Merge-and-Shrink Task Reformulation for Classical Planning

Álvaro Torralba, Silvan Sievers

HSDIP 2019
Classical Planning

Definition. A planning task is a 4-tuple $\Pi = (V, A, I, G)$ where:

1. $V$ is a set of state variables, each $v \in V$ with a finite domain $D_v$.
2. $A$ is a set of actions; each $a \in A$ is a triple $(\text{pre}_a, \text{eff}_a, c_a)$, of precondition and effect (partial assignments), and the action's cost $c_a \in \mathbb{R}_0^+$.
3. Initial state $I$ (complete assignment), goal $G$ (partial assignment).

→ Solution (“Plan”): Action sequence mapping $I$ into $s$ s.t. $s \models G$. 
Classical Planning

Definition. A planning task is a 4-tuple $\Pi = (V, A, I, G)$ where:

- $V$ is a set of state variables, each $v \in V$ with a finite domain $D_v$.
- $A$ is a set of actions; each $a \in A$ is a triple $(\text{pre}_a, \text{eff}_a, c_a)$, of precondition and effect (partial assignments), and the action’s cost $c_a \in \mathbb{R}_0^+$.  
- Initial state $I$ (complete assignment), goal $G$ (partial assignment).

$\rightarrow$ Solution (“Plan”): Action sequence mapping $I$ into $s$ s.t. $s \models G$.

Running Example:

- $V = \{T, F\}$ with $D_T = \{A, B, C, D\}$, $D_F = \{0, 1, 2\}$.
- $A = \{\text{drive}(x, x', f, f')\}$
- $I = \{T = A, F = 2\}$
- $G = \{T = D\}$
Accidental Complexity

- $V = \{T, F\}$ with $D_t = \{A, B, C, D\}$, $D_F = \{0, 1, 2\}$.
- $A = \{\text{drive}(x, x', f, f')\}$
- $I = \{T = A, F = 2\}$
- $G = \{T = D\}$

**Accidental complexity**: when solving the problem is harder due to how it is encoded.
Accidental Complexity

- $V = \{T, F\}$ with $D_t = \{A, B, C, D\}$,
  $D_F = \{0, 1, 2\}$.
- $A = \{\text{drive}(x, x', f, f')\}$
- $I = \{T = A, F = 2\}$
- $G = \{T = D\}$

**Accidental complexity**: when solving the problem is harder due to how it is encoded

- $V = \{T, F, E\}$ with $D_t = \{A, B, C, D\}$,
  $D_F = \{0, 1, 2\}$, $D_E = \{\text{on, off}\}$.
- $A = \{\text{drive}(x, x', f, f'), \text{turnon}, \text{turnoff}\}$
- $I = \{T = A, F = 2, E = \text{off}\}$
- $G = \{T = D\}$
Reformulation

Transform the model to get rid of “accidental” complexity
Reformulation

Transform the model to get rid of “accidental” complexity

Task Π
Reformulation

Transform the model to get rid of “accidental” complexity

\[ \text{Task } \Pi \xrightarrow{\text{reformulation } (\rho)} \text{Task } \rho(\Pi) \]
Reformulation

Transform the model to get rid of “accidental” complexity

\[
\text{Task } \Pi \xrightarrow{\text{reformulation (} \rho \text{)}} \text{Task } \rho(\Pi)
\]

\[
\text{solve}
\]

\[
\text{Plan } \pi^\rho
\]
Reformulation

Transform the model to get rid of “accidental” complexity

Task $\Pi$ \xrightarrow{\text{reformulation ($\rho$)}} Task $\rho(\Pi)$

Plan $\pi = \leftarrow \rho (\pi^\rho)$ \xleftarrow{\text{plan reconstruction ($\leftarrow \rho$)}} Plan $\pi^\rho$
Reformulation

Transform the model to get rid of “accidental” complexity

\[
\text{Task } \Pi \xrightarrow{\text{reformulation } (\rho)} \text{Task } \rho(\Pi) \xrightarrow{\text{solve}} \text{Plan } \pi^\rho
\]

Plan \( \pi = \xleftarrow{\rho} (\pi^\rho) \xleftarrow{\text{plan reconstruction } (\xleftarrow{\rho})} \text{Plan } \pi^\rho
\]

