

A Survey on Plan Optimization

Pascal Bercher¹, Patrik Haslum¹, and Christian Muise²



The Australian National University and Queen's University firstName.lastName@anu.edu.au and christian.muise@queensu.ca

Why to optimize plans?

- Finding *any plan* is faster than finding an optimal one
- Increase the flexibility of a plan
- One might 'receive' a plan (but can't generate one).

What to achieve? (Metrics)

- Number of actions
- Action costs
- Makespan (execution time under parallelism)
- Number of ordering constraints (proxy for the next)

What is a plan?

- An action sequence! Cf. (3) and (4), but without links
- Partially Ordered (=PO) plan. Cf. (1), but without links
- Block-decomposed PO (=BDPO) plans. Cf. (2)
- Partial-Order Causal Link (POCL) plans. Cf. (1), (3), (4)

• Number of linearizations (flexibility!)

What to optimize?

- Delete orderings \rightarrow Deordering
- Exchange orderings. \rightarrow Reordering
- Delete actions \rightarrow standard plan optimization
- Exchange actions \rightarrow less usual, but possible

Survey content

• Problem settings:

Classical and hierarchical planning



- Only deterministic plans are considered
- No additional inputs (like search space)
- Different approaches to optimize plans:
 - To optimize plan length / cost
 - To optimize orderings / linearizations
- Overview of complexity investigations
 - To optimize plan length
 - To optimize orderings / linearizations





Short (and incomplete) sneak-peek

What do our examples show?

- Optimization techniques:
 - Some just remove actions
 - Some replace actions
 - Some replace action parameters
- Several different approaches are deployed
 - Based on SAT encodings
 - Based on a planning encoding
 - Based on search (in the plan neighborhood)
 - Specialized algorithms
- Complexity results:
 - Only a few problems are in P
 - Most are NP-complete, both for plan length and orderings
 - Many investigations base on POCL plans

- They are all in the Blocks World domain
 - (1) uses two grippers (and hence allows parallelism)
 - (2) to (4) use just one gripper (thus, no parallelism)
- Different kinds of plans:
 - (1) a POCL plan: any linearization of the steps works
 - (2) a BDPO plan; any linearization of the blocks works
 - (3) and (4) action sequences (but also showing the causal link structure; may be ignored)
- (3) and (4) are reorderings of each other, but:
 - (3) is perfectly justified: no subsequence works!
 - (4) can be optimized, the red block can be removed