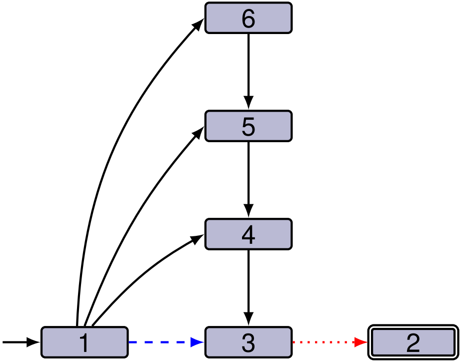
New Refinement Strategies for Cartesian Abstractions

David Speck [1,2]   **Jendrik Seipp** [2]

[1] University of Freiburg, Germany

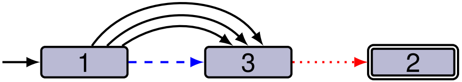
[2] Linköping University, Sweden

Motivation

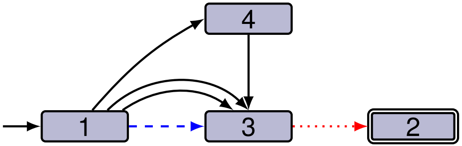


Motivation

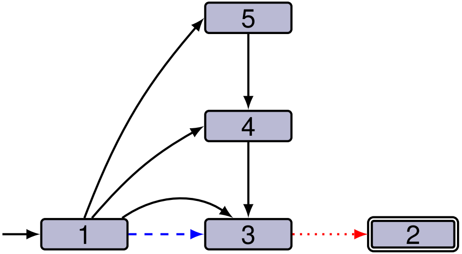
Motivation



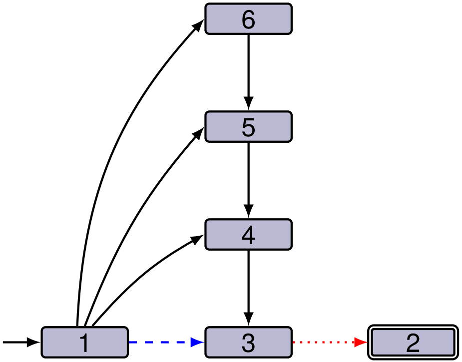
Motivation



Motivation



Motivation



New refinement strategies

Counterexample-Guided Cartesian Abstraction Refinement (CEGAR)

- Repeatedly find counterexamples
  - I.e., abstract plans that fail for the concrete task
  - Repair the flaw by splitting a state

New refinement strategies

Counterexample-Guided Cartesian Abstraction Refinement (CEGAR)

- Repeatedly find counterexamples
  - I.e., abstract plans that fail for the concrete task
  - Repair the flaw by splitting a state
- Previously: choose an arbitrary optimal abstract plan

New refinement strategies

Counterexample-Guided Cartesian Abstraction Refinement (CEGAR)

- Repeatedly find counterexamples
  - I.e., abstract plans that fail for the concrete task
  - Repair the flaw by splitting a state
- Previously: choose an arbitrary optimal abstract plan

Contribution

- Consider all optimal abstract plans
- New refinement strategies = flaw + split selection strategies

How can we find all flaws?

Flaw Search

- Consider all optimal abstract plans
- Depth-first search in the concrete transition system
- Consider only f-optimal transitions of the abstraction
- Collect all encountered flaws
- Goal state expanded $\rightsquigarrow$ optimal concrete plan
- Open list empty $\rightsquigarrow$ all flaws found

Flaw selection strategy

FIRST strategy [SH18]

- Considers an arbitrary optimal abstract plan π
- Selects the first flaw found for π
- Returns π if it works for the concrete task

Flaw selection strategy

FIRST strategy [SH18]

- Considers an arbitrary optimal abstract plan π
- Selects the first flaw found for π
- Returns π if it works for the concrete task

MINH strategy

- Considers all optimal abstract plans
- Selects a flaw with the lowest h-value $\rightsquigarrow$ close to the goal
- Returns a concrete solution if one exists