Properties:
- **Polynomial:** \( \rho \) and \( \xleftarrow{\rho} \) run in polynomial time in the \(|\Pi|\) and \(|\xleftarrow{\rho} (\pi^\rho)|\).
Reformulation

Transform the model to get rid of “accidental” complexity

Task $\Pi \xrightarrow{\text{reformulation } (\rho)} \text{ Task } \rho(\Pi)$

Plan $\pi = \xleftarrow{\rho} (\pi^\rho)$

Plan reconstruction $\xleftarrow{\rho} (\pi^\rho)$

Properties:
- Polynomial: $\rho$ and $\xleftarrow{\rho}$ run in polynomial time in the $|\Pi|$ and $|\xleftarrow{\rho} (\pi^\rho)|$.
- Optimal: $\pi^\rho$ is optimal for $\rho(\Pi) \Rightarrow \xleftarrow{\rho} (\pi^\rho)$
Abstraction Heuristics

θ
### Abstraction Heuristics

Θ
Abstraction Heuristics

\[ \Theta \xrightarrow{\text{abstraction } \alpha} \Theta^\alpha \]
An abstraction is refinable if a solution for the abstract task can be transformed in polynomial time in a solution for the original task.

\[ \Theta \xrightarrow{\text{abstraction } \alpha} \Theta^\alpha \xrightarrow{\text{solve}} h^\alpha \]
An abstraction is refinable if a solution for the abstract task can be transformed in polynomial time in a solution for the original task.
An abstraction is **refinable** if a solution for the abstract task can be transformed in polynomial time in a solution for the original task.
FDR Reformulation

Free DTG: The domain transition graph of a variable only considering actions without preconditions or effects on other variables.

Variable Abstraction (Helmert, 2006)

Any variable whose free DTG is strongly connected can be abstracted away.
FDR Reformulation

**Free DTG:** The domain transition graph of a variable only considering actions without preconditions or effects on other variables.

**Variable Abstraction (Helmert, 2006)**

Any variable whose free DTG is strongly connected can be abstracted away

→ In our example: gets rid of variable $E$

Abstract plan $= \langle D^R_{A-B,2-1}, D^R_{B-D,1-0} \rangle$

Original plan $= \langle \rangle$
FDR Reformulation

**Free DTG**: The domain transition graph of a variable only considering actions without preconditions or effects on other variables.

**Variable Abstraction (Helmert, 2006)**
Any variable whose free DTG is strongly connected can be abstracted away

→ In our example: gets rid of variable $E$

Abstract plan $= \langle DR_{A-B,2-1}, DR_{B-D,1-0} \rangle$
Original plan $= \langle turnon, DR_{A-B,2-1}, DR_{B-D,1-0} \rangle$
FDR Reformulation

Free DTG: The domain transition graph of a variable only considering actions without preconditions or effects on other variables.

Variable Abstraction (Helmert, 2006)
Any variable whose free DTG is strongly connected can be abstracted away.

→ In our example: gets rid of variable $E$

Abstract plan = $\langle DR_{A-B,2-1}, DR_{B-D,1-0} \rangle$
Original plan = $\langle turnon, DR_{A-B,2-1}, DR_{B-D,1-0} \rangle$

→ In Logistics: solves the problem without doing any search
FDR Reformulation

Free DTG: The domain transition graph of a variable only considering actions without preconditions or effects on other variables.

Variable Abstraction (Helmert, 2006)

Any variable whose free DTG is strongly connected can be abstracted away

→ In our example: gets rid of variable $E$

Abstract plan  $= \langle DR_{A-B,2-1}, DR_{B-D,1-0} \rangle$

Original plan  $= \langle turnon, DR_{A-B,2-1}, DR_{B-D,1-0} \rangle$

→ In Logistics: solves the problem without doing any search

Extensions:

• Haslum (2007) gave a stronger criteria

• Tozicka et al. (2016) use this to merge values of a variable
Running Example with Accidental Complexity

- \( V = \{T,F,E\} \) with \( D_t = \{A,B,C,D\} \),
  \( D_F = \{0,1,2\} \), \( D_E = \{off, rd, on\} \).
- \( A = \{\text{drive}(x,x',f,f'), \text{turnon, checkfuel}\} \)
- \( I = \{T = A, F = 2, E = \text{off}\} \)
- \( G = \{T = D\} \)

