Accelerating Partial-Order Planners: Some Techniques for Effective Search Control and Pruning

WRITTEN BY: ALFONSO GEREVINI AND LENHART SCHUBERT

PUBLISHED: JOURNAL OF ARTIFICIAL INTELLIGENCE RESEARCH



Partial-Order Planners

• POP – sound and complete partial planner

• **UCPOP** – extends POP by making it more powerful (conditional effects and universal quantifiers)

• What's the next step?

UCPOP

- Where do we make choices within the algorithm?
- Do these choices matter?
- Better choices can speed up the algorithm.

- Algorithm UCPOP $(P = \prec S, B, O, L \succ, G, \Lambda)$
- 1. **Termination:** If G is empty, report success and return P.
- Goal selection: Choose a goal ≺c, S≻∈ G. If a link S_i^{e,→c}S exists in L, fail (an impossible plan). Note that c is universally ground.
- Operator selection: Nondeterministically choose any existing (from S) or new (instantiated from Λ) step S_s with effect e and a universally ground clause p ∈ Υ(θ_ε) where MGU(c, p) ≠⊥. If no such choice exists then fail. Otherwise, let L' = L ∪ {S_s^{e₁∈}S}, B' = B ∪ MGU(c, p) ∪ β_ε ∪ β_{S_s}, O' = O ∪ {S_s < S}, G' = G − ≺c, S ≻, and let S' = S ∪ {S_s}.
- 4. Subgoal generation: If effect e has not already been used to establish a link in \mathcal{L} with bindings MGU(c, p) then let G' = G and for each $\sigma \in \Upsilon(\rho_e \setminus MGU(c, p))$ add $\prec \sigma, S_s \succ$ to G'. If $S_s \notin S$, for each $\sigma \in \Upsilon(\rho_S \setminus MGU(c, p))$ also add $\prec \sigma, S_s \succ$ to G'.
- 5. Causal link protection: Let l be a causal link $S_i^{e_{i,q}}S_j$ in \mathcal{L} . Let S_k be any step with an effect e_k and postcondition $p \in \theta_{e_k}$. Step S_k THREAT-ENS link l with clause $v_p \in \Upsilon(p)$ if possibly $S_i < S_k < S_j$ when $MGU(\neg q, v_p) \neq \bot$ is consistent with \mathcal{B} . For all such S_k , e_k , l and v_p such that S_k threatens l with v_p , nondeterministically do one of the following (or, if no choice exists, fail):
 - (a) **Promotion** If possibly $S_j < S_k$, let $\mathcal{O}' = \mathcal{O} \cup \{S_j < S_k\}.$
 - (b) **Demotion** If possibly $S_k < S_i$, let $\mathcal{O}' = \mathcal{O} \cup \{S_k < S_i\}$.
 - (c) Separation Let $\mathcal{O}' = \mathcal{O} \cup \{S_i < S_k < S_j\}$ then nondeterministically
 - i. Choose constraints β' on existentially quantified variables such that $MGU(\neg q, v_p)$ = \bot and let $\beta' = \beta' \cup \beta'$, or
 - ii. Choose a precondition $r \in \Upsilon(\rho_{E_k} \setminus MGU(\neg q, v_p))$ and let $G' = G' \cup \{\prec \neg r, S_k \succ \}.$
- Recursive invocation: If B' is inconsistent then fail; else call UCPOP(≺S', B', O', L'≻, G', Λ).

Plan Selection Strategy

• All partial plans must be fully refined at some point.

• Objective is to have a smart way to estimate which plans it would be best to refine.

2. Goal selection: Choose a goal $\prec c, S \succ \in G$. If a link $S_i \stackrel{e, \neg c}{\rightarrow} S$ exists in \mathcal{L} , fail (an impossible plan). Note that c is universally ground.

A* - Greedy Best First Search

 A* gives a good estimate on the "best" partial plan to refine.

