Hybrid Planning – From Theory to Practice

Pascal Bercher
Supervisor: *Susanne Biundo*

Institute of Artificial Intelligence,
Ulm University, Germany

14. July 2019

ulm university universität uulm

Companion-Technology

Today's every-day technical devices are:

- complex with a wide functionality spectrum and
- they often require expert knowledge.

Companion-Technology

Vision:

Future technical systems are *Companion*-systems
i.e., competent assistants that:

- support their users fully autonomously,
- react to unforeseen events, and
- act transparently due to explanations.

www.companion-technology.org

S. Biundo, D. Höller, B. Schattenberg, and **P. Bercher**. "Companion-Technology: An Overview". In: ***Künstliche Intelligenz*** 30.1 (2016), pp. 11–20

Why *Hierarchical* Planning?

To achieve this, we use *hybrid planning*, a hierarchical planning approach.

Hierarchical planning is concerned with task decomposition:

Refine abstract tasks into primitive ones by adhering a given task hierarchy. — *But why should we do so?*

Why *Hierarchical* Planning?

To achieve this, we use *hybrid planning*, a hierarchical planning approach.

Hierarchical planning is concerned with task decomposition:

Refine abstract tasks into primitive ones by adhering a given task hierarchy. — *But why should we do so?* E.g., because:

- Many domain experts might want to model it that way!

Why *Hierarchical* Planning?

To achieve this, we use *hybrid planning*, a hierarchical planning approach.

Hierarchical planning is concerned with task decomposition:

Refine abstract tasks into primitive ones by adhering a given task hierarchy. — *But why should we do so?* E.g., because:

- Many domain experts might want to model it that way!
- Plans can be presented *more abstract* by relying on task hierarchies.

Why *Hierarchical* Planning?

To achieve this, we use *hybrid planning*, a hierarchical planning approach.

Hierarchical planning is concerned with task decomposition:

Refine abstract tasks into primitive ones by adhering a given task hierarchy. — *But why should we do so?* E.g., because:

- Many domain experts might want to model it that way!
- Plans can be presented *more abstract* by relying on task hierarchies.
- More control on the generated plans.

Why *Hierarchical* Planning?

To achieve this, we use *hybrid planning*, a hierarchical planning approach.

Hierarchical planning is concerned with task decomposition:

Refine abstract tasks into primitive ones by adhering a given task hierarchy. — *But why should we do so?* E.g., because:

- Many domain experts might want to model it that way!
- Plans can be presented *more abstract* by relying on task hierarchies.
- More control on the generated plans.
- Search space restriction.

Problem Definition & Solution Criteria

$$\mathcal{P} = (V, P, \delta, C, M, s_I, c_I)$$

- V a set of state variables.

T. Geier and **P. Bercher**. "On the Decidability of HTN Planning with Task Insertion". In: ***IJCAI 2011***, pp. 1955–1961

Problem Definition & Solution Criteria

primitive
tasks

compound
tasks

$$\mathcal{P} = (V, P, \delta, C, M, s_I, c_I)$$

- V a set of state variables.
- P a set of primitive task names.
- $\delta : P \rightarrow (2^V)^3$ the task name mapping.
- C a set of compound task names.

T. Geier and **P. Bercher**. "On the Decidability of HTN Planning with Task Insertion". In: ***IJCAI 2011***, pp. 1955–1961

Problem Definition & Solution Criteria

$$\mathcal{P} = (V, P, \delta, C, M, s_I, c_I)$$

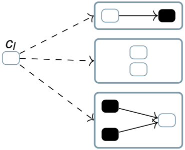
- V a set of state variables.
- P a set of primitive task names.
- $\delta : P \rightarrow (2^V)^3$ the task name mapping.
- C a set of compound task names.
- $c_I \in C$ the initial task.

A solution task network *tn* must:

- be a refinement of c_I,

T. Geier and **P. Bercher**. "On the Decidability of HTN Planning with Task Insertion". In: ***IJCAI 2011***, pp. 1955–1961

Problem Definition & Solution Criteria



$$\mathcal{P} = (V, P, \delta, C, M, s_I, c_I)$$

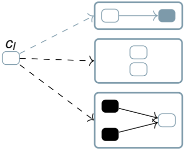
- V a set of state variables.
- P a set of primitive task names.
- $\delta : P \to (2^V)^3$ the task name mapping.
- C a set of compound task names.
- $c_I \in C$ the initial task.
- $M \subseteq C \times 2^{TN}$ the methods.

A solution task network *tn* must:

- be a refinement of c_I,
- only contain primitive tasks, and

T. Geier and **P. Bercher**. "On the Decidability of HTN Planning with Task Insertion". In: ***IJCAI 2011***, pp. 1955–1961

Problem Definition & Solution Criteria



$$\mathcal{P} = (V, P, \delta, C, M, s_I, c_I)$$

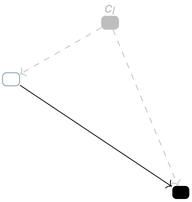
- V a set of state variables.
- P a set of primitive task names.
- $\delta : P \to (2^V)^3$ the task name mapping.
- C a set of compound task names.
- $c_I \in C$ the initial task.
- $M \subseteq C \times 2^{TN}$ the methods.

A solution task network *tn* must:

- be a refinement of c_I,
- only contain primitive tasks, and

T. Geier and **P. Bercher**. "On the Decidability of HTN Planning with Task Insertion". In: ***IJCAI 2011***, pp. 1955–1961

Problem Definition & Solution Criteria



$$\mathcal{P} = (V, P, \delta, C, M, s_I, c_I)$$

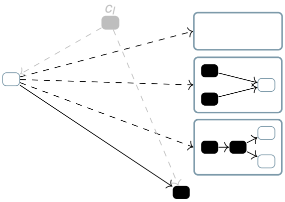
- V a set of state variables.
- P a set of primitive task names.
- $\delta : P \to (2^V)^3$ the task name mapping.
- C a set of compound task names.
- $c_I \in C$ the initial task.
- $M \subseteq C \times 2^{TN}$ the methods.