Flaw selection strategy

FIRST strategy [SH18]

- Considers an arbitrary optimal abstract plan π
- Selects the first flaw found for π
- Returns π if it works for the concrete task

MINH strategy

- Considers all optimal abstract plans
- Selects a flaw with the lowest h-value $\rightsquigarrow$ close to the goal
- Returns a concrete solution if one exists

MAXH strategy

- Considers all optimal abstract plans
- Selects a flaw with the highest h-value $\rightsquigarrow$ far to the goal
- Returns a concrete solution if no flaw exists

Flaw selection strategy – Batch refinement

- Searching for all flaws in every step can be expensive

⇝ Repair several flaws at once

UNI FREIBURG

Flaw selection strategy – Batch refinement

- Searching for all flaws in every step can be expensive

⇝ Repair several flaws at once

BATCH strategy

- Search for all flaws
- Return a concrete plan if the flaw search found one
- Iteratively repairs the flaw with the lowest h-value

Flaw selection strategy – Batch refinement

- Searching for all flaws in every step can be expensive

⇝ Repair several flaws at once

Batch strategy

- Search for all flaws
- Return a concrete plan if the flaw search found one
- Iteratively repairs the flaw with the lowest h-value
- Attention: repairing a flaw can change the abstraction!
- Check if h-values of flaws have changed
- Repair all flaws that maintain the h-value

Flaw selection strategy – Theoretical results

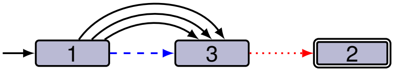
FIRST/MAXH can lead to arbitrarily larger abstractions until a concrete solution is found, compared to MINH/BATCH.

Flaw selection strategy – Theoretical results

FIRST/MAXH can lead to arbitrarily larger abstractions until a concrete solution is found, compared to MINH/BATCH.

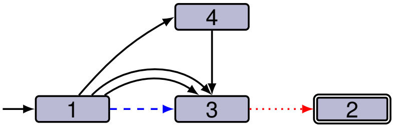
Flaw selection strategy – Theoretical results

FIRST/MAXH can lead to arbitrarily larger abstractions until a concrete solution is found, compared to MINH/BATCH.



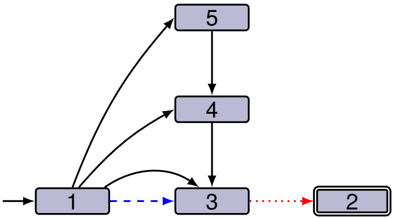
Flaw selection strategy – Theoretical results

FIRST/MAXH can lead to arbitrarily larger abstractions until a concrete solution is found, compared to MINH/BATCH.



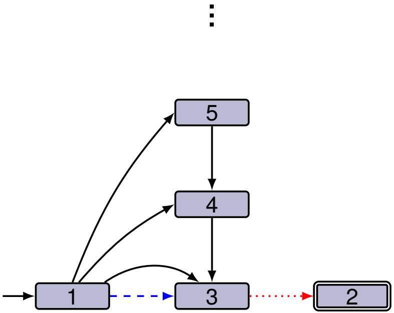
Flaw selection strategy – Theoretical results

FIRST/MAXH can lead to arbitrarily larger abstractions until a concrete solution is found, compared to MINH/BATCH.



Flaw selection strategy – Theoretical results

FIRST/MAXH can lead to arbitrarily larger abstractions until a concrete solution is found, compared to MINH/BATCH.