Before turning on the engine, we need to check that we have 2 units of fuel with the check-fuel action.
Running Example with Accidental Complexity

- $V = \{T, F, E\}$ with $D_t = \{A, B, C, D\}$, $D_F = \{0, 1, 2\}$, $D_E = \{\text{off, rd, on}\}$.
- $A = \{\text{drive}(x, x', f, f'), \text{turnon}, \text{checkfuel}\}$
- $I = \{T = A, F = 2, E = \text{off}\}$
- $G = \{T = D\}$

→ Before turning on the engine, we need to check that we have 2 units of fuel with the check-fuel action

- check-fuel:  
  pre: $E = \text{off}, F = 2$  
  eff: $E = \text{rd}$

- turn-on:  
  pre: $E = \text{rd}$  
  eff: $E = \text{on}$
Atomic: One TS per variable such that $\Theta_1 \otimes \Theta_2 \otimes \Theta_3 = \Theta$
Atomic: One TS per variable such that $\Theta_1 \otimes \Theta_2 \otimes \Theta_3 = \Theta$
**Atomic**: One TS per variable such that $\Theta_1 \otimes \Theta_2 \otimes \Theta_3 = \Theta$
Atomic: One TS per variable such that $\Theta_1 \otimes \Theta_2 \otimes \Theta_3 = \Theta$
Atomic: One TS per variable such that $\Theta_1 \otimes \Theta_2 \otimes \Theta_3 = \Theta$
**Merge-and-Shrink**

**Atomic:** One TS per variable such that $\Theta_1 \otimes \Theta_2 \otimes \Theta_3 = \Theta$
M&S as Reformulation Framework

\[ \Pi^{FDR} \rightarrow \{\Theta_1, \ldots, \Theta_n\}, L \rightarrow \text{M&S} \rightarrow \{\Theta^\alpha\} \]

Search

\( h \)
M&S as Reformulation Framework

\[ \Pi^{FDR} \rightarrow \{\Theta_1, \ldots, \Theta_n\}, L \]

M&S

\[ \{\Theta'_1, \ldots, \Theta'_k\}, L' \rightarrow \text{Search} \]

h

\[ \{\Theta^\alpha\} \]
M&S as Reformulation Framework

\[ \Pi^{FDR} \rightarrow \{\Theta_1, \ldots, \Theta_n\}, L \]

\[ \text{M&S} \]

\[ \{\Theta'_1, \ldots, \Theta'_k\}, L' \rightarrow \text{Search} \]

\[ \Pi^{FDR} \rightarrow h^{FF} \]

\[ \text{delete relaxation} \]

\[ \text{Search} \]
M&S as Reformulation Framework

$$\Pi^{FDR} \rightarrow \{\Theta_1, \ldots, \Theta_n\}, L$$

M&S

$$\{\Theta'_1, \ldots, \Theta'_k\}, L'$$

Search

$$\Pi^{FDR} \rightarrow \{\Theta_1, \ldots, \Theta_n\}, L$$

M&S

delete relaxation

$$h^{FF}$$

$$h$$

$$h$$

$$\Pi^{FDR} \rightarrow \{\Theta_1, \ldots, \Theta_n\}, L$$

M&S

$$\{\Theta'_1, \ldots, \Theta'_k\}, L'$$

Search

$$\{\Theta^\alpha\}$$
FTS Representation and Successor Generation

\[ \Theta \]

\[ \Theta_T \]

\[ \Theta_F \]

\[ \Theta_E \]

\[ \text{Successor: } \langle T=A, F=2, E=\text{rd} \rangle \]

\[ \text{Successor (D}_{\text{R}}A-B, 2-1): \langle T=B, F=1, E=\text{on} \rangle \]

\[ \text{Successor (D}_{\text{R}}A-C, 2-1): \langle T=C, F=1, E=\text{on} \rangle \]

\[ \text{Successor (D}_{\text{R}}): \langle T=B, F=1, E=\text{on} \rangle \]

\[ \text{Successor (D}_{\text{R}}): \langle T=C, F=1, E=\text{on} \rangle \]

\[ \text{DR}_{\ast} \]

\[ \text{DR}_{\ast} \]
State: \( \langle T=A, F=2, E=\text{off} \rangle \)
FTS Representation and Successor Generation

