On the Computational Complexity of Stackelberg Planning and Meta-Operator Verification

Primary Keywords: *(4) Theory*

Explanation of this Supplement

The submitted main version of the paper is a short paper. This short-paper-version does not contain any proofs. If the paper is accepted, we will publish the proofs for all theorems as a technical report, e.g., on arXiv.

Alternatively, we have prepared a long-paper version of the paper, i.e., a version of the paper that contains all the proofs. This long-paper version adheres to the 8-page page limit of ICAPS.

If you, the reviewers judge our paper to be of acceptable quality for ICAPS, we would leave the decision to you whether this paper should be accepted as a short or long paper.

The long-paper version of the paper uses the exact same wording as the short-paper version, but differs in exactly two places from it:
(1) we added a section discussing the delete-free case of Stackel-berg planning and why this is not a sensible relaxation for obtaining heuristics.
(2) there are two additional sentences in the related work section explaining conditional and conformant planning.
Otherwise the papers are word-by-word identical.

This supplement contains

- First, the proposed technical report containing only the proofs.
- Second, the long-paper version of our paper that contains both the main text and all proofs.

On the Computational Complexity of Stackelberg Planning and Meta-Operator Verification

Primary Keywords: *(4) Theory*

Proposition 1. *STACKELSAT is polynomially reducible to STACKELMIN.*

Proof. Let $\Pi^{LF} = \langle V, A^L, A^F, I, G^F \rangle$ be a Stackelberg task. Then STACKELSAT is true iff STACKELMIN is true, setting $B^L = 2^{|V|} \cdot \max_{a^L \in A^L} c(a^L)$ and $B^F = 2^{|V|} \cdot \max_{a^F \in A^L} c(a^F)$. Clearly, both bounds can be computed in time linear in the size of Π^{LF}. □

Theorem 1. *STACKELSAT is PSPACE-complete.*

Proof. Membership: By Savitch's theorem (Savitch 1970), we only have to prove membership in NPSPACE. We can non-deterministically guess a leader plan and compute the resulting state s^L. We then have to check that the follower's task $\langle V, A^F, s^L, G^F \rangle$ is unsolvable. For this, we use the same idea as in the Immerman–Szelepcsényi theorem (Szelepcsényi 1987; Immerman 1988): We know that classical plan existence is NPSPACE-complete. By Savitch's theorem, there then is a deterministic poly-space algorithm that determines classical plan existence. We can apply this deterministic algorithm to determine the solvability of $\langle V, A^F, s^L, G^F \rangle$. If it is not, we return true, otherwise false.

Hardness: We reduce from PLANSAT. Given a classical planning problem $\Pi = (V, A, I, G)$, we create the Stackelberg planning problem $\Pi^{LF} = (V, \emptyset, A, I, G)$, i.e., we treat all actions as follower actions. Apply a deterministic algorithm to solve STACKELSAT for Π^{LF}. If the answer was true, return false, otherwise return true. Since the leader cannot perform any action, STACKELSAT is true iff the original planning task was unsolvable. □

Theorem 2. *STACKELMIN is PSPACE-complete.*

Proof. Membership: Determining whether a given classical planning problem has a plan of cost at most c is PSPACE complete (Bylander 1994). As such there is a deterministic poly-space Turing machine that determines whether there is a plan of cost at most c for a given planning problem. To decide the base version of Stackelberg planning, we can now perform the following algorithm: (1) From the state I, non-deterministiaclly guess an applicable sequence of actions with cost at most c_L and compute the resulting state s^L. (2) Apply the deterministic algorithm to determine whether there is a plan of cost at most c_F in the classical planning problem $\Pi = (V, A^F, s^L, G)$. If not, return yes, otherwise no. This algorithm solves the decision variant of Stackelberg planning. Step (1) can be performed in polynomial space, as the sequence can never plausibly be longer than exponential.

Hardness: We reduce from the plan existence problem for classical planning. Given a classical planning problem $\Pi = (V, A, I, G)$, we create the Stackelberg planning problem $\Pi = (V, \emptyset, A, I, G)$, i.e., we treat all actions as follower actions. We set $c_L = 0$ and $c_F = 1 + 2^{|V|} \cdot \max_{a \in A} c(a)$. Since the leader cannot perform any action, if it is possible to force the follower cost above c_F, then the original planning problem was unsolvable. □

Theorem 3. *STACKELPOLY is Σ_2^P-complete.*

Proof. Membership: Membership in Σ_2^P can be shown by providing an alternating Turing Machine, which switches only once from existential to universal nodes during each run. Using existential nodes, we guess a leader plan π^L with cost of at most c^L, execute it (if possible), to reach a state $s^L = I[\![\pi^L]\!]$. As argued above, $|\pi^L|$ is polynomially bounded, so s^L can be computed in polynomial time. Once s^L is computed, we switch to universal nodes and then guess a follower plan π^F of cost at most c^F which is again at most polynomially long. We then determine whether π^F is applicable in s^L and whether $s^L[\![\pi^F]\!] \subseteq G$. If so we return false, otherwise true.

Hardness: We reduce from the corresponding restricted QBF problem – which is to determine whether formulae of the form $\exists x_i \forall y_j \phi$ are satisfiable. W.l.o.g. we can assume that ϕ is in DNF.[1] Let ψ_i be the ith cube of ϕ. We construct a Stackelberg task $\Pi^{LF} = \langle V, A^L, A^F, I, G^F \rangle$, in which the leader selects the x_i variable assignment, and the follower tries to find a y_j assignment making ϕ evaluate to false:

$$V = \{T_i^x, F_i^x, S_i^x \mid x_i\} \cup \{T_j^y, F_j^y, S_j^y \mid y_j\} \cup \{c_i \mid \psi_i \in \phi\}$$

The initial state is $I = \{\}$. The leader actions consists of:

- $sel_i^x\text{-}T$ with $pre(sel_i^x\text{-}T) = \{\neg S_i^x\}$ and $add(sel_i^x\text{-}T) = \{S_i^x, T_i^x\}$
- $sel_i^x\text{-}F$ with $pre(sel_i^x\text{-}F) = \{\neg S_i^x\}$ and $add(sel_i^x\text{-}F) = \{S_i^x, F_i^x\}$

[1]Satisfiability of $\exists x_i \forall y_j \phi$ is trivial if ϕ is in CNFs as tautology is trivial for CNFs.

The follower has the following actions

- sel^y_j-T with $pre(sel^y_j\text{-}T) = \{\neg S^y_j\}$ and $add(sel^y_j\text{-}T) = \{S^y_j, T^y_j\}$
- sel^y_j-F with $pre(sel^y_j\text{-}F) = \{\neg S^y_j\}$ and $add(sel^y_j\text{-}F) = \{S^y_j, F^y_j\}$
- $val^j_{c_i}$ with $add(val^j_{c_i}) = \{c_i\}$, where l_j is the j-th literal in the i-th cube.
  - If it is positive literal then $pre(val^j_{c_i}) = \{F^l_j\}$
  - If it is a negative literal, then $pre(val^j_{c_i}) = \{T^l_j\}$
- $valS^j_{c_i}$ with $add(valS^j_{c_i}) = \{c_i\}$ and $pre(valS^j_{c_i}) = \{\neg S^x_k\}$, where l_j is the j-th literal in the i-th cube, and $l_j \in \{x_k, \neg x_k\}$ for some k.

All actions have cost 1. We set the goal to $G = \{c_i \mid \text{for every cube } i \text{ in } \phi\}$. We lastly set $B^L = |\{x_i | i\}|$ and $B^F = |\{y_j \mid j\}| + \#cubes + 1$.

The leader chooses the x_i assignment by executing either sel^x_i-T or sel^x_i-F for every x_i variable. After that, the follower can select truth values of the y_j variables using the sel^y_j-T and sel^y_j-F actions, in attempts to make one of the val_{c_i} actions for every cube c_i applicable. If this is possible, the respective cubes must be violated. If all cubes evaluate to false, then so does the overall formula ϕ. The additional $valS^j_{c_i}$ actions are necessary to forces the leader to choose an assignment to all x_i variables. Otherwise, unassigned x_i variables could make it impossible for the follower to find violations to all cubes. The value of B^L allows the leader to choose an assignment for all x_i variables. If the follower can reach her goal, she obviously has a plan with cost less than B^F. If there is a leader plan π^L where $c^F(\pi^L) \geq B^F$, then the formula $\exists x_i \forall y_j \phi$ is satisfiable. □

Theorem 5. *STACKELSAT$^1_{+1}$ is Σ^P_2-complete.*

Proof. Membership: As there are no delete effects, no action ever needs to be applied more than once. Hence, if a leader plan satisfying STACKELSAT$^1_{+1}$ exists, then there exists one whose size is polynomially bounded. The same also holds for the follower. To decide STACKELSAT$^1_{+1}$, we can thus use a similar approach as in Theorem 3.

Hardness: We show hardness again via a reduction from the satisfiability of restricted QBF of the form $\exists x_i \forall y_j \phi$, assuming ϕ to be in DNF. Similar to the proof of Theorem 3, the idea of our construction is to let the leader choose an assignment to x_i, which the follower needs to counter by finding an assignment to y_j that makes ϕ false.