Split selection strategy

- Usually many different ways to repair a flaw
- I.e., how to split the abstract state

Split selection strategy

- Usually many different ways to repair a flaw
- I.e., how to split the abstract state

MaxRefined strategy [SH18]

- Splits the domain of the variable that has been refined the most

Split selection strategy

- Usually many different ways to repair a flaw
- I.e., how to split the abstract state

MaxRefined strategy [SH18]

- Splits the domain of the variable that has been refined the most

Cover strategy

- Consider multiple flaws at once
- Chooses split that addresses the most flaws at once

UNI FREIBURG

Experiments

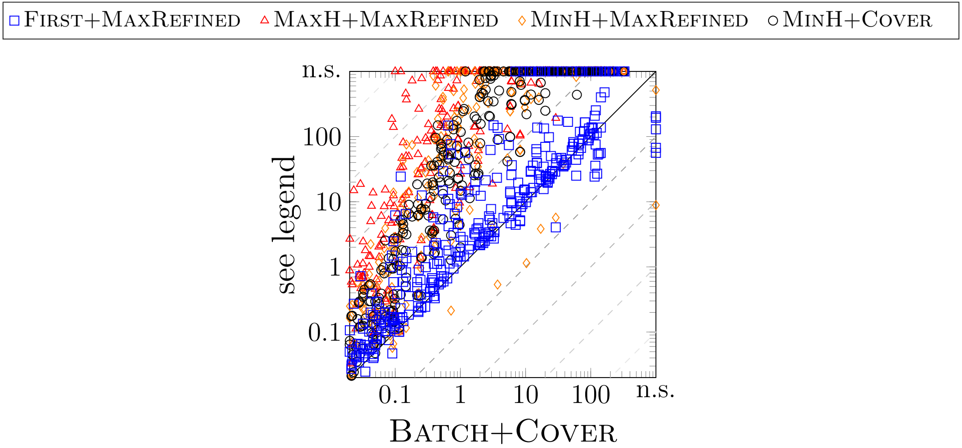
- Implemented refinement strategies in Scorpion [Sei18]
- Planning tasks from optimal track of IPCs
- 15 min and 3.5GB for CEGAR
- 30 min and 4GB memory for $A^* + h^{\text{CEGAR}}$

UNI FREIBURG

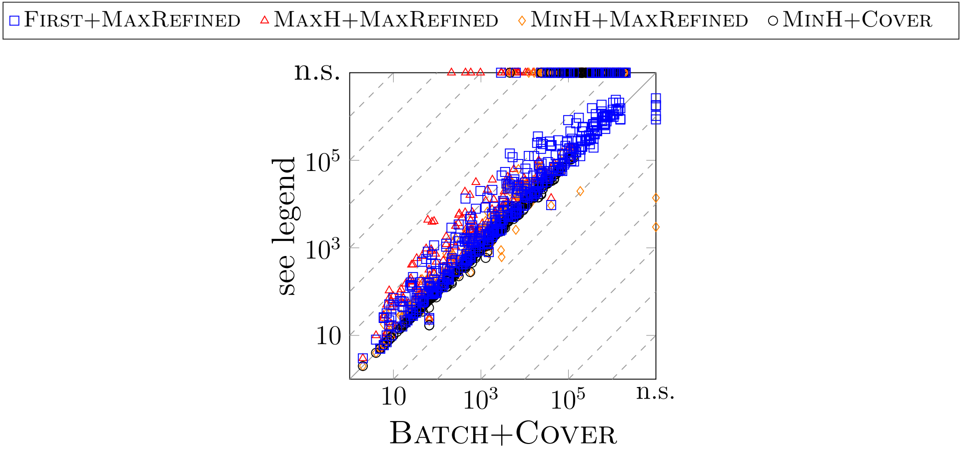
Experiments – Coverage

	MaxRefined			Cover	
Strategy	First	MaxH	MinH	MinH	Batch
CEGAR	499	345	382	393	**534**