State: \( \langle T=A, F=2, E=\text{off} \rangle \)

Applicable labels:
FTS Representation and Successor Generation

State: \( \langle T=A, F=2, E=\text{off} \rangle \)
Applicable labels: \( \{CF\} \)
FTS Representation and Successor Generation

State: \( \langle T=A, F=2, E=\text{off} \rangle \)
Applicable labels: \( \{ \text{CF} \} \)
Successor: \( \langle T=A, F=2, E=\text{rd} \rangle \)
FTS Representation and Successor Generation

State: \( \langle T=A, F=2, E=\text{on} \rangle \)
FTS Representation and Successor Generation

State: \( \langle T=A, F=2, E=on \rangle \)
Applicable labels: \( \{ DR_{A-B,2-1}, DR_{A-C,2-1} \} \)
FTS Representation and Successor Generation

State: \( \langle T=A, F=2, E=on \rangle \)
Applicable labels: \( \{ DR_{A-B,2-1}, DR_{A-C,2-1} \} \)
Successor (\( DR_{A-B,2-1} \)): \( \langle T=B, F=1, E=on \rangle \)
Successor (\( DR_{A-C,2-1} \)): \( \langle T=C, F=1, E=on \rangle \)
FTS Representation and Successor Generation

\[ \Theta \]

\[ T = A, F = 2, E = \text{rd} \]

\[ \Theta_T \]

\[ T = B, F = 1, E = \text{on} \]

\[ \Theta_F \]

\[ T = C, F = 1, E = \text{on} \]

\[ \Theta_E \]
State: $\langle T=A, F=2, E=on \rangle$
Applicable labels: $\{DR\}$
Successor (DR): $\langle T=B, F=1, E=on \rangle$
Successor (DR): $\langle T=C, F=1, E=on \rangle$

Alvaro Torralba, Silvan Sievers
Merge-and-Shrink Task Reformulation for Classical Planning
What’s the Difference Anyway?

Advantage of FTS over FDR:
- Limited form of disjunctive preconditions
- Limited form of conditional effects
- Limited form of non-deterministic effects
What’s the Difference Anyway?

Advantage of FTS over FDR:
- Limited form of disjunctive preconditions
- Limited form of conditional effects
- Limited form of non-deterministic effects

Drive:
\[
\begin{align*}
\text{pre: } & F=2 \lor F=1 \\
\text{eff: } & F=2 \implies F=1 \\
& F=1 \implies F=0 \\
& \text{at} = A \lor \text{at} = D \implies \text{at} = B \lor \text{at} = C \\
& \text{at} = B \lor \text{at} = C \implies \text{at} = A \lor \text{at} = D
\end{align*}
\]
What’s the Difference Anyway?

Advantage of FTS over FDR:
- Limited form of disjunctive preconditions
- Limited form of conditional effects
- Limited form of non-deterministic effects

\[ \text{pre: } F = 2 \lor F = 1 \quad \rightarrow \quad F = 1 \]
\[ \rightarrow \quad F = 0 \]
\[ \text{eff: } \]
\[ \text{at: } A \lor \text{at: } D \quad \rightarrow \quad \text{at: } B \lor \text{at: } C \]
\[ \rightarrow \quad \text{at: } A \lor \text{at: } D \]
What’s the Difference Anyway?

Advantage of FTS over FDR:
- Limited form of disjunctive preconditions
- Limited form of conditional effects
- Limited form of non-deterministic effects

![Image with colors]
What’s the Difference Anyway?

Advantage of FTS over FDR:
- Limited form of disjunctive preconditions
- Limited form of conditional effects
- Limited form of non-deterministic effects

Drive:
pre: F=2 ∨ F=1
eff: F=2 → F=1
F=1 → F=0
at=A ∨ at=D → at=B OR at=C
at=B ∨ at=C → at=A OR at=D

Transforming from FTS to FDR may cause:
an exponential blow-up in the number of actions, or
an increase in plan length
What’s the Difference Anyway?