The Stackelberg problem is defined as follows: The state variables are $V = \{T^x_i, T^y_j, C_k\}_{i,j,k}$ for appropriately ranging i, j, k. The initial state is $I = \emptyset$. The follower's goal is $G^F = \{C_k | \text{for each cube } k \text{ in } \phi\}$. The leader can choose the truth value for each x_i: via either sel^x_i-T with $pre(sel^x_i\text{-}T) = \{\neg F^x_i\}$ and $add(sel^x_i\text{-}T) = \{T^x_i\}$ or sel^x_i-F with $pre(sel^x_i\text{-}F) = \{\neg T^x_i\}$ and $add(sel^x_i\text{-}F) = \{F^x_i\}$. The follower can choose the truth value for each y_j via either sel^y_j-T with $pre(sel^y_j\text{-}T) = \{\neg F^y_j\}$ and $add(sel^y_j\text{-}T) = \{T^y_j\}$ or sel^y_j-F with $pre(sel^y_j\text{-}F) = \{\neg T^y_j\}$ and $add(sel^y_j\text{-}F) = \{F^y_j\}$, and she can make false each cube c_k in ϕ via each literal $l_i \in c_k$ by $val^i_{c_k}$ where $add(val^i_{c_k}) = \{c_k\}$ and if l_i is positive, then $pre(val^i_{c_k}) = \{\neg T^l_i\}$, else if if l_i is negative, then $pre(val^i_{c_k}) = \{\neg F^l_i\}$. This task obviously satisfies the STACKELSAT$^1_{+1}$ planning task restrictions. Moreover, note that $\exists x_i \forall y_j \phi$ is satisfiable iff the answer to STACKELSAT$^1_{+1}$ is yes. □

Theorem 6. *STACKELSAT$^+_1$ is NP-complete.*

Proof. Membership: Due to the restrictions, no action needs to be executed more than once. Hence, as before, the consideration of polynomially length-bounded plans suffices for answering Stackelberg plan existence for this class of tasks. To solve STACKELSAT$^+_1$, non-deterministically choose a (polynomially bounded) leader plan π^L and construct the corresponding follower task $\Pi^F(\pi^L)$. This can be done in polynomial time. PLANSAT for $\Pi^F(\pi^L)$ can be answered in (deterministic) polynomial time (Bylander 1994). Return true if the follower task is unsolvable, otherwise return false.

Hardness: By reduction from Boolean satisfiability. Let ϕ be a CNF over propositional variables $x_1, \ldots, x_n$. We construct a Stackelberg task, in which the leader decides the variable assignment, and the follower evaluates the chosen assignment so that it has a plan iff the leader's chosen assignment does not satisfy ϕ. The task is composed of the state variables $V = \{T_i, F_i \mid 1 \leq i \leq n\} \cup \{U\}$. The initial state is $I = \{T_i, F_i \mid 1 \leq i \leq n\}$. The follower's goal is $G = \{U\}$. The leader chooses the truth assignment by removing the unwanted value via either sel_i-T with $pre(sel_i\text{-}T) = \{T_i\}$ and $del(sel_i\text{-}T) = \{F_i\}$ or sel_i-F with $pre(sel_i\text{-}F) = \{F_i\}$ and $del(sel_i\text{-}F) = \{T_i\}$. The follower can evaluate each clause $C_k \in \phi$ via val_k where $add(val_k) = \{U\}$ and $pre(val_k) = \{F_i | x_i \in C_k\} \cup \{T_i | \neg x_i \in C_k\}$ (the negation of the clause). The construction obviously fulfills the syntactic restrictions. Moreover, the answer to STACKELSAT$^+_1$ is yes iff ϕ is satisfiable. □

Theorem 7. *STACKELSAT^0 is polynomial.*

Proof. Any $v \in V \setminus G$ can be ignored. Consider the set L^F of all follower actions $a^F \in A^F$ with $del(a^F) = \emptyset$. The last action of any follower plan must be an action $a^F \in L^F$, i.e., if $L^F = \emptyset$, the follower can only use the empty plan. Otherwise, the follower can always execute all $a^F \in L^F$ as its last actions. We can thus remove any $v \in add(a^F)$ for any $a^F \in L^F$ from consideration (remove it from G^F and V). We can now recalculate L^F and repeat this process until $L^F = \emptyset$. This process terminates after polynomially many steps. If at this point $G^F \not\subseteq I$, the follower has no plan for the empty leader plan. Otherwise, the follower has no plan iff there is an action $v \in G^F$ s.t. there is $a^L \in A^L$ with $v \in del(a^L)$. The leader plan is then a^L. □

Corrolary 1. *STACKELMIN$^1_{+1}$ is Σ^P_2-complete.*

Proof. Follows directly from Theorem 5. □

Theorem 8. *STACKELMIN$^{+1}_1$ is Σ^P_2-complete.*

Proof. Membership: As argued in Theorem 6, the consideration of polynomially long plans suffices to answer STACKELMIN$_1^{+1}$. Membership then follows via the procedure sketched in Theorem 3.

Hardness: Reduction from the satisfiability problem for restricted QBFs $\exists x_i \forall y_j \phi$, assuming ϕ to be in DNF. Let n be the number of x_i variables and m the number of y_j variables. For convenience of notation, we assume for this proof (and only this proof) that the y_j variables are numbered from y_{n+1} to y_{n+m}. Let k be the number of cubes in ϕ. The idea of our Stackelberg planning task construction is similar to all prior proofs. The state variables are $V = \{T_i, F_i | 1 \leq i \leq n+m\} \cup \{S_{n+i} | 1 \leq i \leq m\} \cup \{C_j | 1 \leq j \leq k\}$. The initial state is $I = \{T_i, F_i | 1 \leq i \leq n\}$. The follower's goal is $G^F = \{S_{n+i} | 1 \leq i \leq m\} \cup \{C_i | 1 \leq i \leq k\}$. The leader can choose the x_i truth assignments by removing the unwanted value ($1 \leq i \leq n$) via sel_i-T with $pre(sel_i\text{-}T) = \{T_i\}$ and $del(sel_i\text{-}T) = \{F_i\}$ and sel_i-F with $pre(sel_i\text{-}F) = \{F_i\}$ and $del(sel_i\text{-}F) = \{T_i\}$. The follower can choose the truth value for each y_j ($n+1 \leq i \leq n+m$) via sel_i-T with $add(sel_i\text{-}T) = \{T_i\}$ or sel_i-F with $add(sel_i\text{-}F) = \{F_i\}$. The follower can indicate that y_j has been assigned through ($n+1 \leq i \leq n+m$): via $done_i$-T with $pre(done_i\text{-}T) = \{T_i\}$ and $add(done_i\text{-}T) = \{S_i\}$ or $done_i$-F with $pre(done_i\text{-}F) = \{F_i\}$ and $add(done_i\text{-}F) = \{S_i\}$, and, finally, it can evaluate each cube c_j in ϕ through each of the literals $l_i \in c_k$ by val_j^i where $add(val_j^i) = \{C_j\}$ and if l_i is positive, then $pre(val_j^i) = \{F_i\}$ and otherwise if l_i is negative, then $pre(val_j^i) = \{T_i\}$. All actions have unit cost. Note that the construction satisfies the syntactic restrictions of STACKELMIN$_1^{+1}$. In order to reach its goal, the follower must execute one of the $done_i$ actions for each variable y_j, which in turn requires executing one of the sel_i actions for each variable y_j, and it must execute one of the val_j actions for each cube. Hence, there is no follower plan shorter than $2m+k$. Plans which assign some y_j variable multiple values are possible, but they have to be longer than $2m+k$. If the follower has a plan with exactly that length, then the formula ϕ can be falsified given the x_i assignments chosen by the leader. So, let $B^F = 2m+k+1$ and $B^L = n$. The latter suffices to allow the leader to choose an assignment for every x_i. The answer to STACKELMIN$_1^{+1}$ for these bounds is yes iff the QBF is satisfiable. □

Theorem 9. *STACKELMIN$_2^0$ is Σ_2^p-complete.*

Proof. Membership: Since actions have no preconditions, it never makes sense to execute an action more than once. As such, if a plan exists, a polynomially long plan exists as well. We can thus use the same algorithm as in Theorem 3.

Hardness: We again reduce from satisfiability of QBF formulae of the form $\exists x_i \forall y_j \phi$. We assume that ϕ is in DNF. We further assume that the variables x_i are numbered 1 to n and the y_j are numbered $n+1$ to $n+m$.