A solution task network *tn* must:

- be a refinement of c_I,
- only contain primitive tasks, and

T. Geier and **P. Bercher**. "On the Decidability of HTN Planning with Task Insertion". In: ***IJCAI 2011***, pp. 1955–1961

Problem Definition & Solution Criteria



$$\mathcal{P} = (V, P, \delta, C, M, s_I, c_I)$$

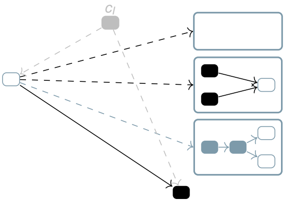
- V a set of state variables.
- P a set of primitive task names.
- $\delta : P \to (2^V)^3$ the task name mapping.
- C a set of compound task names.
- $c_I \in C$ the initial task.
- $M \subseteq C \times 2^{TN}$ the methods.

A solution task network *tn* must:

- be a refinement of c_I,
- only contain primitive tasks, and

T. Geier and **P. Bercher**. "On the Decidability of HTN Planning with Task Insertion". In: ***IJCAI 2011***, pp. 1955–1961

Problem Definition & Solution Criteria



$$\mathcal{P} = (V, P, \delta, C, M, s_I, c_I)$$

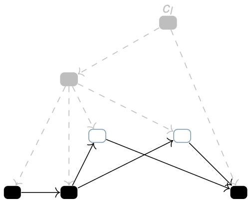
- V a set of state variables.
- P a set of primitive task names.
- $\delta : P \to (2^V)^3$ the task name mapping.
- C a set of compound task names.
- $c_I \in C$ the initial task.
- $M \subseteq C \times 2^{TN}$ the methods.

A solution task network *tn* must:

- be a refinement of c_I,
- only contain primitive tasks, and

T. Geier and **P. Bercher**. "On the Decidability of HTN Planning with Task Insertion". In: ***IJCAI 2011***, pp. 1955–1961

Problem Definition & Solution Criteria



T. Geier and **P. Bercher**. “On the Decidability of HTN Planning with Task Insertion”. In: ***IJCAI 2011***, pp. 1955–1961

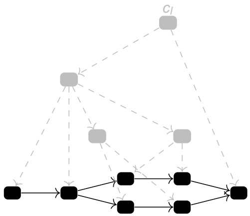
$$\mathcal{P} = (V, P, \delta, C, M, s_I, c_I)$$

- V a set of state variables.
- P a set of primitive task names.
- $\delta : P \to (2^V)^3$ the task name mapping.
- C a set of compound task names.
- $c_I \in C$ the initial task.
- $M \subseteq C \times 2^{TN}$ the methods.

A solution task network *tn* must:

- be a refinement of c_I,
- only contain primitive tasks, and

Problem Definition & Solution Criteria



$$\mathcal{P} = (V, P, \delta, C, M, s_I, c_I)$$

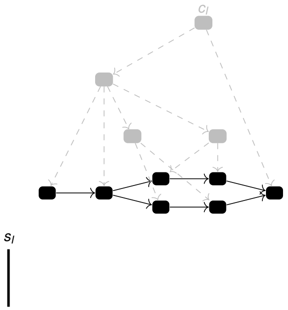
- V a set of state variables.
- P a set of primitive task names.
- $\delta : P \to (2^V)^3$ the task name mapping.
- C a set of compound task names.
- $c_I \in C$ the initial task.
- $M \subseteq C \times 2^{TN}$ the methods.

T. Geier and **P. Bercher**. "On the Decidability of HTN Planning with Task Insertion". In: ***IJCAI 2011***, pp. 1955–1961

A solution task network *tn* must:

- be a refinement of c_I,
- only contain primitive tasks, and

Problem Definition & Solution Criteria



T. Geier and **P. Bercher**. "On the Decidability of HTN Planning with Task Insertion". In: ***IJCAI 2011***, pp. 1955–1961

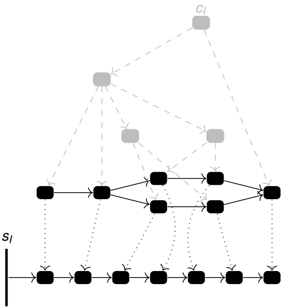
$$\mathcal{P} = (V, P, \delta, C, M, s_I, c_I)$$

- V a set of state variables.
- P a set of primitive task names.
- $\delta : P \to (2^V)^3$ the task name mapping.
- C a set of compound task names.
- $c_I \in C$ the initial task.
- $M \subseteq C \times 2^{TN}$ the methods.
- $s_I \subseteq V$ the initial state.

A solution task network *tn* must:

- be a refinement of c_I,
- only contain primitive tasks, and

Problem Definition & Solution Criteria



T. Geier and **P. Bercher**. "On the Decidability of HTN Planning with Task Insertion". In: ***IJCAI 2011***, pp. 1955–1961

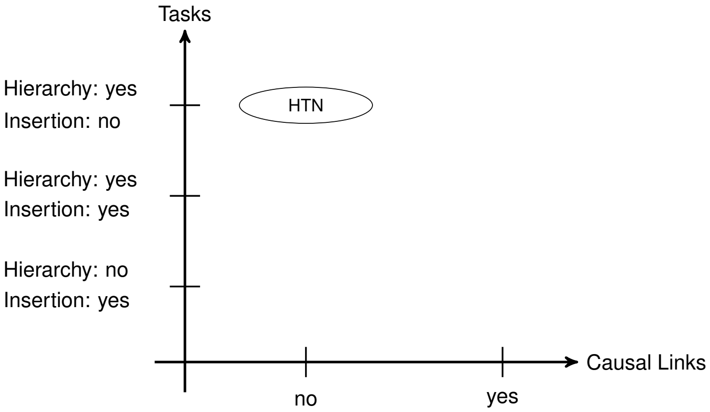
$$\mathcal{P} = (V, P, \delta, C, M, s_I, c_I)$$

- V a set of state variables.
- P a set of primitive task names.
- $\delta : P \rightarrow (2^V)^3$ the task name mapping.
- C a set of compound task names.
- $c_I \in C$ the initial task.
- $M \subseteq C \times 2^{TN}$ the methods.
- $s_I \subseteq V$ the initial state.