Advantage of FTS over FDR:
- Limited form of disjunctive preconditions
- Limited form of conditional effects
- Limited form of non-deterministic effects
Shrink

Replaces a TS $\Theta_i$ by an abstraction thereof ($\alpha(\Theta_i)$).
The state space of the new task is an abstraction of the original.
Shrink

Replaces a TS $\Theta_i$ by an abstraction thereof ($\alpha(\Theta_i)$)

The state space of the new task is an abstraction of the original

\[ \text{Plan} = \langle \rangle \Rightarrow \]

Only "refinable" abstractions are suitable for reformulation
Replaces a TS $\Theta_i$ by an abstraction thereof ($\alpha(\Theta_i)$)
The state space of the new task is an abstraction of the original
Shrink

Replaces a TS $\Theta_i$ by an abstraction thereof $(\alpha(\Theta_i))$
The state space of the new task is an abstraction of the original

Plan = $\langle \rangle$
Shrink

Replaces a TS $\Theta_i$ by an abstraction thereof ($\alpha(\Theta_i)$)
The state space of the new task is an abstraction of the original

Plan $= \langle \rangle \Rightarrow$ Only “refinable” abstractions are suitable for reformulation
(Strong) Bisimulation Shrinking

Two states are equivalent if they have the same outgoing transitions
(Strong) Bisimulation Shrinking

Two states are equivalent if they have the same outgoing transitions
(Strong) Bisimulation Shrinking

Two states are equivalent if they have the same outgoing transitions

Plan = \langle CF, ON, DR(BC), DR(D) \rangle \rightarrow
(Strong) Bisimulation Shrinking

Two states are equivalent if they have the same outgoing transitions

Plan = \langle \text{CF}, \text{ON}, \text{DR}(BC), \text{DR}(D) \rangle \rightarrow \langle \text{CF}, \text{ON}, \text{DR}(B), \text{DR}(D) \rangle
Weak Bisimulation Shrinking

1. Identify $\tau$ labels that are internal to a TS (self-loop everywhere)

```
Plan = ⟨CF, DR(BC), DR(D)⟩ → ⟨CF, ON, DR(BC), DR(D)⟩
```
Weak Bisimulation Shrinking

1. Identify $\tau$ labels that are internal to a TS (self-loop everywhere)
2. Bisimulation allowing free-use of $\tau$ labels
Weak Bisimulation Shrinking

1. Identify $\tau$ labels that are internal to a TS (self-loop everywhere)
2. Bisimulation allowing free-use of $\tau$ labels

---

**Plan =** $\Theta_T \rightarrow \Theta_F \rightarrow \Theta_E$
Weak Bisimulation Shrinking

1. Identify $\tau$ labels that are internal to a TS (self-loop everywhere)
2. Bisimulation allowing free-use of $\tau$ labels

![Diagram of Weak Bisimulation Shrinking]

- $\Theta_T$ and $\Theta_F$ represent the transition systems.
- $\Theta_E$ represents the expanded system.
- $\Theta_T$ and $\Theta_F$ are bisimilar, but $\Theta_E$ is not.
Weak Bisimulation Shrinking

1. Identify $\tau$ labels that are internal to a TS (self-loop everywhere)
2. Bisimulation allowing free-use of $\tau$ labels

Plan = $\langle CF, DR(BC), DR(D) \rangle \to$
Weak Bisimulation Shrinking

1. Identify $\tau$ labels that are internal to a TS (self-loop everywhere)
2. Bisimulation allowing free-use of $\tau$ labels

\[
\begin{align*}
\Theta_T & \quad \Theta_F & \quad \Theta_E \\
D \xleftarrow{\text{DR}} & \quad \text{CF} & \quad 0 \xleftarrow{\text{DR}} & \quad \text{CF} & \quad \text{ro} \xleftarrow{\text{DR}} & \quad \text{off} \\
\text{BC} \xleftarrow{\text{DR}} & \quad \text{CF} & \quad 1 \xleftarrow{\text{DR}} & \quad \text{CF} & \quad \text{ef} \xleftarrow{\text{DR}} & \quad \text{on} \\
A \xleftarrow{\text{DR}} & \quad \text{CF} & \quad 2 \xleftarrow{\text{DR}} & \quad \text{CF} & \quad \text{off} \xleftarrow{\text{DR}} & \quad \text{on} \\
\end{align*}
\]