Let k be the total number of cubes in ϕ. Our Stackelberg task encoding follows once again also the same idea as before. The state variables are $V = \{notT_i^x, notF_i^x, S_i^x \mid 1 \leq i \leq n\} \cup \{notT_j^y, notF_j^y, S_j^y \mid n+1 \leq i \leq n+m\} \cup \{C_i \mid 1 \leq i \leq k\}$. The initial state is $\{notT_i^x, notF_i^x \mid 1 \leq i \leq n\} \cup \{notT_j^y, notF_j^y \mid n+1 \leq i \leq n+m\}$. The follower's goal is $G^F = \{notT_i^x, notF_i^x, S_i^x | 1 \leq i \leq n\} \cup \{notT_j^y, notF_j^y, S_j^y | n+1 \leq i \leq n+m\} \cup \{C_j | 1 \leq j \leq k\}$. We then add the following leader actions sel_i-T with $add(sel_i\text{-}T) = \{notF_i\}$ and $del(sel_i\text{-}T) = \{notT_i\}$ and sel_i-F with $del(sel_i\text{-}F) = \{notT_i\}$ and $del(sel_i\text{-}F) = \{notF_i\}$. For the follower, we add the following actions: (1) to assume the truth value of a variable (x_i or y_j) to be $B \in \{T, F\}$ ($1 \leq i \leq n+m$): $assume_i$-B with $add(assume_i\text{-}B) = \{S_i\}$ and $del(assume_i\text{-}B) = \{notB_i\}$, (2) to evaluate the i-th cube to false by using the assumption that literal $l_j \in C_i$ is false: $add(val_{C_i}^j) = \{C_i\}$ and if l_j is a positive literal, then $del(val_{C_i}^j) = \{notT_j\}$ and otherwise if it is a negative literal, then $del(val_{C_i}^j) = \{notF_j\}$. Note that if the assumption is indeed satisfied, the delete effect becomes a noop. (3) And finally, to revert an assumption: $revert_i$-B with $add(revert_i\text{-}B) = \{notB_i\}$ All actions have cost 1.

To reach the goal, the follower needs to perform three things: (1) Make an assumption about the value of every x_i and y_j variable. (2) Evaluate all cubes to false by picking one literal and forcing its negation to be true. (3) Unassign every variable by applying revert according to the deleted facts. All in all, each follower plan must contain at least $2(n+m)+k$ actions. If there is a plan with exactly this length, then all the chosen val_j actions had to use an already assumed variable-truth-value; and every variable must have exactly one assumed truth value; in particular, the follower plan must assume the truth value of the x_i variables that was chosen by the leader. Hence, each such plan corresponds to a violating assignment to ϕ. If, on the other hand, for the x_i assignment chosen by the leader $\forall y_j : \phi$ is true, the length of an optimal follower plan must exceed $2(n+m)+k$, as making false all cubes in ϕ then requires assuming both truth-values for at least one variable (meaning additional 2 actions). The answer to STACKELMIN$_2^0$ for $B^L = n$ and $B^F = 2(n+m)+k+1$ is yes iff the QBF is satisfiable. □

Theorem 10. *STACKELMIN$_1^0$ is NP-complete in general, but polynomial when additionally assuming unit cost.*

Proof. For the leader it only makes sense to execute actions with a deleting effect and for the follower actions with an adding effect. More specifically, let $G' := G \cap I$. In order to increase the plan cost of the follower, the leader needs to apply actions that delete some fact from G'. On the other hand, the follower has to apply an action for every $G \setminus G'$, and in addition an action for every fact from G' the leader has deleted. If all costs are equal, the leader either has to delete a state variable that the follower cannot add or the cost bound B^L and the available actions must allow to delete at least $B^F + |G'| - |G|$ many facts from G'. Otherwise the leader cannot solve the task. This can be checked in polynomial time. Suppose that actions may have non-unit cost.

Membership: We can non-deterministically guess a subset of the leader actions of cost at most B^L and execute them. From the resulting state s, the follower has to execute her actions that make the state variables in $G \setminus s$ true. We can select per variable the cheapest action and add the costs up. We return true if this is above B^F.

Hardness: We reduce from integer knapsack (Garey and Johnson 1979, MP10). Let $U = \{u_1, \dots, u_n\}$ be a set of objects, $s : U \mapsto \mathbb{N}^+$ be their sizes, $v : U \mapsto \mathbb{N}^+$ their values, B the size limit, and K the minimal desired total value. We construct a Stackelberg task following the same intuition as in the proof of Theorem 6: the leader picks a possible solution and the follower's plans correspond to the evaluation of this solution. We set facts V, initial state I, and goal G^F all to be the set of objects U, i.e., $V = I = G^F = U$. The leader has for every u_i an action sel_{u_i} with $del(sel_{u_i}) = \{u_i\}$ and cost $s(u_i)$. The follower has for every u_i an action $take_{u_i}$ with $add(takeu_i) = \{u_i\}$ and cost $v(u_i)$. We set $B^L = B$ and $B^F = K$. The leader's selection of sel_{u_i} actions encodes a set of objects $S \subseteq U$ fitting the size limit, i.e., $\sum_{u \in S} s(u) \leq B$. In order to achieve its goal, the follower needs to take (at least) all the objects selected by the leader, resulting in a cost of at least $\sum_{u \in S} v(u)$. Therefore, the leader selection is a solution to the bin-packing problem if the follower's optimal plan cost is at least $K = B^F$. The answer to STACKELMIN$_1^0$ is yes iff the bin-packing instance has a solution. □

Theorem 11. *METAOPVER is PSPACE-complete.*

Proof. Membership: Iterate over all states in Π (which only requires to store the currently considered state, i.e., can be done in polynomial space). For each state s: (1) check if $s \models pre(\sigma)$, and if so (2) check whether s is reachable from I, and if this is also the case, (3) check whether $s[\![\sigma]\!]$ is reachable from s. (1) can be clearly tested in polynomial space. (2) and (3) can be done in polynomial space with a small modification of the algorithm used to show plan existence in classical planning: instead of using the subset-based goal termination test, we enforce equality, terminating only at states t with (2) $t = s$ respectively (3) $t = s[\![\sigma]\!]$. We return true if (3) was satisfied for states tested, and false otherwise.

Hardness: We reduce from PLANSAT. Let $\Pi = \langle V, A, I, G \rangle$ be a classical planning task. Let g be a fresh state variable, and a_g be a fresh action. We create a new planning task $\Pi' = \langle V \cup \{g\}, A \cup \{a_g\}, I, \{g\} \rangle$ where $pre(a_g) = G$, $add(a_g) = \{g\}$, $del(a_g) = V$. Note that Π is solvable iff Π' is solvable. We define a new meta-operator σ for Π', setting $pre(\sigma) = \{p | p \in I\} \cup \{\neg p | p \in V \setminus I\}$, $add(\sigma) = \{g\}$, and $del(\sigma) = V$. Obviously, σ is a meta-operator for Π' iff Π' is solvable, what shows the claim. □

Theorem 12. *polyMETAOPVER is Π_2^P-complete.*

Proof. Membership: Membership in Π_2^P can be show by providing an alternating Turing Machine, which switches only once from universal to existential nodes during each run. Using universal nodes, we guess a plan of cost at most c_R, execute it (if possible), to reach a state s^P and check whether $s^P \models pre(\sigma)$. If not, return true (as we can not disprove validity with this trace). If $s^P \models pre(\sigma)$, then using existentially quantified decision nodes, guess a plan of cost at most c_M, check its applicabiltiy (else return false) and whether it reaches $s^P[[\sigma]]$. If so, return true, else false.

Hardness: We reduce from the respective restricted QBF satisfiability problem – which are formulae of the form $\forall x_i \exists y_j \phi$. We can assume that ϕ is in 3-CNF. We define the state variables

$$V = \{B\} \cup \{T_i^x, F_i^x, S_i^x \mid x_i\} \cup \{T_j^y, F_j^y, S_j^y \mid y_j\} \cup \{cl_i \mid \text{for every clause } i \text{ in } \phi\}$$

The initial state is $\{B\}$. We then define actions

- $sel_i^x\text{-}T$ with $pre(sel_i^x\text{-}T) = \{\neg S_i^x, B\}$ and $add(sel_i^x\text{-}T) = \{S_i^x, T_i^x\}$
- $sel_i^x\text{-}F$ with $pre(sel_i^x\text{-}F) = \{\neg S_i^x, B\}$ and $add(sel_i^x\text{-}F) = \{S_i^x, F_i^x\}$
- $do\text{-}block$ with $pre(do\text{-}block) = \{B\} \cup \{S_i^x \mid x_i\}$ and $del(do\text{-}block) = \{B\}$
- $sel_j^y\text{-}T$ with $pre(sel_j^y\text{-}T) = \{\neg S_j^y, \neg B\}$ and $add(sel_j^y\text{-}T) = \{S_j^y, T_j^y\}$
- $sel_j^y\text{-}F$ with $pre(sel_j^y\text{-}F) = \{\neg S_j^y, \neg B\}$ and $add(sel_j^y\text{-}F) = \{S_j^y, F_j^y\}$
- $val_{cl_i}^j$ with $add(val_{cl_i}^j) = \{cl_i\}$. Let l_j be the jth literal in the clause i.
  - If it is positive literal then $pre(val_{cl_i}^j) = \{\neg B, T_j^l\}$
  - If it is a negative literal, then $pre(val_{cl_i}^j) = \{\neg B, F_j^l\}$
- $re\text{-}block$ with $pre(re\text{-}block) = \{\neg B\} \cup \{S_j^y \mid y_j\}$, $add(re\text{-}block) = \{B\}$, and $del(re\text{-}block) = \{T_j^y, F_j^y \mid y_j\}$

All actions have cost 1.