•f(p) = overall solution cost

 $\bullet g(p) = cost of current partial plan$

 h(p) = estimate of the additional cost of the best complete solution that extends partial plan p

Partial Plan Costs

•Cost function g(p):

•S – the number of steps within a partial plan

•CL – the number of causal links within a partial plan

•Heuristic function h(p):

- •OC the number of open conditions (unsatisfied goals and preconditions)
- •UC the number of unsafe conditions (pairs that threaten causal link)

Heuristic Validity

• A* doesn't guarantee an optimal solution

•If h(p) overestimates the remaining solution, then it risks not guaranteeing an optimal solution

Heuristic Validity

•Cost Function Terms:

•S (step) is safe, less steps are generally good

•CL (causal link)

•Does a larger CL indicate a more complex plan? Is this bad?

 Large CL = more subgoals achieved by reuse of actions

Cargo Problem Example

At(P1, LEX) \land At(P2,LEX) \land At(C1, LEX)

At(P1, LEX) \land At(P2,LEX) \land At(C1, LEX)

At(P2,CVG) \land At(C1, CVG)

At(P2,CVG) \land At(C1, CVG)

Cargo Problem Example

At(P1, LEX) \land At(P2,LEX) \land At(C1, LEX)

At(P1, LEX) \land At(P2,LEX) \land At(C1, LEX)





At(P1, LEX) \land At(P2, LEX) \land At(C1, LEX)

At(P1, LEX) \land At(P2,LEX) \land At(C1, LEX)



Heuristic Validity

•Heuristic Function Terms:

•OC – an open condition typically indicates the need for another step.

•However, it's possible to utilize existing steps to satisfy an open condition.

•UC – does more unsafe conditions imply the partial plan shouldn't be refined?

•The final value of UC should be 0 for a complete plan.

Final Heuristic

• f(p) = S + OC

•OC, despite shortcomings, is the best option for h(p).

•UC, is too volatile and also overestimates

• Paper suggests UC could still be used, but at an augmented level.

•Alternative f(p) = S+OC+0.1UC

Plan-selection	CPU sec	Plans
S+OC+UC	204.51	160,911/107,649
S+OC	0.97	751/511

Goal Selection Strategy

•How do we choose what open condition to prioritize after selecting a partial plan?

•Least Commitment Flaw Repair (LCFR)

•Least Commitment strategy selects an open condition that generates least number of refined plans.

•Search Reductions + High Overhead

Goal Selection Strategy

- Certain refinements to the partial plan can be seen as deterministic
- Case 1: Open conditions that cannot be achieved
- "Doomed plans"
- Case 2: Open conditions that can only be achieved in one way
- There's no guesswork, it's our only option so we can view it as "zero commitment"

Cargo Problem Example



Cargo Problem Example



Zero Commitment and LIFO

• How to choose a goal once the deterministic options are gone?

• Paper suggests LIFO because it prioritizes one goal at a time.



Algorithm (ZLIFO)

- 1. A definite threat, utilize LIFO to pick among them.
- 2. An open condition that cannot be established in any way.
- 3. An open condition that can be resolved in only one way, with preference towards adding a new step rather than establishing through the *start* step.
- 4. An open condition, utilize LIFO to pick among them.

Experimental Results



Goal-selection	Plan-selection	CPU sec	Plans
LIFO	S+OC+UC	204.51	160,911/107,649
LIFO	S+OC	0.97	751/511
ZLIFO	S+OC+UC	6.90	1816/1291
ZLIFO	S+OC	0.54	253/184

Table I: Performance of plan/goal selection strategies on T-of-H1

Goal-selection	Plan-selection	CPU sec	Plans
LIFO	S+OC+UC	> 600	> 500,000
LIFO	S+OC	8.54	5506/3415
ZLIFO	S+OC+UC	> 600	> 500,000
ZLIFO	S+OC	1.24	641/420

Table II: Performance of plan/goal selection strategies on T-of-H3

Further Work: Domain Parameters

•Paper also discusses precomputing parameter domains.

- •Basic idea is to precompute constraints on the search space to prune the search during planning.
- •This algorithm can increase the number of zero commitment choices which can cause further speedups.

Questions?