Extending SymbolicPlanners with forward propagation landmark extraction

Ka Fui Yang

TU Delft

Background

- **Classical planning problem**
  - Find a sequence of actions to go from the initial state to the goal state, where the states are defined by predicates
  - Actions changes state by adding or deleting predicates
- **Landmarks**
  - States necessary to achieve goal state in every successful plan
- **Relaxed planning graph**
  - Planning graph is a directed, leveled graph where the levels alternate between action and predicate node
  - RPG is a planning graph with delete predicates omitted
  - Mutex exclusion not necessary but verification of landmark is
- **Grounding using PDDL.jl**
  - Predicates and action schemas instantiated using objects from problem instance and domain file

Problem description

- **"What design choices can be made to adapt the forward propagation extraction algorithm into SymbolicPlanners?"**
  - Bases on the algorithm described by Zhu & Givan[1]
  - Using labels to forward propagate predicates and extracting landmarks using the labels of goal state predicates
- **SymbolicPlanners**
  - Written in Julia and missing landmark extraction methods
- **Fast Downward**
  - Written in C++ and has forward propagation extraction method

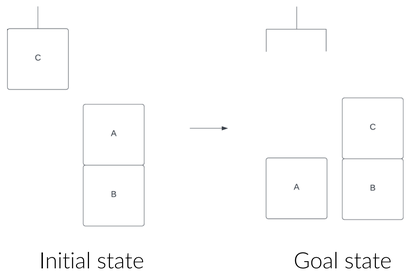

Figure 1. A problem instance in Blocksworld domain where A, B and C are objects.

Methodology

- From SymbolicPlanners use the build RPG method
  - **act_parents:** Contains the effect of action
  - **act_children:** Contains preconditions of action
  - **cond_children:** Contains actions where precondition is equal to condition
- Design choices for representing the layers with labels using the RPG
- Propagate labels using combination of union and intersection
- Using queue to add actions that needs to be applied
- Using map to get actions from newly added conditions in new layer
- Create Landmark graph using last layer and goal state.

Algorithm 1 Forward propagation

```
while queue not empty do
    create new queue
    copy old layer
    for action in queue do
        if action is applicable then
            label the effect of action in layer
            if Check for difference old and new layer then
                Add action of difference in new queue
            end
        end
    end
    queue = new queue
end
```

- **Performance**
  - Equal amount of LM extracted using SymbolicPlanners and Fast Downward implementation
  - Runtime between forward propagation, backward search and Fast Downward

Result

- Implementation does not extract the same number of LM as Fast Downward in complex domains such as Freecell and Grid.
- Implementation extracts the correct number LM for simple domains such as Blocksworld and Logistics.

Domain	Problem instance with same amount LM extracted
Blocksworld	102/102
Logistics	82/82
Tireworld	16/30
Grid	0/5
Freecell	0/60
Gripper	20/20

Table 1. Problem instances with correct amount of LM extracted per domain.

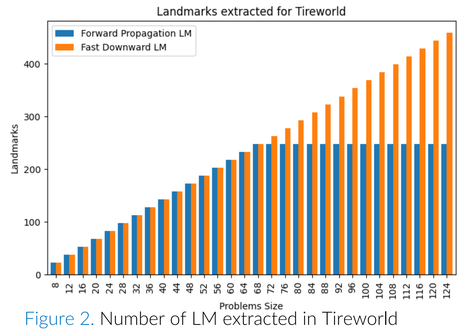

Figure 2. Number of LM extracted in Tireworld domain.

- Implementation stops extracting new LM at problem sizes bigger than 68 in Tireworld domain
- Method is 2 times slower compared to Fast Downward while running forward propagation but not extracting new LM

- Grounding complex domains is expensive and SymbolicPlanners RPG method has a limitation.
- Forward propagation with verification is significantly slower compared to without verification while removing no LM.

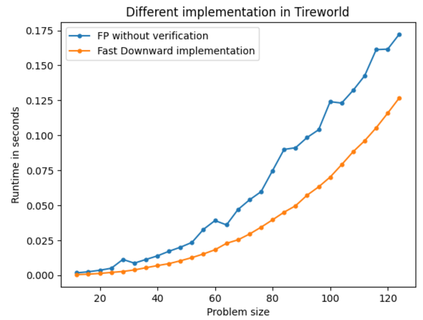

Figure 3. Runtime of SymbolicPlanners and Fast Downward implementation on Tireworld domain.

Conclusion

- Incorrect amount of LM extracted in complex domains but correct amount of LM extracted in simple domains
- PDDL.jl that SymbolicPlanners uses for RPG is not grounding complex domains properly
- Runtime for the implementation without verification is only 2 times slower compared to Fast Downward implementation

References

[1] Lin Zhu and Robert Givan. Landmark extraction via planning graph propagation. ICAPS Doctoral Consortium, pages 156–160, 2003

Acknowledgement

Firstly, Paul Tervoort for providing me with his implementation of landmark extraction and verification. Secondly, the rest of the research group consisting of Pauline Hengst, Noah Tjoen and Bar van Maris for constant discussion and insight on the shared topic. Thirdly, our supervisor Issa Hanou and responsible professor Sebastijan Dumančić for providing feedback and answering questions weekly.

Author: Ka Fui Yang Contact: K.F.Yang@student.tudelft.nl Supervisor: Issa K. Hanou Responsible professor: Sebastijan Dumančić