We then ask, whether the meta operator σ with $pre(\sigma) = \{B\} \cup \{S_i^x \mid x_i\} \cup \{\neg S_j^y \mid y_j\}$ and $add(\sigma) = \{cl_i \mid \text{for every clause} i \text{ in } \phi\} \cup \{S_j^y \mid y_j\}$ is valid under the cost limits $c_R = |\{x_i \mid x_i\}|$ and $c_M = |\{y_j \mid y_j\}| + |\{i \mid \text{for every clause} i \text{ in } \phi\}| + 2$

We claim that the meta operator σ is valid if and only if the formula ϕ is satisfiable. To validate σ, we have to consider any reachable state s^P (with cost at most c_R) in which B, all the S_i^x, but none of the S_j^y are true. Since the block variable B has to be true in this state, we cannot have executed $do\text{-}block$ – otherwise we would also require a $re\text{-}block$ which exceeds together with the necessary sel^x action the cost limit c_R. Thus in any such state s^P, we have enforced that truth values for all the x_i variables have been selected, but for none of the y_j variables.

For σ to be valid, for any such s^P, we have to find a plan that reaches $s^P[[\sigma]]$. Given the effects of σ, this means that we have to select a value for all y_j variables and satisfy all clauses (via the cl_i variables). As the first action of any such plan, we have to perform $do\text{-}block$ – as all other actions (except the sel^x which we can't execute anyhow) require $\neg B$. We then have to select truth values for the variables y_j using the sel^y actions. At this point a single, non-modifiable valuation of the x_i and y_j has been chosen. Executing the appropriate selection of val_{cl_i} actions then marks all clauses as satisfied (if this is indeed the case). Lastly, the plan has use the $re\text{-}block$ action to clear the information on how we set the truth values for the y_j variables and to make the variable B true again. This is required as we have to reach $s^P[[\sigma]]$ exactly. In essence, the $re\text{-}block$ action allows us not to "leak" any information on how we selected the truth values of the y_j variables out of the execution of the meta operator.

If σ is valid then every valuation of the x_i corresponds to a reachable state s^P and the fact that σ is valid means that for every such valuation we can find a plan that sets the y_j in a way that all clauses in the formula are satisfied. If σ is not valid, we can on the other hand find a valuation of the x_i for which we cannot achieve the target state of σ thus it is impossible to set the y_j to satisfy the formula. Thus σ is valid if and only if the original formula is true. □

References

Bylander, T. 1994. The Computational Complexity of Propositional STRIPS Planning. *Artificial Intelligence*, 69(1): 165–204.

Garey, M. R.; and Johnson, D. S. 1979. *Computers and Intractability. A Guide to the Theory of NP-Completeness*. W. H. Freeman.

Immerman, N. 1988. Nondeterministic space is closed under complementation. *SIAM Journal on computing*, 17(5): 935–938.

Savitch, W. J. 1970. Relationships between nondeterministic and deterministic tape complexities. *Journal of computer and system sciences*, 4(2): 177–192.

Szelepcsényi, R. 1987. The method of forcing for nondeterministic automata. *Bulletin European Association for Theoretical Computer Science*, 33: 96–100.

On the Computational Complexity of Stackelberg Planning and Meta-Operator Verification

Primary Keywords: *(4) Theory*

Abstract

Stackelberg planning is a two-player variant of classical planning, in which one player tries to "sabotage" the other player in achieving its goal. This yields a bi-objective planning problem, which appears to be computationally more challenging than the single-player case. But is this actually true? All investigations so far focused on practical aspects, i.e., algorithms, and applications like cyber-security or very recently for meta-operator verification in classical planning. We close this gap by conducting the first theoretical complexity analysis of Stackelberg planning. We show that in general Stackelberg planning is no harder than classical planning. Under a polynomial plan-length restriction, however, Stackelberg planning is a level higher up in the polynomial complexity hierarchy, suggesting that compilations into classical planning come with an exponential plan-length increase. In attempts to identify tractable fragments exploitable, e.g., for Stackelberg planning heuristic design, we further study its complexity under various planning task restrictions, showing that Stackelberg planning remains intractable where classical planning is not. We finally inspect the complexity of the meta-operator verification, which in particular gives rise to a new interpretation as the dual problem of Stackelberg plan existence.

Introduction

Stackelberg planning (Speicher et al. 2018a) is a two-player variant of classical planning, where one player (the *leader*) tries to "sabotage" the other player (the *follower*). The leader moves first, committing to an action sequence, which subsequently the follower needs to complete to a plan. The leader's objective is maximizing the follower's optimal plan cost while minimizing her own cost. This type of planning is useful for real-world adversarial settings commonly found in the cyber-security domain (Speicher et al. 2018b; Di Tizio et al. 2023). *Leader-follower search* (Speicher et al. 2018a) is the so far only algorithm paradigm proposed for solving such tasks. In essence, it boils down to a search in the leader state space, solving for every visited leader state the follower's associated classical planning task. Given that exponentially such follower tasks must be solved in the worst case, one might wonder whether Stackelberg planning is in fact computationally more difficult than classical variant. Past work on Stackelberg planning however so far focused on algorithmic improvements rather than studying this question (Torralba et al. 2021; Sauer et al. 2023).

To close this gap, we present the first theoretical investigation of Stackelberg planning's complexity. We show that Stackelberg planning remains PSPACE-complete in general. However, Stackelberg planning with polynomial plan-length bounds is Σ_2^P-complete, contrasting the NP-completeness of the corresponding classical planning problem (Bylander 1994). Assuming that the polynomial hierarchy does not collapse, this suggests that compilations of Stackelberg planning into classical planning need to come with an exponential increase in plan length.

The analysis of tractable fragments has shown to be an important source for the development of domain-independent heuristic in classical planning (e.g., Hoffmann and Nebel 2001; Domshlak, Hoffmann, and Katz 2015). With the vision of establishing a basis for the development of leader-follower search heuristics, we analyze the complexity of Stackelberg planning under various syntactic restrictions. An overview of our results is given in Tab. 1.

Lastly, we explore a problem related to Stackelberg planning: *meta-operator* (Pham and Torralba 2023) verification. Meta-operators are action-sequence wild cards, which can be instantiated freely for every state satisfying the operator's precondition as long as operator's effects match. Pham and Torralba have cast verifying whether a given action is a valid meta-operator as a Stackelberg planning task. We show that meta-operator verification PSPACE-complete and Π_2^P-complete under a polynomial plan-length restriction. This gives rise to a new interpretation of the meta-operator verification as the dual problem of Stackelberg planning.

Note to Reviewers: This is a short-paper version without proofs. All proofs are in the supplement, which we will publish. Alternatively, if you so desire, we can include all proofs into a long version of paper (see alternative attached).

Background

Classical Planning We assume STRIPS notation (Fikes and Nilsson 1971). A planning task is a tuple $\Pi = \langle V, A, I, G\rangle$ consisting of a set of propositional *state variables* (or *facts*) V, a set of *actions* A, an *initial state* $I \subseteq V$, and a *goal* $G \subseteq V$. For $p \in V$, p and $\neg p$ are called *literals*. A *state* s is a subset of V, with the interpretation that all state variables not in s do not hold in s. Each action $a \in A$ has a *precondition* $pre(a)$, a conjunction of literals, an *add effect* (also called positive effect) $add(a) \subseteq V$, a *delete effect* (neg-

Syntactic restrictions	Plan existence		Optimal planning		METAOPVER
	PLANSAT	STACKELSAT	PLANMIN	STACKELMIN	
$*$ preconds $*$ effects $\lvert\pi\rvert$ not bounded	PSPACE	PSPACE (Theorem 1)	PSPACE	PSPACE (Theorem 2)	PSPACE (Theorem 11)
$*$ preconds $*$ effects $\lvert\pi\rvert \in \mathcal{O}(n^k)$	NP	Σ_2^{P} (Theorem 3)	NP	Σ_2^{P} (Theorem 3)	Π_2^{P} (Theorem 12)
1 precond 1+ effect	NP	Σ_2^{P} (Theorem 5)	NP	Σ_2^{P} (Corollary 1)	–
$*+$ preconds 1 effect	P	NP (Theorem 6)	NP	Σ_2^{P} (Theorem 8)	–
0 preconds 2 effects	P	P for ∞ effects (Theorem 7)	NP	Σ_2^{P} (Theorem 9)	–
0 preconds 1 effect non-unit cost	P	P for ∞ effects (Theorem 7)	P	NP (Theorem 10)	–

Table 1: Overview of our complexity results. For comparison, the PLANSAT and PLANMIN columns show the complexity of classical planning under the respective task restrictions, as given by (Bylander 1994). All results prove completeness with respect to the different complexity classes. $*$ means arbitrary number, $+$ only positive, $*+$ arbitrary positive, and $n+$ n positive.

ative effect) $del(a) \subseteq V$, and a non-negative *cost* $c(a) \in \mathbb{N}_0$. a is applicable in a state s iff $s \models pre(a)$. Executing a in s yields the state $s[\![a]\!] = (s \setminus del(a)) \cup add(a)$. These definitions are extended to action sequences π in an iterative manner. The cost of π is the sum of costs of its actions. π is called an *s-plan* if π is applicable in s and $G \subseteq s[\![\pi]\!]$. π is an *optimal* s-plan if $c(\pi)$ is minimal among all s-plans. An (optimal) plan for Π is an (optimal) I-plan. If there is no I-plan, we say that Π is *unsolvable*. Two decision problem formulations of classical planning are considered in the literature. *PLANSAT* is the problem of given a planning task Π, deciding whether there exists any plan for Π. *PLANMIN* asks, given in addition a (binary-encoded) cost bound B, whether there is a plan π for Π with cost $c(\pi) \leq B$. Both problems are known to be PSPACE-complete (Bylander 1994).