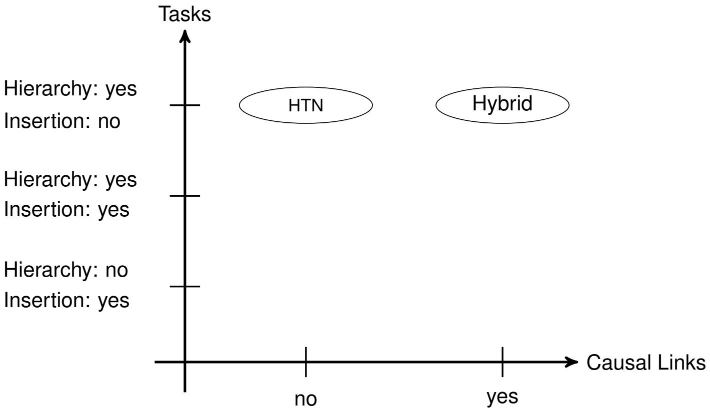
A solution task network *tn* must:

- be a refinement of c_I,
- only contain primitive tasks, and
- have an executable linearization.

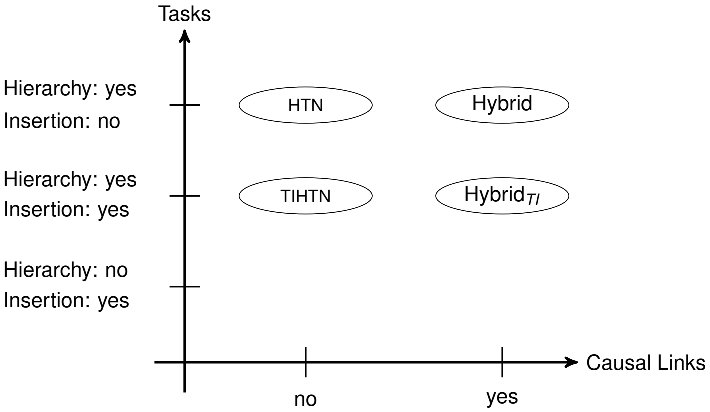
Considered Problem Classes and Complexity Investigations



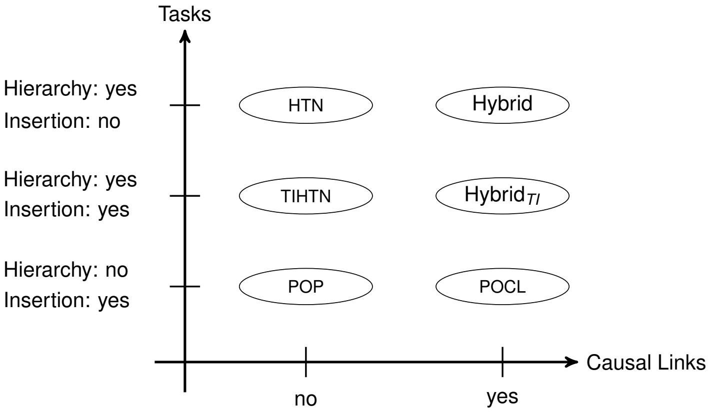
Considered Problem Classes and Complexity Investigations



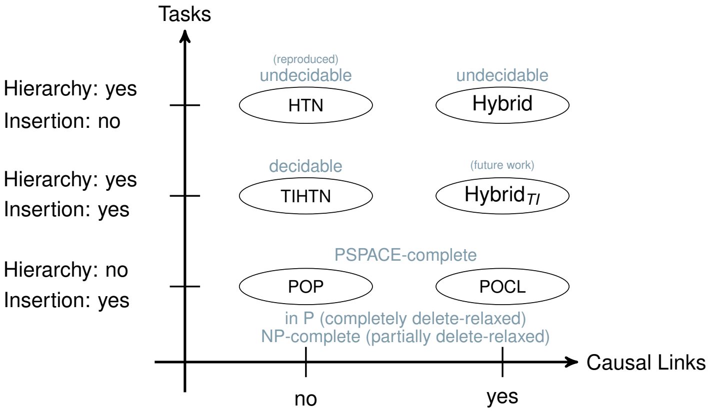
Considered Problem Classes and Complexity Investigations



Considered Problem Classes and Complexity Investigations



Considered Problem Classes and Complexity Investigations



P. Bercher, D. Höller, G. Behnke, and S. Biundo. “More than a Name? On Implications of Preconditions and Effects of Compound HTN Planning Tasks”. In: ***ECAI 2016***, pp. 225–233

P. Bercher, T. Geier, F. Richter, and S. Biundo. “On Delete Relaxation in Partial-Order Causal-Link Planning”. In: ***ICTAI 2013***, pp. 674–681

T. Geier and **P. Bercher**. “On the Decidability of HTN Planning with Task Insertion”. In: ***IJCAI 2011***, pp. 1955–1961

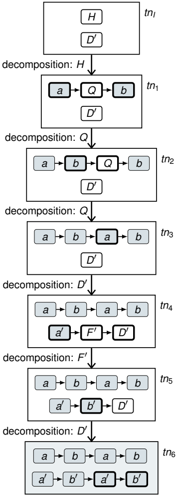
On the (Un)Decidability of HTN Planning

Theorem: HTN planning is undedicable (Erol et al., 1994)

But what about TIHTN planning?