Plan = $\langle \text{CF, DR(BC), DR(D)} \rangle \rightarrow \langle \text{CF, ON, DR(BC), DR(D)} \rangle$
Replace $\Theta_i$ and $\Theta_j$ by their product: $\Theta_i \otimes \Theta_j$
Replace $\Theta_i$ and $\Theta_j$ by their product: $\Theta_i \otimes \Theta_j$
Replace $\Theta_i$ and $\Theta_j$ by their product: $\Theta_i \otimes \Theta_j$
Merge

Replace $\Theta_i$ and $\Theta_j$ by their product: $\Theta_i \otimes \Theta_j$

[Diagram showing the merge operation with nodes and edges labeled]
Merge

Replace $\Theta_i$ and $\Theta_j$ by their product: $\Theta_i \otimes \Theta_j$
Merge

Replace $\Theta_i$ and $\Theta_j$ by their product: $\Theta_i \otimes \Theta_j$
Relation to FDR Reformulation Methods

An FTS reformulation method dominates an FDR reformulation method if it can do the same reformulations:

\[ \Pi \xrightarrow{\rho^{FDR}} \rho^{FDR}(\Pi) \]

\[ \{\Theta_1, \ldots, \Theta_n\}, L \]

\[ \{\Theta'_1, \ldots, \Theta'_m\}, L' \]
Relation to FDR Reformulation Methods

An FTS reformulation method dominates an FDR reformulation method if it can do the same reformulations:

$$\Pi \xrightarrow{\rho^{FDR}} \rho^{FDR}(\Pi)$$

$$\{\Theta_1, \ldots, \Theta_n\}, L \xrightarrow{\rho^{FTS}} \{\Theta'_1, \ldots, \Theta'_m\}, L'$$

Variable abstraction and merge values are dominated by weak bisimulation shrinking (plus removing TSs with a core state)

Generalize actions is dominated by label reduction
Relation to FDR Reformulation Methods

An FTS reformulation method dominates an FDR reformulation method if it can do the same reformulations:

\[
\begin{align*}
\Pi & \xrightarrow{\rho^{FDR}} \rho^{FDR}(\Pi) \\
\{\Theta_1, \ldots, \Theta_n\}, L & \xrightarrow{\rho^{FTS}} \{\Theta'_1, \ldots, \Theta'_m\}, L'
\end{align*}
\]

- Variable abstraction and merge values are dominated by weak bisimulation shrinking (plus removing TSs with a core state)
- Generalize actions is dominated by label reduction
Search Space Reduction: Optimal

- **Bisimulation + LR**
- **Merge (DFP) + LR + Bisimulation**

![Graphs showing search space reduction](image-url)
Search Space Reduction: Satisficing

Weak Bisimulation + LR  Merge (DFP) + LR + Weak Bisimulation
## Optimal Planning

### Table

<table>
<thead>
<tr>
<th></th>
<th>FDR</th>
<th>a</th>
<th>a-ls</th>
<th>d-ls</th>
<th>m-ls</th>
<th>tot</th>
<th>orcl</th>
</tr>
</thead>
<tbody>
<tr>
<td>FDR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>a</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>770</td>
<td></td>
</tr>
<tr>
<td>a-ls</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>780</td>
<td></td>
</tr>
<tr>
<td>d-ls</td>
<td></td>
<td></td>
<td></td>
<td>600</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>m-ls</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>632</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FDR</td>
<td>797</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>a</td>
<td></td>
<td>770</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>a-ls</td>
<td></td>
<td>780</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>d-ls</td>
<td></td>
<td></td>
<td>600</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>m-ls</td>
<td></td>
<td></td>
<td></td>
<td>632</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FDR</td>
<td>822</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>a</td>
<td></td>
<td>826</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>a-ls</td>
<td></td>
<td>831</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>d-ls</td>
<td></td>
<td></td>
<td>815</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>m-ls</td>
<td></td>
<td></td>
<td></td>
<td>849</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