Stackelberg Planning A Stackelberg planning task (Speicher et al. 2018a) is a tuple $\Pi^{LF} = \langle V, A^L, A^F, I, G^F \rangle$, where the set of actions is partitioned into one for each player. A *leader plan* is an action sequence $\pi^L = \langle a_1^L, \ldots, a_n^L \rangle \in (A^L)^n$ that is applicable in I. π^L induces the *follower task* $\Pi^F(\pi^L) = \langle V, A^F, I[\![\pi^L]\!], G^F \rangle$. An (optimal) *follower response* to π^L is an (optimal) plan for $\Pi^F(\pi^L)$. We denote by $c^F(\pi^L)$ the cost of the optimal follower response to π^L, defining $c^F(\pi^L) = \infty$ if $\Pi^F(\pi^L)$ is unsolvable. Leader plans are compared via a dominance order between cost pairs where $\langle c_1^L, c_1^F \rangle$ *weakly dominates* $\langle c_2^L, c_2^F \rangle$ ($\langle c_1^L, c_1^F \rangle \sqsubseteq \langle c_2^L, c_2^F \rangle$), if $c_1^L \leq c_2^L$ and $c_1^F \geq c_2^F$. $\langle c_1^L, c_1^F \rangle$ (strictly) *dominates* $\langle c_2^L, c_2^F \rangle$ ($\langle c_1^L, c_1^F \rangle \sqsubset \langle c_2^L, c_2^F \rangle$), if $\langle c_1^L, c_1^F \rangle \sqsubseteq \langle c_2^L, c_2^F \rangle$ and $\langle c_1^L, c_1^F \rangle \neq \langle c_2^L, c_2^F \rangle$. To simplify notation, we write $\pi_1^L \sqsubset \pi_2^L$ if $\langle c(\pi_1^L), c^F(\pi_1^L) \rangle \sqsubset \langle c(\pi_2^L), c^F(\pi_2^L) \rangle$. A leader plan π^L is optimal if it is not dominated by any leader plan. Previous works have considered algorithms for computing the set of all optimal solutions, called the *Pareto frontier*.

Stackelberg Planning Decision Problems

We distinguish between two decision-theoretic formulations of Stackelberg planning, akin to classical planning:

Definition 1 (STACKELSAT)**.** *Given Π^{LF}, STACKELSAT is the problem of deciding whether there is a leader plan π^L that makes $\Pi^F(\pi^L)$ unsolvable.*

Definition 2 (STACKELMIN)**.** *Given Π^{LF}, and two binary-encoded numbers $B^L, B^F \in \mathbb{N}_0$. STACKELMIN is the problem of deciding whether there is a leader plan π^L with $\langle c(\pi^L), c^F(\pi^L) \rangle \sqsubseteq \langle B^L, B^F \rangle$.*

Interpreting the leader's objective as rendering the follower's objective infeasible, the first definition directly mirrors the PLANSAT plan-existence decision problem. Similarly, the second definition mirrors PLANMIN in looking for solutions matching a given quantitative cost bound. It is worth mentioning that both decision problems are implicitly looking for only a single point in the Pareto frontier, whereas previous practical works dealt with algorithms computing this frontier entirely. In terms of computational complexity, this difference is however unimportant. In particular, answering even just a single STACKELMIN question does in fact subsume the computation of the entire Pareto frontier – if the answer is no, one necessarily had to compare the given bounds to *every* element in the Pareto frontier.

As in classical planning, STACKELSAT can be easily (with polynomial overhead) reduced to STACKELMIN:

Proposition 1. *STACKELSAT is polynomially reducible to STACKELMIN.*

Proof. Let $\Pi^{LF} = \langle V, A^L, A^F, I, G^F \rangle$ be a Stackelberg task. Then STACKELSAT is true iff STACKELMIN is true, setting $B^L = 2^{|V|} \cdot \max_{a^L \in A^L} c(a^L)$ and $B^F = 2^{|V|} \cdot \max_{a^F \in A^L} c(a^F)$. Clearly, both bounds can be computed in time linear in the size of Π^{LF}. $\square$

Given that Stackelberg planning is a proper generalization of classical planning, the Stackelberg decision problems are guaranteed to be at least as hard as the respective classical planning decision problem. By applying the same proof idea as the Immerman–Szelepcsényi theorem (Szelepcsényi 1987; Immerman 1988), we can prove that it is also no harder than classical planning in the general case:

Theorem 1. *STACKELSAT is PSPACE-complete.*

Proof. Membership: By Savitch's theorem (Savitch 1970), we only have to prove membership in NPSPACE. We can non-deterministically guess a leader plan and compute the resulting state s^L. We then have to check that the follower's task $\langle V, A^F, s^L, G^F \rangle$ is unsolvable. For this, we use the same idea as in the Immerman–Szelepcsényi theorem (Szelepcsényi 1987; Immerman 1988): We know that classical plan existence is NPSPACE-complete. By Savitch's theorem, there then is a deterministic poly-space algorithm that determines classical plan existence. We can apply this deterministic algorithm to determine the solvability of $\langle V, A^F, s^L, G^F \rangle$. If it is not, we return true, otherwise false.

Hardness: We reduce from PLANSAT. Given a classical planning problem $\Pi = (V, A, I, G)$, we create the Stackelberg planning problem $\Pi^{LF} = (V, \emptyset, A, I, G)$, i.e., we treat all actions as follower actions. Apply a deterministic algorithm to solve STACKELSAT for Π^{LF}. If the answer was true, return false, otherwise return true. Since the leader cannot perform any action, STACKELSAT is true iff the original planning task was unsolvable. □

Theorem 2. *STACKELMIN is PSPACE-complete.*

Proof. Membership: Determining whether a given classical planning problem has a plan of cost at most c is PSPACE complete (Bylander 1994). As such there is a deterministic poly-space Turing machine that determines whether there is a plan of cost at most c for a given planning problem. To decide the base version of Stackelberg planning, we can now perform the following algorithm: (1) From the state I, non-deterministiaclly guess an applicable sequence of actions with cost at most c_L and compute the resulting state s^L. (2) Apply the deterministic algorithm to determine whether there is a plan of cost at most c_F in the classical planning problem $\Pi = (V, A^F, s^L, G)$. If not, return yes, otherwise no. This algorithm solves the decision variant of Stackelberg planning. Step (1) can be performed in polynomial space, as the sequence can never plausibly be longer than exponential.

Hardness: We reduce from the plan existence problem for classical planning. Given a classical planning problem $\Pi = (V, A, I, G)$, we create the Stackelberg planning problem $\Pi = (V, \emptyset, A, I, G)$, i.e., we treat all actions as follower actions. We set $c_L = 0$ and $c_F = 1 + 2^{|V|} \cdot \max_{a \in A} c(a)$. Since the leader cannot perform any action, if it is possible to force the follower cost above c_F, then the original planning problem was unsolvable. □

In spite of these results, algorithms for Stackelberg planning are significantly more complicated than their classical planning counterparts. In particular, the results raise the question of whether it is possible to leverage directly the classical planning methods for solving Stackelberg tasks via compilation. Polynomial compilations necessarily exist as per the theorems, yet, it is interesting to investigate which "side-effects" these might need to have. For example, it is possible any such compilation will have exponentially longer plan, rendering this approach infeasible in practice. In order to investigate these questions, we turn to a more fine granular analysis by considering the complexity under various previously studied syntactic classes of planning tasks.

Stackelberg Planning under Restrictions

Polynomial Plan Length

For classical planning, it is commonly known that restricting the length of the plans to be *polynomial* in the size of the planning task description, makes the decision problems become NP-complete.

Definition 3 (Polynomial Stackelberg Decision)**.** *Given Π^{LF} with non-0 action costs, and two binary-encoded numbers $B^L, B^F \in \mathbb{N}_0$ that are bounded by some polynomial $p \in \mathcal{O}(\ell^k)$ for $\ell = |V| + |A^L| + |A^F|$. STACKELPOLY is the problem of deciding whether there is a leader plan π^L such that $\langle c(\pi^L), c^F(\pi^L) \rangle \sqsubseteq \langle B^L, B^F \rangle$.*

We restrict the action cost to be strictly positive, ensuring that considering leader and follower plans with polynomial length is sufficient to answer the decision problem. STACKELPOLY is harder than the corresponding classical problem.

Theorem 3. *STACKELPOLY is Σ_2^P-complete.*

Proof. Membership: Membership in Σ_2^P can be shown by providing an alternating Turing Machine, which switches only once from existential to universal nodes during each run. Using existential nodes, we guess a leader plan π^L with cost of at most c^L, execute it (if possible), to reach a state $s^L = I[\![\pi^L]\!]$. As argued above, $|\pi^L|$ is polynomially bounded, so s^L can be computed in polynomial time. Once s^L is computed, we switch to universal nodes and then guess a follower plan π^F of cost at most c^F which is again at most polynomially long. We then determine whether π^F is applicable in s^L and whether $s^L[\![\pi^F]\!] \subseteq G$. If so we return false, otherwise true.