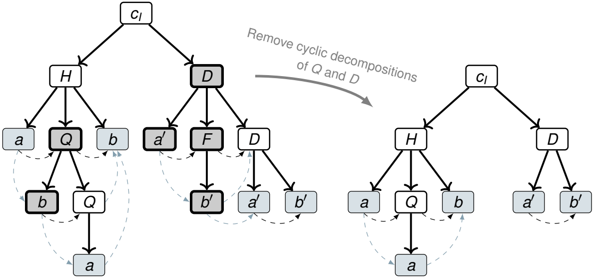
I.e., makes the capability to insert tasks arbitrarily the plan existence problem harder? Or easier? Does it change anything?

HTN Planning with Task Insertion (TIHTN Planning)

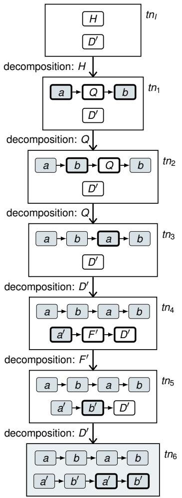


Theorem: Task Insertion makes HTN planning decidable

Idea: Restrict to *acyclic* decompositions, fill the rest with task insertion, and verify.

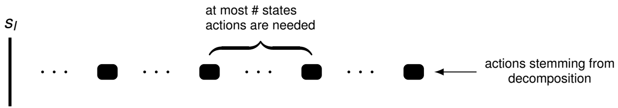


HTN Planning with Task Insertion (TIHTN Planning)



Theorem: Task Insertion makes HTN planning decidable

1. *Step:* Guess an acyclic decomposition.
2. *Step:* Guess the actions and orderings to be inserted.



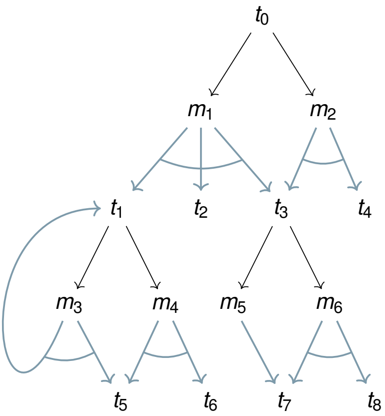
In summary:

If there exists a solution to an TIHTN planning problem, then there exists a solution based on:

- an acyclic decomposition tree and
- at most exponentially many inserted actions.

TDG-based Heuristics

A Task Decomposition Graph (TDG) represents the decomposition structure:



A TDG is a (possibly cyclic) bipartite graph $\mathcal{G} = \langle N_T, N_M, E_{(T,M)}, E_{(M,T)} \rangle$ with

- N_T, the task nodes,
- N_M, the method nodes,
- $E_{(T,M)}$, the task edges,
- $E_{(M,T)}$, the method edges.

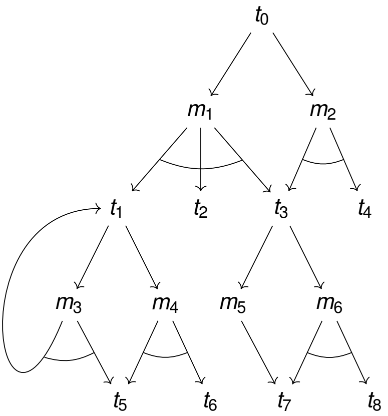
P. Bercher, G. Behnke, D. Höller, and S. Biundo. “An Admissible HTN Planning Heuristic”. In: ***IJCAI 2017***, pp. 480–488

P. Bercher, S. Keen, and S. Biundo. “Hybrid Planning Heuristics Based on Task Decomposition Graphs”. In: ***SoCS 2014***, pp. 35–43

M. Elkawkagy, **P. Bercher**, B. Schattenberg, and S. Biundo. “Improving Hierarchical Planning Performance by the Use of Landmarks”. In: ***AAAI 2012***, pp. 1763–1769

TDG-based Heuristics

A Task Decomposition Graph (TDG) represents the decomposition structure:



How to use the TDG to calculate an heuristic estimate?

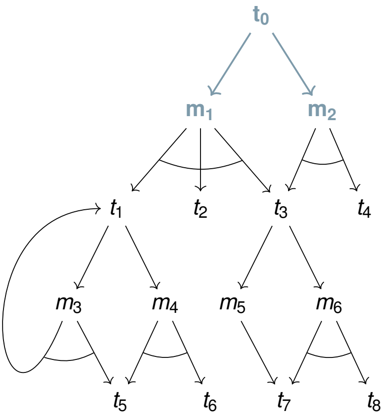
Step 1:
Calculate the TDG in a preprocessing step.

Step 2:
Calculate heuristic $h(t)$ for each task t in TDG (still via preprocessing).

Step 3:
For a search node (task network) P and its task identifiers T, calculate $h(P) := \sum_{t \in T} h(t)$.

TDG-based Heuristics

A Task Decomposition Graph (TDG) represents the decomposition structure:

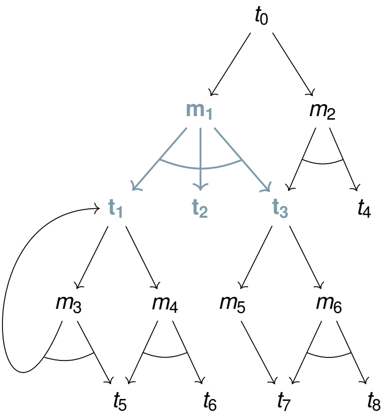


Example:

$$h_T(t_0) = min\{h_M(m_1), h_M(m_2)\}$$

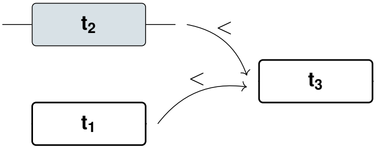
TDG-based Heuristics

A Task Decomposition Graph (TDG) represents the decomposition structure:



Example:

Method $m_1 = (t_0, tn)$ with task network tn:

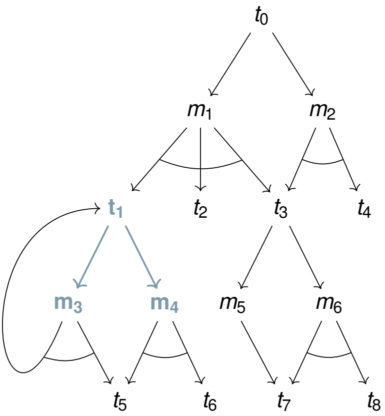


$$h_M(m_1) = \sum_{t_i \in \{t_1, t_2, t_3\}} h_T(t_i)$$

$$= h_T(t_1) + cost(t_2) + h_T(t_3)$$

TDG-based Heuristics

A Task Decomposition Graph (TDG) represents the decomposition structure:

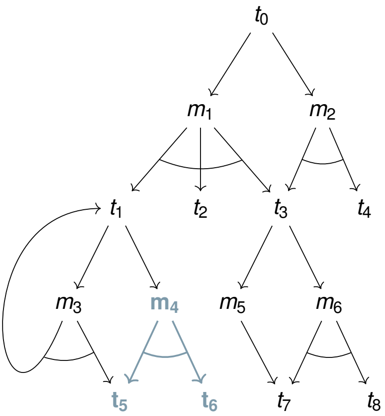


Example:

$$h_T(t_1) = min\{h_M(m_3), h_M(m_4)\}$$

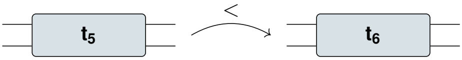
TDG-based Heuristics

A Task Decomposition Graph (TDG) represents the decomposition structure:



Example:

Method $m_4 = (t_1, tn)$ with task network tn:



$$h_M(m_4) = \sum_{t_i \in \{t_5, t_6\}} h_T(t_i)$$
$$= h_T(t_5) + h_T(t_6)$$
$$= cost(t_5) + cost(t_6)$$

Further Results in Heuristic Search

- Improvement of a plan space-based planning system (PANDA).
- Co-development of landmarks in hierarchical planning.
- Development of a POCL heuristic that can exploit the pruning power of causal links.
- Development of a technique that allows using state-based heuristics in POCL planning.

Planning-based Assistance Systems

We integrated our hierarchical planner and various *user-centered* planning capabilities into a running assistance system.

P. Bercher, D. Höller, G. Behnke, and S. Biundo. "User-Centered Planning". In: ***Companion Technology – A Paradigm Shift in Human-Technology Interaction***. Ed. by S. Biundo and A. Wendemuth. Cognitive Technologies. Springer, 2017. Chap. 5

P. Bercher et al. "A Planning-based Assistance System for Setting Up a Home Theater". In: ***AAAI 2015***, pp. 4264–4265

P. Bercher et al. "Plan, Repair, Execute, Explain – How Planning Helps to Assemble your Home Theater". In: ***ICAPS 2014***, pp. 386–394

S. Biundo, **P. Bercher**, T. Geier, F. Müller, and B. Schattenberg. "Advanced user assistance based on AI planning". In: ***Cognitive Systems Research*** 12.3-4 (Apr. 2011). Special Issue on Complex Cognition

Planning-based Assistance Systems

We integrated our hierarchical planner and various *user-centered* planning capabilities into a running assistance system.

- We can provide detailed step-by-step instructions.

P. Bercher, D. Höller, G. Behnke, and S. Biundo. "User-Centered Planning". In: ***Companion Technology – A Paradigm Shift in Human-Technology Interaction***. Ed. by S. Biundo and A. Wendemuth. Cognitive Technologies. Springer, 2017. Chap. 5

P. Bercher et al. "A Planning-based Assistance System for Setting Up a Home Theater". In: ***AAAI 2015***, pp. 4264–4265

P. Bercher et al. "Plan, Repair, Execute, Explain – How Planning Helps to Assemble your Home Theater". In: ***ICAPS 2014***, pp. 386–394

S. Biundo, **P. Bercher**, T. Geier, F. Müller, and B. Schattenberg. "Advanced user assistance based on AI planning". In: ***Cognitive Systems Research*** 12.3-4 (Apr. 2011). Special Issue on Complex Cognition

Planning-based Assistance Systems

We integrated our hierarchical planner and various *user-centered* planning capabilities into a running assistance system.

- We can provide detailed step-by-step instructions.
- We can present them in a user-friendly (i.e., meaningful) order.

P. Bercher, D. Höller, G. Behnke, and S. Biundo. "User-Centered Planning". In: ***Companion Technology – A Paradigm Shift in Human-Technology Interaction***. Ed. by S. Biundo and A. Wendemuth. Cognitive Technologies. Springer, 2017. Chap. 5

P. Bercher et al. "A Planning-based Assistance System for Setting Up a Home Theater". In: ***AAAI 2015***, pp. 4264–4265

P. Bercher et al. "Plan, Repair, Execute, Explain – How Planning Helps to Assemble your Home Theater". In: ***ICAPS 2014***, pp. 386–394

S. Biundo, **P. Bercher**, T. Geier, F. Müller, and B. Schattenberg. "Advanced user assistance based on AI planning". In: ***Cognitive Systems Research*** 12.3-4 (Apr. 2011). Special Issue on Complex Cognition

Planning-based Assistance Systems

We integrated our hierarchical planner and various *user-centered* planning capabilities into a running assistance system.

- We can provide detailed step-by-step instructions.
- We can present them in a user-friendly (i.e., meaningful) order.
- We can repair plans to deal with execution errors.