Experiments – Runtime



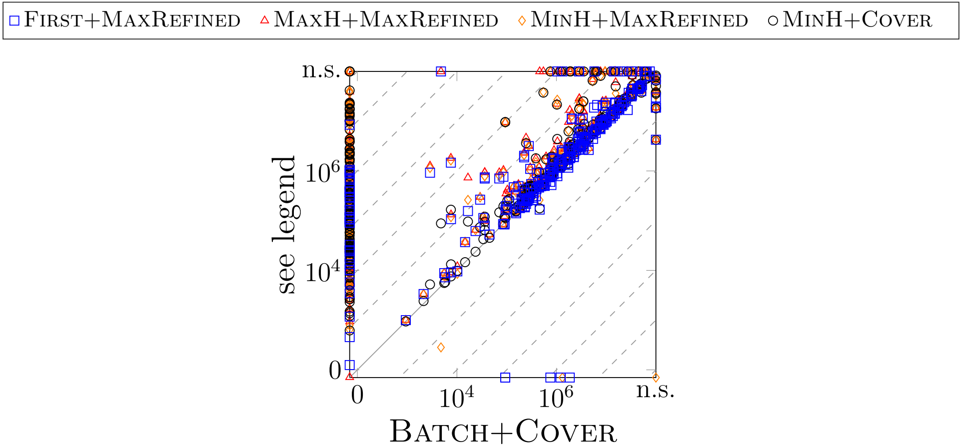
Experiments – Abstraction Size



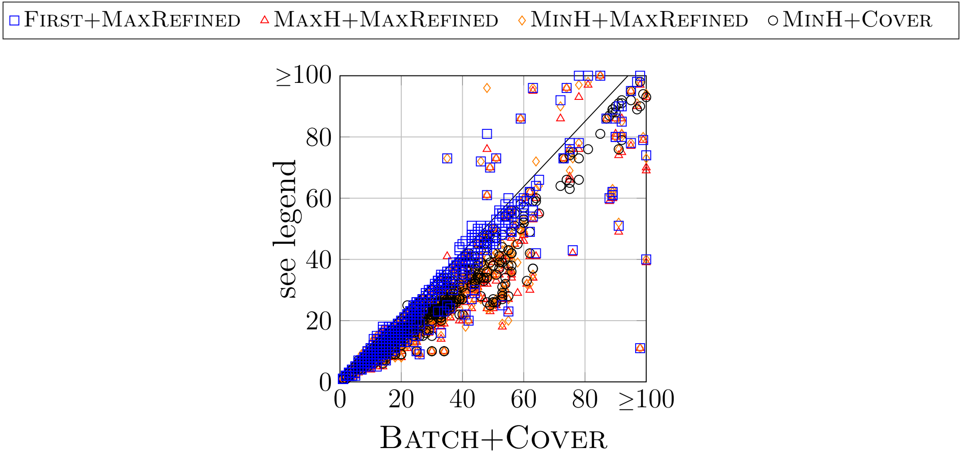
Experiments – Coverage

	MaxRefined			Cover	
Strategy	First	MaxH	MinH	MinH	Batch
CEGAR	499	345	382	393	**534**
$A^* + h^{\text{CEGAR}}$	802	780	792	797	**812**

Experiments – Expansions



Experiments – Heuristic Accuracy



Conclusion

- New refinement strategies for CEGAR $\rightsquigarrow$ flaw + split selection strategies
- Flaw Search $\rightsquigarrow$ determine all flaws simultaneously

Conclusion

- New refinement strategies for CEGAR ⇝ flaw + split selection strategies
- Flaw Search ⇝ determine all flaws simultaneously

Findings

- Refine states close to the goal
- Split states such that multiple flaws are repaired at once
- Repair as many flaws as possible in one step (batch)

Conclusion

- New refinement strategies for CEGAR $\rightsquigarrow$ flaw + split selection strategies
- Flaw Search $\rightsquigarrow$ determine all flaws simultaneously

Findings

- Refine states close to the goal
- Split states such that multiple flaws are repaired at once
- Repair as many flaws as possible in one step (batch)

Future work

- Compare refinement strategies for multiple Cartesian abstractions

References I

[Sei18] Jendrik Seipp, *Fast Downward Scorpion*, IPC-9 Planner Abstracts, 2018, pp. 77–79.

[SH18] Jendrik Seipp and Malte Helmert, *Counterexample-guided Cartesian abstraction refinement for classical planning*, JAIR **62** (2018), 535–577.