Hardness: We reduce from the corresponding restricted QBF problem – which is to determine whether formulae of the form $\exists x_i \forall y_j \phi$ are satisfiable. W.l.o.g. we can assume that ϕ is in DNF.[1] Let ψ_i be the ith cube of ϕ. We construct a Stackelberg task $\Pi^{LF} = \langle V, A^L, A^F, I, G^F \rangle$, in which the leader selects the x_i variable assignment, and the follower tries to find a y_j assignment making ϕ evaluate to false:

$$V = \{T_i^x, F_i^x, S_i^x \mid x_i\} \cup \{T_j^y, F_j^y, S_j^y \mid y_j\} \cup \{c_i \mid \psi_i \in \phi\}$$

The initial state is $I = \{\}$. The leader actions consists of:

- sel_i^x-T with $pre(sel_i^x\text{-}T) = \{\neg S_i^x\}$ and $add(sel_i^x\text{-}T) = \{S_i^x, T_i^x\}$
- sel_i^x-F with $pre(sel_i^x\text{-}F) = \{\neg S_i^x\}$ and $add(sel_i^x\text{-}F) = \{S_i^x, F_i^x\}$

The follower has the following actions

- sel_j^y-T with $pre(sel_j^y\text{-}T) = \{\neg S_j^y\}$ and $add(sel_j^y\text{-}T) = \{S_j^y, T_j^y\}$

[1] Satisfiability of $\exists x_i \forall y_j \phi$ is trivial if ϕ is in CNFs as tautology is trivial for CNFs.

- sel^y_j-F with $pre(sel^y_j\text{-}F) = \{\neg S^y_j\}$ and $add(sel^y_j\text{-}F) = \{S^y_j, F^y_j\}$
- $val^j_{c_i}$ with $add(val^j_{c_i}) = \{c_i\}$, where l_j is the j-th literal in the i-th cube.
  - If it is positive literal then $pre(val^j_{c_i}) = \{F^l_j\}$
  - If it is a negative literal, then $pre(val^j_{c_i}) = \{T^l_j\}$
- $valS^j_{c_i}$ with $add(valS^j_{c_i}) = \{c_i\}$ and $pre(valS^j_{c_i}) = \{\neg S^x_k\}$, where l_j is the j-th literal in the i-th cube, and $l_j \in \{x_k, \neg x_k\}$ for some k.

All actions have cost 1. We set the goal to $G = \{c_i \mid \text{for every cube } i \text{ in } \phi\}$. We lastly set $B^L = |\{x_i|i\}|$ and $B^F = |\{y_j \mid j\}| + \#cubes + 1$.

The leader chooses the x_i assignment by executing either sel^x_i-T or sel^x_i-F for every x_i variable. After that, the follower can select truth values of the y_j variables using the sel^y_j-T and sel^y_j-F actions, in attempts to make one of the val_{c_i} actions for every cube c_i applicable. If this is possible, the respective cubes must be violated. If all cubes evaluate to false, then so does the overall formula ϕ. The additional $valS^j_{c_i}$ actions are necessary to forces the leader to choose an assignment to all x_i variables. Otherwise, unassigned x_i variables could make it impossible for the follower to find violations to all cubes. The value of B^L allows the leader to choose an assignment for all x_i variables. If the follower can reach her goal, she obviously has a plan with cost less than B^F. If there is a leader plan π^L where $c^F(\pi^L) \geq B^F$, then the formula $\exists x_i \forall y_j \phi$ is satisfiable. □

This result strongly suggests that a compilation of Stackelberg planning into classical planning is in general not possible without an exponential blow-up of some kind. Namely, suppose it were possible to compile any Stackelberg planning task into classical planning in a way so that the size as well as the length of the plans of the classical planning task can be related polynomially to the size of the Stackelberg task. Suppose the plans of the Stackelberg task are polynomially bounded. Since polynomial length plan existence for classical planning is NP-complete, this would, together with our result, imply that NP = Σ^{P}_2, thus collapsing the polynomial hierarchy (Arora and Barak 2007, Theorem 5.6). As this is unlikely given out current knowledge, we hence surmise that such polynomial compilations do not exist. Or in other words: we know that an exponential blow-up in the computation is not avoidable in all circumstances.

Delete-Free Stackelberg Planning

Delete-free classical planning (Hoffmann and Nebel 2001), with its application to heuristic computation, is probably the class of planning tasks that probably has received most attention in planning literature. Formally, a planning task Π is called delete-free if (1) there are no negative preconditions, and (2) there are no delete effects.

Applying these assumptions to Stackelberg planning, the leader's actions can now only add facts to the state the following is starting in. As executability is monotone w.r.t. the state, any plan for the follower is a plan independent of the actions the leader executes. I.e. the leader is no longer able to affect any of the follower's options in any way, rendering this sub-class uninteresting for Stackelberg planning. The complexity of Stackelberg planning follows directly from the results for classical planning:

Theorem 4. *Let Π^{LF} be a delete-free Stackelberg task. STACKELSAT can be decided in polynomial time. STACKELMIN is NP-complete.*

Stackelberg Planning under Bylander's Syntactic Restrictions

Bylander (1994) studied the complexity of classical planning under various syntactic restrictions, drawing a concise borderline between planning's tractability and infeasibility. Bylander distinguishes between different planning task classes based on the number of action preconditions and effects, and the existence of negative preconditions or effects. Table 1 provides an overview of the main classes. Here, we take up his analysis and show that even for the classes where classical planning is tractable, Stackelberg may not be. We consider STACKELSAT and STACKELMIN in this order.

Definition 4. *Let $m, n \in \mathbb{N}_0 \cup \{\infty\}$. STACKELSAT$^m_n$ is the problem of deciding STACKELSAT for Stackelberg tasks so that $|\,pre(a)| \leq m$ and $|\,add(a)| + |\,del(a)| \leq n$ hold for all actions a. If m is preceded by "$+$", actions may not have negative preconditions. If n is preceded by "$+$", actions may not have delete effects. STACKELMIN^m_n is defined similarly.*

We omit m (n) if $m = \infty$ ($n = \infty$). We consider only cases where the classical-planning decision problems are in NP. Stackelberg planning is PSPACE-hard when classical planning is.

Plan Existence

Bylander (1994) has shown that PLANSAT is already NP-complete for tasks with actions that even have just a single precondition and a single effect. Here we show that the corresponding Stackelberg decision problem is even one step above in the polynomial hierarchy:

Theorem 5. *STACKELSAT$^1_{+1}$ is Σ^P_2-complete.*

Proof. Membership: As there are no delete effects, no action ever needs to be applied more than once. Hence, if a leader plan satisfying STACKELSAT$^1_{+1}$ exists, then there exists one whose size is polynomially bounded. The same also holds for the follower. To decide STACKELSAT$^1_{+1}$, we can thus use a similar approach as in Theorem 3.

Hardness: We show hardness again via a reduction from the satisfiability of restricted QBF of the form $\exists x_i \forall y_j \phi$, assuming ϕ to be in DNF. Similar to the proof of Theorem 3, the idea of our construction is to let the leader choose an assignment to x_i, which the follower needs to counter by finding an assignment to y_j that makes ϕ false.

The Stackelberg problem is defined as follows: The state variables are $V = \{T^x_i, T^y_j, C_k\}_{i,j,k}$ for appropriately ranging i, j, k. The initial state is $I = \emptyset$. The follower's goal is $G^F = \{C_k | \text{for each cube } k \text{ in } \phi\}$. The leader can choose the truth value for each x_i: via either sel^x_i-T with $pre(sel^x_i\text{-}T) = \{\neg F^x_i\}$ and $add(sel^x_i\text{-}T) = \{T^x_i\}$ or sel^x_i-F with $pre(sel^x_i\text{-}F) = \{\neg T^x_i\}$ and $add(sel^x_i\text{-}F) =$

$\{F_i^x\}$. The follower can choose the truth value for each y_j via either $sel_j^y\text{-}T$ with $pre(sel_j^y\text{-}T) = \{\neg F_j^y\}$ and $add(sel_j^y\text{-}T) = \{T_j^y\}$ or $sel_j^y\text{-}F$ with $pre(sel_j^y\text{-}F) = \{\neg T_j^y\}$ and $add(sel_j^y\text{-}F) = \{F_j^y\}$, and she can make false each cube c_k in ϕ via each literal $l_i \in c_k$ by $val^i_{c_k}$ where $add(val^i_{c_k}) = \{c_k\}$ and if l_i is positive, then $pre(val^i_{c_k}) = \{\neg T_i^l\}$, else if if l_i is negative, then $pre(val^i_{c_k}) = \{\neg F_i^l\}$. This task obviously satisfies the STACKELSAT$^1_{+1}$ planning task restrictions. Moreover, note that $\exists x_i \forall y_j \phi$ is satisfiable iff the answer to STACKELSAT$^1_{+1}$ is yes. □

Bylander (1994) has shown that PLANSAT is polynomial if only positive preconditions and only a single effect per action are allowed. Even under these restrictive conditions, STACKELSAT however still remains intractable:

Theorem 6. *STACKELSAT$^+_1$ is NP-complete.*

Proof. Membership: Due to the restrictions, no action needs to be executed more than once. Hence, as before, the consideration of polynomially length-bounded plans suffices for answering Stackelberg plan existence for this class of tasks. To solve STACKELSAT$^+_1$, non-deterministically choose a (polynomially bounded) leader plan π^L and construct the corresponding follower task $\Pi^F(\pi^L)$. This can be done in polynomial time. PLANSAT for $\Pi^F(\pi^L)$ can be answered in (deterministic) polynomial time (Bylander 1994). Return true if the follower task is unsolvable, otherwise return false.