$\hat{h}$ Max: 801

$\hat{h}$, M&Sd: 910
## Optimal Planning

<table>
<thead>
<tr>
<th></th>
<th>FDR</th>
<th>a</th>
<th>a-Is</th>
<th>d-Is</th>
<th>m-Is</th>
<th>tot</th>
<th>orcl</th>
</tr>
</thead>
<tbody>
<tr>
<td>FDR</td>
<td>797</td>
<td>770</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>a</td>
<td>780</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>a-Is</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>d-Is</td>
<td>2</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>m-Is</td>
<td>4</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FDR</td>
<td>822</td>
<td>826</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>a</td>
<td>831</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>a-Is</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>d-Is</td>
<td>11</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>m-Is</td>
<td>15</td>
<td>15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\( h_{\text{max}} = 801 \)

\( h_{\text{M&Sd}} = 910 \)
## Optimal Planning

<table>
<thead>
<tr>
<th></th>
<th>FDR</th>
<th>a</th>
<th>a-ls</th>
<th>d-ls</th>
<th>m-ls</th>
<th>tot</th>
<th>orcl</th>
</tr>
</thead>
<tbody>
<tr>
<td>FDR</td>
<td>–</td>
<td>12</td>
<td>13</td>
<td>37</td>
<td>36</td>
<td>797</td>
<td></td>
</tr>
<tr>
<td>a</td>
<td>1</td>
<td>–</td>
<td>1</td>
<td>36</td>
<td>36</td>
<td>770</td>
<td></td>
</tr>
<tr>
<td>a-ls</td>
<td>3</td>
<td>4</td>
<td>–</td>
<td>36</td>
<td>35</td>
<td>780</td>
<td></td>
</tr>
<tr>
<td>d-ls</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>–</td>
<td>7</td>
<td>600</td>
<td></td>
</tr>
<tr>
<td>m-ls</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>19</td>
<td>–</td>
<td>632</td>
<td></td>
</tr>
</tbody>
</table>

- $h_{\text{max}}$: 801
- $h_{\text{M&Sd}}$: 910

$\text{Álvaro Torralba, Silvan Sievers}$

$\text{Merge-and-Shrink Task Reformulation for Classical Planning}$
## Satisficing Planning

<table>
<thead>
<tr>
<th></th>
<th>FDR</th>
<th>a</th>
<th>a-ls</th>
<th>d-ls</th>
<th>m-ls</th>
<th>tot</th>
<th>orcl</th>
</tr>
</thead>
<tbody>
<tr>
<td>FDR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1326</td>
<td></td>
</tr>
<tr>
<td>a</td>
<td></td>
<td></td>
<td>1272</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>a-ls</td>
<td></td>
<td></td>
<td>1368</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>d-ls</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1208</td>
<td></td>
</tr>
<tr>
<td>m-ls</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1224</td>
<td></td>
</tr>
</tbody>
</table>

\( h_{FF} : 1413 \)

<table>
<thead>
<tr>
<th></th>
<th>FDR</th>
<th>a</th>
<th>a-ls</th>
<th>d-ls</th>
<th>m-ls</th>
<th>tot</th>
<th>orcl</th>
</tr>
</thead>
<tbody>
<tr>
<td>FDR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1502</td>
<td></td>
</tr>
<tr>
<td>a</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1461</td>
<td></td>
</tr>
<tr>
<td>a-ls</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1471</td>
<td></td>
</tr>
<tr>
<td>d-ls</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1357</td>
<td></td>
</tr>
<tr>
<td>m-ls</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1322</td>
<td></td>
</tr>
</tbody>
</table>

