Reproducing the concept of ordered landmarks in planning

The effect of ordered landmarks on plan length in forward search

Paul Tervoort[1]

Supervisor(s): Sebastijan Dumančić[1], Issa Hanou[1]

TU Delft

Contact

E-mail: *P.B.Tervoort@student.tudelft.nl*

Introduction

Planning problem

- Generate an action sequence from the initial state to the goal state
- Minimize the sequence length
- The problem is PSPACE-hard

Forward search

- A type of planning algorithm
- Models the planning problem as a graph, then uses A* to find a plan
- The difference between planners of this type is the heuristic used in A*

Landmark

Proposition in the state space that is true at some point in every valid solution

Landmark order

For landmarks A and B:

*A ≤ B holds **iff** when B ∧ ¬A holds, ¬B must hold at some point before A ∧ B can hold*

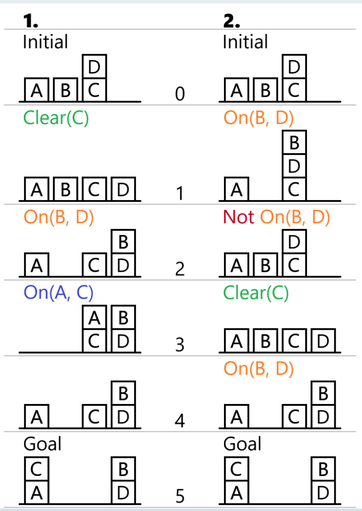


Figure 1: The states in two valid plans for the problem in *Figure 2*. Proposition *On(A, C)* is not a landmark. Landmark *Clear(C)* is reasonably ordered before landmark *On(B, D)*.

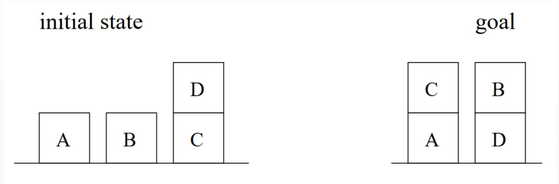


Figure 2: Example problem from *Blocksworld*
Source: [1, Fig. 1]

Research Question

How do ordered landmarks affect the solution length of forward search planning in the SymbolicPlanners framework?

1. Reproduce previous work using SymbolicPlanners
2. Draw conclusion about solution length using landmarks

Methodology

Reproduce

- The results in previous work from Porteous et al. [1] are reproduced
- Possible biases are reduced by implementing all planners in the SymbolicPlanners framework

Implement

- The idea of ordered landmarks is implemented
- The Julia language is used for the implementation

Benchmark

- Find PDDL descriptions of the problem instances used for previous experiments
  PDDL: a format to describe planning problems
- We compare to the forward search planner *Fast-Forward*
- Both our planner using ordered landmarks and a *Fast-Forward* planner from SymbolicPlanners was run on the obtained instances

Compare

- The ratio between the solution length of *Fast-Forward* and the landmark implementation is calculated
- The calculated ratios are compared against the ratios in previous work

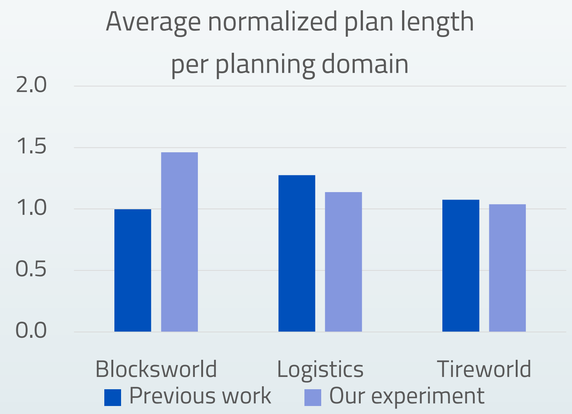
Results

Compared to previous work

- Not exact the same solution lengths
- Mean and standard deviation comparable

Relative solution lengths in our experiment

- With landmarks 22% longer plans than without
- Notable difference in *Blocksworld*




```
foreach f ∈ I do
    enqueue(Q, f);
end
while ¬empty(Q) do
    n ← dequeue(Q);
    foreach p ∈ prerequisites_in_graph(n, P) do
        if ¬reach_without_prop(p, G) then
            enqueue(Q, p);
            insert(N, p);
            insert(E, (p, n));
        end
    end
    foreach n1 ∈ N do
        if true_in_state(n1, G) then
            foreach n2 ∈ N do
                if interferes(n1, n2) then
                    insert(E, (n1, n2));
                end
            else
                foreach n2 ∈ interesting_nodes(n1) do
                    if interferes(n1, n2) then
                        insert(E, (n1, n2));
                    end
end
```

Figure 3: Ordered landmark generation

Limitations

Implementation

- SymbolicPlanners is slower than implementations in previous work
- Design choices in previous work are not well documented

Experiment

- Only one previous experiment verified
- Less than half of the tested problems finished
- Some tested domains are left out due to a lack of results

Conclusion

A planner which uses landmarks generates on average 22% longer plans than a planner which does not use landmarks

Acknowledgements and References

Poster template:
https://cdn.posternerd.com/download/generic-scientific-poster-templates/conceptualizingcobalt_36x24.pptx

[1] J. Porteous, L. Sebastia, and J. Hoffmann. *On the extraction, ordering, and usage of landmarks in planning*. In ECP-01. Sixth European Conference on Planning, Toledo, Spain, pages 37–48, 2001.

[1]EEMCS, Delft University of Technology, The Netherlands