Hardness: By reduction from Boolean satisfiability. Let ϕ be a CNF over propositional variables $x_1, \dots, x_n$. We construct a Stackelberg task, in which the leader decides the variable assignment, and the follower evaluates the chosen assignment so that it has a plan iff the leader's chosen assignment does not satisfy ϕ. The task is composed of the state variables $V = \{T_i, F_i \mid 1 \leq i \leq n\} \cup \{U\}$. The initial state is $I = \{T_i, F_i \mid 1 \leq i \leq n\}$. The follower's goal is $G = \{U\}$. The leader chooses the truth assignment by removing the unwanted value via either $sel_i\text{-}T$ with $pre(sel_i\text{-}T) = \{T_i\}$ and $del(sel_i\text{-}T) = \{F_i\}$ or $sel_i\text{-}F$ with $pre(sel_i\text{-}F) = \{F_i\}$ and $del(sel_i\text{-}F) = \{T_i\}$. The follower can evaluate each clause $C_k \in \phi$ via val_k where $add(val_k) = \{U\}$ and $pre(val_k) = \{F_i | x_i \in C_k\} \cup \{T_i | \neg x_i \in C_k\}$ (the negation of the clause). The construction obviously fulfills the syntactic restrictions. Moreover, the answer to STACKELSAT$^+_1$ is yes iff ϕ is satisfiable. □

Stackelberg plan-existence however becomes easy, when forbidding preconditions throughout. While this class of tasks seems to be trivial at first glance, optimal Stackelberg planning actually remains intractable as we show below.

Theorem 7. *STACKELSAT^0 is polynomial.*

Proof. Any $v \in V \setminus G$ can be ignored. Consider the set L^F of all follower actions $a^F \in A^F$ with $del(a^F) = \emptyset$. The last action of any follower plan must be an action $a^F \in L^F$, i.e., if $L^F = \emptyset$, the follower can only use the empty plan. Otherwise, the follower can always execute all $a^F \in L^F$ as its last actions. We can thus remove any $v \in add(a^F)$ for any $a^F \in L^F$ from consideration (remove it from G^F and V). We can now recalculate L^F and repeat this process until $L^F = \emptyset$. This process terminates after polynomially many steps. If at this point $G^F \not\subseteq I$, the follower has no plan for the empty leader plan. Otherwise, the follower has no plan iff there is an action $v \in G^F$ s.t. there is $a^L \in A^L$ with $v \in del(a^L)$. The leader plan is then a^L. □

Optimal Planning

As per Proposition 1, optimal planning is in general at least as hard as deciding plan existence. All intractability results shown for STACKELSAT carry over to STACKELMIN. As in all classes analyzed in the previous section, the consideration of polynomially length-bounded plans is sufficient for hardness, Σ_2^{P} yields a sharp upper bound to the complexity of STACKELMIN, as per Theorem 3. In particular:

Corrolary 1. *STACKELMIN$^1_{+1}$ is Σ_2^P-complete.*

Proof. Follows directly from Theorem 5. □

The results for STACKELSAT only provide a lower bound to the complexity of STACKELMIN. This lower bound may be strict as demonstrated by Thm. 8 and 9:

Theorem 8. *STACKELMIN$^{+1}_1$ is Σ_2^P-complete.*

Proof. Membership: As argued in Theorem 6, the consideration of polynomially long plans suffices to answer STACKELMIN$^{+1}_1$. Membership then follows via the procedure sketched in Theorem 3.

Hardness: Reduction from the satisfiability problem for restricted QBFs $\exists x_i \forall y_j \phi$, assuming ϕ to be in DNF. Let n be the number of x_i variables and m the number of y_j variables. For convenience of notation, we assume for this proof (and only this proof) that the y_j variables are numbered from y_{n+1} to y_{n+m}. Let k be the number of cubes in ϕ. The idea of our Stackelberg planning task construction is similar to all prior proofs. The state variables are $V = \{T_i, F_i | 1 \leq i \leq n+m\} \cup \{S_{n+i} | 1 \leq i \leq m\} \cup \{C_j | 1 \leq j \leq k\}$. The initial state is $I = \{T_i, F_i | 1 \leq i \leq n\}$. The follower's goal is $G^F = \{S_{n+i} | 1 \leq i \leq m\} \cup \{C_i | 1 \leq i \leq k\}$. The leader can choose the x_i truth assignments by removing the unwanted value ($1 \leq i \leq n$) via $sel_i\text{-}T$ with $pre(sel_i\text{-}T) = \{T_i\}$ and $del(sel_i\text{-}T) = \{F_i\}$ and $sel_i\text{-}F$ with $pre(sel_i\text{-}F) = \{F_i\}$ and $del(sel_i\text{-}F) = \{T_i\}$. The follower can choose the truth value for each y_j ($n+1 \leq i \leq n+m$) via $sel_i\text{-}T$ with $add(sel_i\text{-}T) = \{T_i\}$ or $sel_i\text{-}F$ with $add(sel_i\text{-}F) = \{F_i\}$. The follower can indicate that y_j has been assigned through ($n+1 \leq i \leq n+m$): via $done_i\text{-}T$ with $pre(done_i\text{-}T) = \{T_i\}$ and $add(done_i\text{-}T) = \{S_i\}$ or $done_i\text{-}F$ with $pre(done_i\text{-}F) = \{F_i\}$ and $add(done_i\text{-}F) = \{S_i\}$, and, finally, it can evaluate each cube c_j in ϕ through each of the literals $l_i \in c_k$ by val^i_j where $add(val^i_j) = \{C_j\}$ and if l_i is positive, then $pre(val^i_j) = \{F_i\}$ and otherwise if l_i is negative, then $pre(val^i_j) = \{T_i\}$. All actions have unit cost. Note that the construction satisfies the syntactic restrictions of STACKELMIN$^{+1}_1$. In order to reach its goal, the follower must execute one of the $done_i$ actions for each variable y_j, which in turn requires executing one of the sel_i actions for each variable y_j, and it must execute one of the val_j actions for each cube. Hence, there is no follower plan shorter than $2m + k$. Plans which assign some y_j variable multiple values are possible, but they have to be longer than $2m + k$. If the follower has a plan with exactly that length, then the formula ϕ can be falsified given the x_i assignments chosen by the leader. So, let $B^F = 2m + k + 1$ and $B^L = n$. The latter suffices to allow the leader to choose an assignment for every x_i. The answer to STACKELMIN$_1^{+1}$ for these bounds is yes iff the QBF is satisfiable. □

Theorem 9. *STACKELMIN$_2^0$ is Σ_2^p-complete.*

Proof. Membership: Since actions have no preconditions, it never makes sense to execute an action more than once. As such, if a plan exists, a polynomially long plan exists as well. We can thus use the same algorithm as in Theorem 3.

Hardness: We again reduce from satisfiability of QBF formulae of the form $\exists x_i \forall y_j \phi$. We assume that ϕ is in DNF. We further assume that the variables x_i are numbered 1 to n and the y_j are numbered $n + 1$ to $n + m$.

Let k be the total number of cubes in ϕ. Our Stackelberg task encoding follows once again also the same idea as before. The state variables are $V = \{notT_i^x, notF_i^x, S_i^x \mid 1 \leq i \leq n\} \cup \{notT_j^y, notF_j^y, S_j^y \mid n+1 \leq i \leq n+m\} \cup \{C_i \mid 1 \leq i \leq k\}$. The initial state is $\{notT_i^x, notF_i^x \mid 1 \leq i \leq n\} \cup \{notT_j^y, notF_j^y \mid n+1 \leq i \leq n+m\}$. The follower's goal is $G^F = \{notT_i^x, notF_i^x, S_i^x | 1 \leq i \leq n\} \cup \{notT_j^y, notF_j^y, S_j^y | n+1 \leq i \leq n+m\} \cup \{C_j | 1 \leq j \leq k\}$. We then add the following leader actions $sel_i\text{-}T$ with $add(sel_i\text{-}T) = \{notF_i\}$ and $del(sel_i\text{-}T) = \{notT_i\}$ and $sel_i\text{-}F$ with $del(sel_i\text{-}F) = \{notT_i\}$ and $del(sel_i\text{-}F) = \{notF_i\}$. For the follower, we add the following actions: (1) to assume the truth value of a variable (x_i or y_j) to be $B \in \{T, F\}$ ($1 \leq i \leq n+m$): $assume_i\text{-}B$ with $add(assume_i\text{-}B) = \{S_i\}$ and $del(assume_i\text{-}B) = \{notB_i\}$, (2) to evaluate the i-th cube to false by using the assumption that literal $l_j \in C_i$ is false: $add(val_{C_i}^j) = \{C_i\}$ and if l_j is a positive literal, then $del(val_{C_i}^j) = \{notT_j\}$ and otherwise if it is a negative literal, then $del(val_{C_i}^j) = \{notF_j\}$. Note that if the assumption is indeed satisfied, the delete effect becomes a noop. (3) And finally, to revert an assumption: $revert_i\text{-}B$ with $add(revert_i\text{-}B) = \{notB_i\}$ All actions have cost 1.