P. Bercher, D. Höller, G. Behnke, and S. Biundo. "User-Centered Planning". In: ***Companion Technology – A Paradigm Shift in Human-Technology Interaction***. Ed. by S. Biundo and A. Wendemuth. Cognitive Technologies. Springer, 2017. Chap. 5

P. Bercher et al. "A Planning-based Assistance System for Setting Up a Home Theater". In: ***AAAI 2015***, pp. 4264–4265

P. Bercher et al. "Plan, Repair, Execute, Explain – How Planning Helps to Assemble your Home Theater". In: ***ICAPS 2014***, pp. 386–394

S. Biundo, **P. Bercher**, T. Geier, F. Müller, and B. Schattenberg. "Advanced user assistance based on AI planning". In: ***Cognitive Systems Research*** 12.3-4 (Apr. 2011). Special Issue on Complex Cognition

Planning-based Assistance Systems

We integrated our hierarchical planner and various *user-centered* planning capabilities into a running assistance system.

- We can provide detailed step-by-step instructions.
- We can present them in a user-friendly (i.e., meaningful) order.
- We can repair plans to deal with execution errors.
- We can explain plans to establish trust and transparency.

P. Bercher, D. Höller, G. Behnke, and S. Biundo. "User-Centered Planning". In: ***Companion Technology – A Paradigm Shift in Human-Technology Interaction***. Ed. by S. Biundo and A. Wendemuth. Cognitive Technologies. Springer, 2017. Chap. 5

P. Bercher et al. "A Planning-based Assistance System for Setting Up a Home Theater". In: ***AAAI 2015***, pp. 4264–4265

P. Bercher et al. "Plan, Repair, Execute, Explain – How Planning Helps to Assemble your Home Theater". In: ***ICAPS 2014***, pp. 386–394

S. Biundo, **P. Bercher**, T. Geier, F. Müller, and B. Schattenberg. "Advanced user assistance based on AI planning". In: ***Cognitive Systems Research*** 12.3-4 (Apr. 2011). Special Issue on Complex Cognition

Planning-based Assistance Systems

We integrated our hierarchical planner and various *user-centered* planning capabilities into a running assistance system.

- We can provide detailed step-by-step instructions.
- We can present them in a user-friendly (i.e., meaningful) order.
- We can repair plans to deal with execution errors.
- We can explain plans to establish trust and transparency.
- We conducted an empirical evaluation of the system and plan explanations with 59 test subjects.

P. Bercher, D. Höller, G. Behnke, and S. Biundo. "User-Centered Planning". In: ***Companion Technology – A Paradigm Shift in Human-Technology Interaction***. Ed. by S. Biundo and A. Wendemuth. Cognitive Technologies. Springer, 2017. Chap. 5

P. Bercher et al. "A Planning-based Assistance System for Setting Up a Home Theater". In: ***AAAI 2015***, pp. 4264–4265

P. Bercher et al. "Plan, Repair, Execute, Explain – How Planning Helps to Assemble your Home Theater". In: ***ICAPS 2014***, pp. 386–394

S. Biundo, **P. Bercher**, T. Geier, F. Müller, and B. Schattenberg. "Advanced user assistance based on AI planning". In: ***Cognitive Systems Research*** 12.3-4 (Apr. 2011). Special Issue on Complex Cognition

Home Theater Assembly Assistant

Sink devices:

- Television (requires video)
- Amplifier (requires audio)

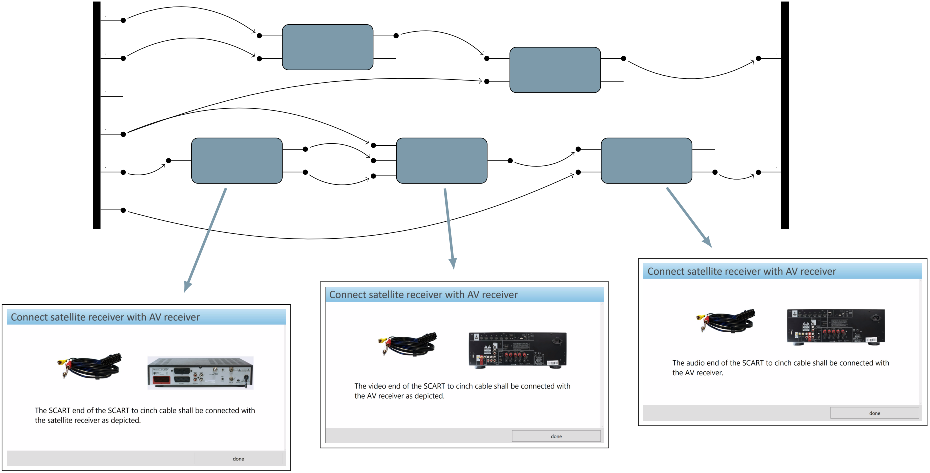
Source devices:

- Blu-ray player
- Satellite receiver
  (both produce audio & video)

P. Bercher et al. "A Planning-based Assistance System for Setting Up a Home Theater". In: ***AAAI 2015***, pp. 4264–4265

P. Bercher et al. "Plan, Repair, Execute, Explain – How Planning Helps to Assemble your Home Theater". In: ***ICAPS 2014***, pp. 386–394

Home Theater Assembly Assistant – The Role of Planning



Home Theater Assembly Assistant – Demonstration

F. Honold, **P. Bercher** et al. "Companion-Technology: Towards User- and Situation-Adaptive Functionality of Technical Systems". In: ***IE 2014***, pp. 378–381

Summary of (Mentioned) Core Contributions

Theoretical Foundations

- New Formalization for HTN Planning
- Investigation of Task Insertion
- Further Complexity Studies for Hybrid Planning and POCL Planning

Heuristic Search

- Novel (admissible) heuristics for HTN planning, hybrid planning, and POCL planning

Practical Application

- (Co-)Realization of an assistant supporting in the assembly of a home theater realizing many *user-centered planning capabilities*

Special Thanks To

Susanne Biundo	Gregor Behnke	Daniel Höller
		

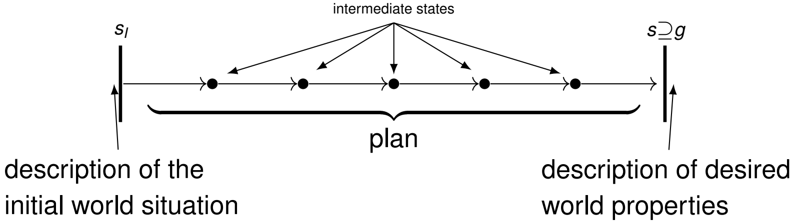
Planning in a Nutshell

In classical planning, the "base case", a problem consists of:

- Given an initial state s_I,
- a portfolio of the system's available actions, and
- a goal description g.