\( h_{FF} \) p.: 1589
Satisficing Planning

<table>
<thead>
<tr>
<th>FDR</th>
<th>a</th>
<th>a-ls</th>
<th>d-ls</th>
<th>m-ls</th>
<th>tot</th>
<th>orcl</th>
</tr>
</thead>
<tbody>
<tr>
<td>FDR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1326</td>
</tr>
<tr>
<td>a</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1272</td>
</tr>
<tr>
<td>a-ls</td>
<td>15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1368</td>
</tr>
<tr>
<td>d-ls</td>
<td>10</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td>1208</td>
</tr>
<tr>
<td>m-ls</td>
<td>15</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td>1224</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>FDR</th>
<th>a</th>
<th>a-ls</th>
<th>d-ls</th>
<th>m-ls</th>
<th>tot</th>
<th>orcl</th>
</tr>
</thead>
<tbody>
<tr>
<td>FDR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1502</td>
</tr>
<tr>
<td>a</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1461</td>
</tr>
<tr>
<td>a-ls</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1471</td>
</tr>
<tr>
<td>d-ls</td>
<td>6</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td>1357</td>
</tr>
<tr>
<td>m-ls</td>
<td>7</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td>1322</td>
</tr>
</tbody>
</table>
## Satisficing Planning

<table>
<thead>
<tr>
<th></th>
<th>FDR</th>
<th>a</th>
<th>a-ls</th>
<th>d-ls</th>
<th>m-ls</th>
<th>tot</th>
<th>orcl</th>
</tr>
</thead>
<tbody>
<tr>
<td>FDR</td>
<td>–</td>
<td>18</td>
<td>15</td>
<td>27</td>
<td>22</td>
<td>1326</td>
<td></td>
</tr>
<tr>
<td>a</td>
<td>6</td>
<td>–</td>
<td>13</td>
<td>28</td>
<td>22</td>
<td>1272</td>
<td></td>
</tr>
<tr>
<td>a-ls</td>
<td>18</td>
<td>15</td>
<td>–</td>
<td>31</td>
<td>24</td>
<td>1368</td>
<td></td>
</tr>
<tr>
<td>d-ls</td>
<td>10</td>
<td>10</td>
<td>4</td>
<td>–</td>
<td>11</td>
<td>1208</td>
<td></td>
</tr>
<tr>
<td>m-ls</td>
<td>13</td>
<td>15</td>
<td>7</td>
<td>21</td>
<td>–</td>
<td>1224</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>FDR</th>
<th>a</th>
<th>a-ls</th>
<th>d-ls</th>
<th>m-ls</th>
<th>tot</th>
<th>orcl</th>
</tr>
</thead>
<tbody>
<tr>
<td>FDR</td>
<td>–</td>
<td>17</td>
<td>15</td>
<td>24</td>
<td>23</td>
<td>1502</td>
<td></td>
</tr>
<tr>
<td>a</td>
<td>8</td>
<td>–</td>
<td>11</td>
<td>25</td>
<td>24</td>
<td>1461</td>
<td></td>
</tr>
<tr>
<td>a-ls</td>
<td>13</td>
<td>8</td>
<td>–</td>
<td>26</td>
<td>26</td>
<td>1471</td>
<td></td>
</tr>
<tr>
<td>d-ls</td>
<td>9</td>
<td>6</td>
<td>2</td>
<td>–</td>
<td>15</td>
<td>1357</td>
<td></td>
</tr>
<tr>
<td>m-ls</td>
<td>9</td>
<td>7</td>
<td>3</td>
<td>16</td>
<td>–</td>
<td>1322</td>
<td></td>
</tr>
</tbody>
</table>
Conclusion

- Task reformulation is an important tool to solve planning tasks
- Merge-and-Shrink is a powerful reformulation framework → dominates similar methods in FDR
Conclusion

- Task reformulation is an important tool to solve planning tasks
- Merge-and-Shrink is a powerful reformulation framework
  → dominates similar methods in FDR
- Adapt search algorithms and heuristics for FTS
  - Successor Generation
  - Delete-relaxation heuristics ($h^{FF}$)
  → More abstraction heuristics for cost-optimal planning
  → Landmarks and Novelty for satisficing planning
Conclusion

- Task reformulation is an important tool to solve planning tasks
- Merge-and-Shrink is a powerful reformulation framework
  → dominates similar methods in FDR
- Adapt search algorithms and heuristics for FTS
  - Successor Generation
  - Delete-relaxation heuristics ($h^{FF}$)
  → More abstraction heuristics for cost-optimal planning
  → Landmarks and Novelty for satisficing planning
- Provide a plan reconstruction for M&S transformations
  - Merge, LR, Pruning, Bisimulation → optimal reformulation
  - Weak bisimulation → satisficing reformulation
  → Dominance-based pruning
  → Tunnel macros