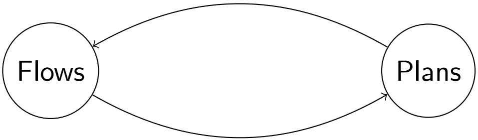
Future work
and conclusion

Future work

- Use other abstractions
  - e.g. projections to multiple variables
- Combine with other heuristics within operator counting framework
  - *LP-based Heuristics for Cost-optimal Planning* ICAPS 2014
- Remove cycles with less overhead

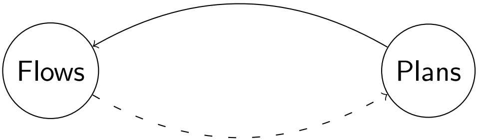
Conclusion

- Estimate cheapest plan cost with cheapest flow



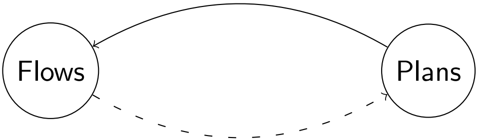
Conclusion

- Estimate cheapest plan cost with cheapest flow



Conclusion

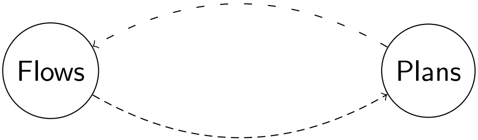
- Estimate cheapest plan cost with cheapest flow



- Introduce time steps

Conclusion

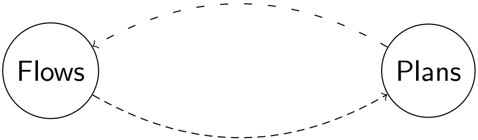
- Estimate cheapest plan cost with cheapest flow



- Introduce time steps

Conclusion

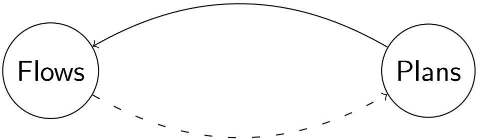
- Estimate cheapest plan cost with cheapest flow



- Introduce time steps
- Repetition in last time layer

Conclusion

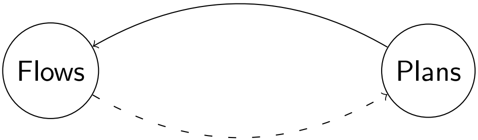
- Estimate cheapest plan cost with cheapest flow



- Introduce time steps
- Repetition in last time layer

Conclusion

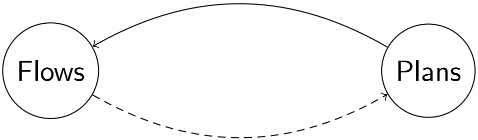
- Estimate cheapest plan cost with cheapest flow



- Introduce time steps
- Repetition in last time layer
- Synchronization per time step

Conclusion

- Estimate cheapest plan cost with cheapest flow



- Introduce time steps
- Repetition in last time layer
- Synchronization per time step