What do we want?

$\rightarrow$ Find a *plan* that transforms s_I into g.



Classical Problems, Formally

A classical planning problem $\mathcal{P} = \langle V, A, s_I, g \rangle$ consists of:

- V,a finite set of *state variables*, $S = 2^V$ is the *state space*.
- A is a finite set of actions. Each action $a \in A$ is a tuple $(pre, add, del) \in 2^V \times 2^V \times 2^V$ consisting of a *precondition* and an *add and delete list*.
- $s_I \in S$ is the initial state (complete state description).
- $g \subseteq V$ is the goal description (encodes a set of goal states).

Action application:

- An action $a \in A$ is called *applicable* (or executable) in a state $s \in S$ if and only if $pre(a) \subseteq s$.
- If a is applicable to s, its application results into the successor state $s' = (s \setminus del(a)) \cup add(a)$.
- Definition transfers to action sequences in the canonical way.

On the (Un)Decidability of HTN Planning

Theorem: HTN planning is undedicable (Erol et al., 1994)

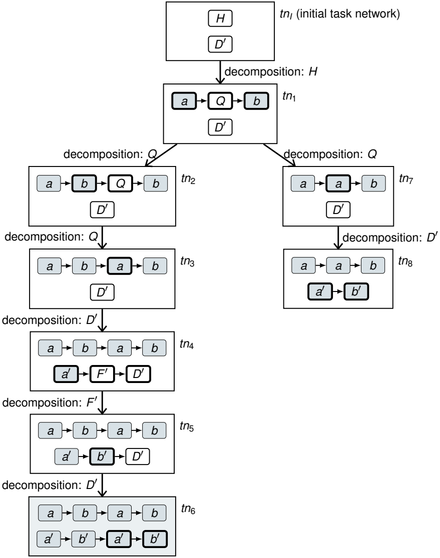
Proof:

Reduction from the language intersection problem of two context-free grammars: given G and G', is there a word ω in both languages $L(G) \cap L(G')$?

Construct an HTN planning problem $\mathcal{P}$ that has a solution if and only if the correct answer is *yes*:

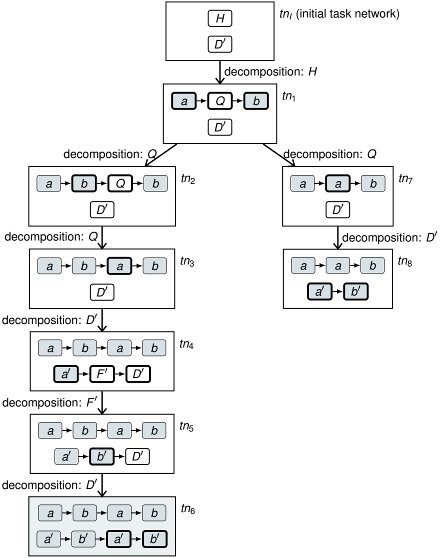
Translate the production rules to decomposition methods. That way only words in $L(G)$ and $L(G')$ can be produced.

Illustration of Grammar Intersection Encoding



In Erol et al.'s encoding a task network is a solution if it contains the same word twice.

Illustration of Grammar Intersection Encoding



In Erol et al.'s encoding a task network is a solution if it contains the same word twice.

Task network tn_6 is a solution!

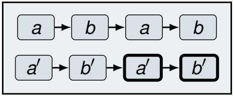
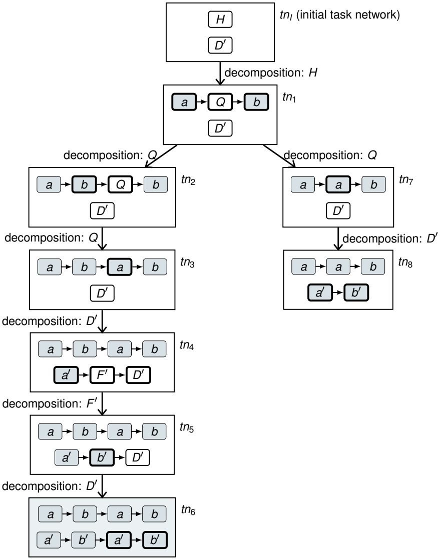


Illustration of Grammar Intersection Encoding



In Erol et al.'s encoding a task network is a solution if it contains the same word twice.

Task network tn_8 is no solution!

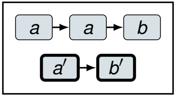
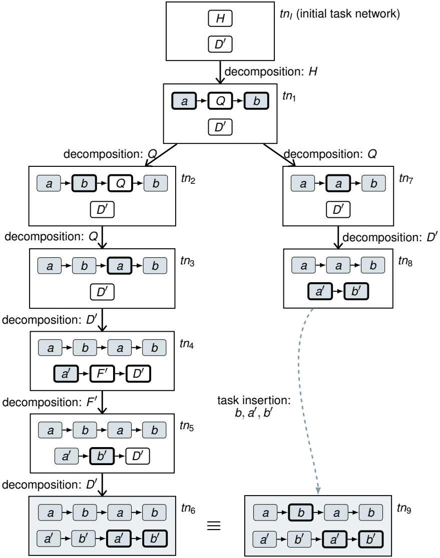
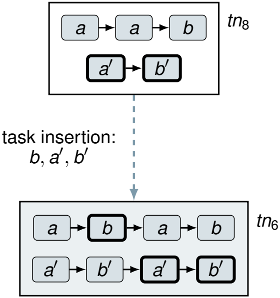


Illustration of Grammar Intersection Encoding

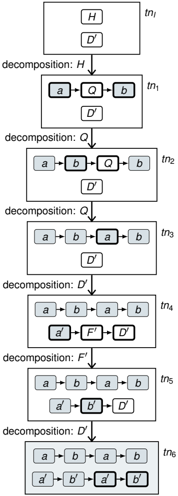


In Erol et al.'s encoding a task network is a solution if it contains the same word twice.

Influence of task insertion:

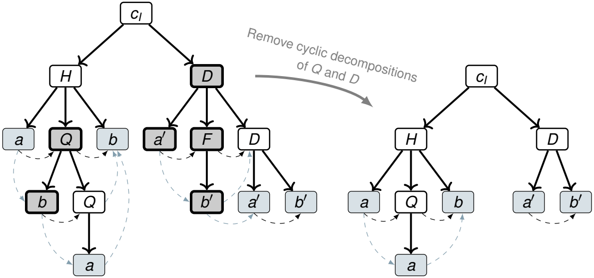


HTN Planning with Task Insertion (TIHTN Planning)

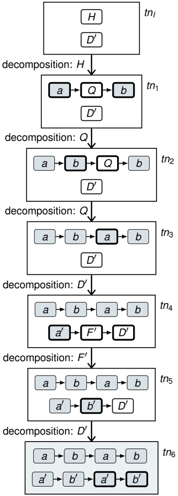


Theorem: Task Insertion makes HTN planning decidable

Idea: Restrict to *acyclic* decompositions, fill the rest with task insertion, and verify.



HTN Planning with Task Insertion (TIHTN Planning)



Theorem: Task Insertion makes HTN planning decidable

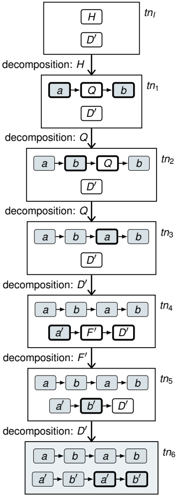
1. Step: Guess an acyclic decomposition:

The guessed decomposition tree describes at most $b^{|C|+1}$ decompositions.

(C = set of compound tasks)
(b = size of largest task network in the model)

Verify in $O(b^{|C|+1})$ whether the tree describes a correct sequence of decompositions.

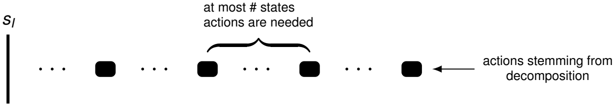
HTN Planning with Task Insertion (TIHTN Planning)



Theorem: Task Insertion makes HTN planning decidable

2. Step: Guess the actions and orderings to be inserted.

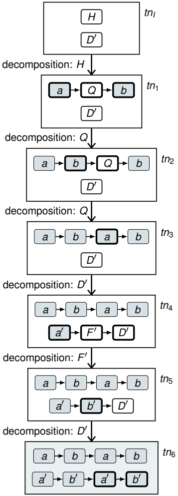
The (guessed) decomposition tree results into a task network with at most $\leq b^{|C|+1}$ tasks.



Between each two actions, at most $2^{|V|}$ actions need to be inserted to achieve the next precondition.

($|V|$ = number of state variables)
($2^{|V|}$ = number of states)

HTN Planning with Task Insertion (TIHTN Planning)



Theorem: Task Insertion makes HTN planning decidable

In summary:

If there exists a solution to an TIHTN planning problem, then there exists a solution based on:

- an acyclic decomposition tree and
- at most exponentially many inserted actions.

Goal of Evaluation

We studied:

- The general acceptance of such a system
  (in the example application of the home assembly assistant).
- The influence of plan explanations.

  Our hypothesis: *Plan explanations foster the users' confidence in the correctness of the solutions.*

Experimental Design

- Assembly Task:
  - Television requires video signal
  - Amplifier requires audio signal
- Test subjects received a step-by-step instruction on an iPad, which they had to follow.
- The study was designed as controlled, randomized trial:
  - The *treatment* group received explanations for two plan steps.
  - The *control* group did not – they only received the instructions.

Information About Test Subjects

- 59 test subjects in total.
- Age:
  - $\leq$ 30 years: 19 female, 27 male.
  - $>$ 30 years: 3 female, 7 male.
  - (Three test subjects did not mention their age.)
- Educational background:
  - 26 test subjects had a University degree, 9 were pursuing their doctoral degree (PhD students).
  - 7 test subject did not have a high school degree (German: Abitur).
  - 30 test subjects had a technical background: Computer Science/Engineering, Natural Sciences , Mathematics.

Results I

Test subjects were asked how confident they are that the presented solution is correct.

- Average/Standard deviation on a 5 point Likert scale: 4.50/0.82 (treatment group), 4.66/0.55 (control group) (difference *not* statistically significant)
- Possible causes:
  - Confidence was already very high (4.66 of 5 for control group).
  - Suboptimal experimental design: The explanations were shown without any necessity (they were also not requested).
- Overall perception (summary variable rating aspects like trust, paternalism, appeal, utility, etc.
  - Average/Standard deviation: 26.63/3.67 out of 30.

Results II

- The animated pictures were often (positively) mentioned.
- High correlation between overall impression and judgement of one own's capabilities: Unskilled people liked the system better.
- Women liked the *system* more than men.
- Test subjects with higher educational level liked the *explanations* more.

Results III

The most positive quotations:

- “supports in a useful way”
- “I would prefer this kind of manual to any other”
- “I think this system is useful as it also allows people without prior knowledge to execute the manual.”
- “This system/assistant would be great for my parents :-)”