To reach the goal, the follower needs to perform three things: (1) Make an assumption about the value of every x_i and y_j variable. (2) Evaluate all cubes to false by picking one literal and forcing its negation to be true. (3) Unassign every variable by applying revert according to the deleted facts. All in all, each follower plan must contain at least $2(n + m) + k$ actions. If there is a plan with exactly this length, then all the chosen val_j actions had to use an already assumed variable-truth-value; and every variable must have exactly one assumed truth value; in particular, the follower plan must assume the truth value of the x_i variables that was chosen by the leader. Hence, each such plan corresponds to a violating assignment to ϕ. If, on the other hand, for the x_i assignment chosen by the leader $\forall y_j : \phi$ is true, the length of an optimal follower plan must exceed $2(n+m)+k$, as making false all cubes in ϕ then requires assuming both truth-values for at least one variable (meaning additional 2 actions). The answer to STACKELMIN$_2^0$ for $B^L = n$ and $B^F = 2(n+m)+k+1$ is yes iff the QBF is satisfiable. □

Optimal Stackelberg planning remains intractable even when all actions have no preconditions and may have only at most one effect.

Theorem 10. *STACKELMIN$_1^0$ is NP-complete in general, but polynomial when additionally assuming unit cost.*

Proof. For the leader it only makes sense to execute actions with a deleting effect and for the follower actions with an adding effect. More specifically, let $G' := G \cap I$. In order to increase the plan cost of the follower, the leader needs to apply actions that delete some fact from G'. On the other hand, the follower has to apply an action for every $G \setminus G'$, and in addition an action for every fact from G' the leader has deleted. If all costs are equal, the leader either has to delete a state variable that the follower cannot add or the cost bound B^L and the available actions must allow to delete at least $B^F + |G'| - |G|$ many facts from G'. Otherwise the leader cannot solve the task. This can be checked in polynomial time. Suppose that actions may have non-unit cost.

Membership: We can non-deterministically guess a subset of the leader actions of cost at most B^L and execute them. From the resulting state s, the follower has to execute her actions that make the state variables in $G \setminus s$ true. We can select per variable the cheapest action and add the costs up. We return true if this is above B^F.

Hardness: We reduce from integer knapsack (Garey and Johnson 1979, MP10). Let $U = \{u_1, \ldots, u_n\}$ be a set of objects, $s : U \mapsto \mathbb{N}^+$ be their sizes, $v : U \mapsto \mathbb{N}^+$ their values, B the size limit, and K the minimal desired total value. We construct a Stackelberg task following the same intuition as in the proof of Theorem 6: the leader picks a possible solution and the follower's plans correspond to the evaluation of this solution. We set facts V, initial state I, and goal G^F all to be the set of objects U, i.e., $V = I = G^F = U$. The leader has for every u_i an action sel_{u_i} with $del(sel_{u_i}) = \{u_i\}$ and cost $s(u_i)$. The follower has for every u_i an action $take_{u_i}$ with $add(takeu_i) = \{u_i\}$ and cost $v(u_i)$. We set $B^L = B$ and $B^F = K$. The leader's selection of sel_{u_i} actions encodes a set of objects $S \subseteq U$ fitting the size limit, i.e., $\sum_{u \in S} s(u) \leq B$. In order to achieve its goal, the follower needs to take (at least) all the objects selected by the leader, resulting in a cost of at least $\sum_{u \in S} v(u)$. Therefore, the leader selection is a solution to the bin-packing problem if the follower's optimal plan cost is at least $K = B^F$. The answer to STACKELMIN$_1^0$ is yes iff the bin-packing instance has a solution. □

Complexity of Meta Operator Verification

Pham and Torralba (2023) have recently leveraged Stackelberg planning for synthesizing *meta-operators* in classical planning. Meta-operators, like macro-actions (Fikes and Nilsson 1971), are artificial actions that aggregate the effect of action sequences, therewith introducing shortcuts in state-space search. Formally, we are given a classical planning task $\mathit{\Pi}$ and an action σ that is not in $\mathit{\Pi}$'s action set. σ is a *meta-operator for* $\mathit{\Pi}$ if, for every state $s \models pre(\sigma)$ that is reachable from I, there exists a sequence π of $\mathit{\Pi}$'s actions such that $s[\![\sigma]\!] = s[\![\pi]\!]$. Whether a given σ is a meta-operator can be *verified* by solving a Stackelberg planning task.

Here, we consider the question whether using an expressive and computationally difficult formalism like Stackelberg planning is actually necessary. For this, we determine the computational complexity of meta-operator synthesis and compare it to that of Stackelberg planning, and based on this analysis point out an interesting connection.

Definition 5 (Meta-Operator Verification)**.** *Given* $\mathit{\Pi}$ *and a fresh action* σ*. METAOPVER is the problem of deciding whether* σ *is a meta-operator for* $\mathit{\Pi}$*.*

Like for Stackelberg planning, the complexity of meta-operator verification in general remains the same as that of classical planning:

Theorem 11. *METAOPVER is PSPACE-complete.*

Proof. Membership: Iterate over all states in $\mathit{\Pi}$ (which only requires to store the currently considered state, i.e., can be done in polynomial space). For each state s: (1) check if $s \models pre(\sigma)$, and if so (2) check whether s is reachable from I, and if this is also the case, (3) check whether $s[\![\sigma]\!]$ is reachable from s. (1) can be clearly tested in polynomial space. (2) and (3) can be done in polynomial space with a small modification of the algorithm used to show plan existence in classical planning: instead of using the subset-based goal termination test, we enforce equality, terminating only at states t with (2) $t = s$ respectively (3) $t = s[\![\sigma]\!]$. We return true if (3) was satisfied for states tested, and false otherwise.

Hardness: We reduce from PLANSAT. Let $\mathit{\Pi} = \langle V, A, I, G\rangle$ be a classical planning task. Let g be a fresh state variable, and a_g be a fresh action. We create a new planning task $\mathit{\Pi}' = \langle V \cup \{g\}, A \cup \{a_g\}, I, \{g\}\rangle$ where $pre(a_g) = G$, $add(a_g) = \{g\}$, $del(a_g) = V$. Note that $\mathit{\Pi}$ is solvable iff $\mathit{\Pi}'$ is solvable. We define a new meta-operator σ for $\mathit{\Pi}'$, setting $pre(\sigma) = \{p | p \in I\} \cup \{\neg p | p \in V \setminus I\}$, $add(\sigma) = \{g\}$, and $del(\sigma) = V$. Obviously, σ is a meta-operator for $\mathit{\Pi}'$ iff $\mathit{\Pi}'$ is solvable, what shows the claim. □

In other words, meta-operator verification could as well be compiled directly into a classical rather than a Stackelberg planning task. But how difficult or effective would such a compilation be? To shed light on this question, we again turn to a length bounded version of the problem.

Definition 6 (Polynomial Meta-Operator Verification)**.** *Given* $\mathit{\Pi}$ *with non-*0 *action costs, a fresh action* σ*, and two binary-encoded numbers* $B^P, B^M \in \mathbb{N}_0$ *that are bounded by some polynomial* $p \in \mathcal{O}(\ell^k)$ *for* $\ell = |V| + |A|$*. polyMETAOPVER is the problem of deciding whether for all states* $s \models pre(\sigma)$ *reachable from* I *with a cost of at most* B^P*, there exists* π *with* $c(\pi) \le B^M$ *and* $s[\![\pi]\!] = s[\![\sigma]\!]$*.*

The parameters B^P and B^M define the perimeter around the initial state respectively the reached state under which the meta-operator conditions are to be verified. As for Stackelberg planning, we require that the cost of all actions is strictly positive, which together with the cost bounds ensures that the radius of the perimeter is polynomially bounded.

Polynomial meta-operator verification too is on the second level of the polynomial hierarchy. We again point out that, under the assumption that the polynomial hierarchy does not collapse, this result shows that all classical-planning encodings of meta-operator verification generally need to come with an exponential explosion of some kind.

Theorem 12. *polyMETAOPVER is* $\mathit{\Pi}_2^P$*-complete.*

Note that polyMETAOPVER is therefore in the co-complexity-class of polynomial Stackelberg plan-existence, i.e., they belong to co-classes on the same level of the polynomial hierarchy. This may not be surprising given that meta-operator verification can indeed be seen as the dual of Stackelberg plan existence: while the latter asks for the existence of a (leader) action sequence where all induced (follower) action sequences satisfy some property, meta-operator verification swaps the quantifiers.

Proof. Membership: Membership in $\mathit{\Pi}_2^P$ can be show by providing an alternating Turing Machine, which switches only once from universal to existential nodes during each run. Using universal nodes, we guess a plan of cost at most c_R, execute it (if possible), to reach a state s^P and check whether $s^P \models pre(\sigma)$. If not, return true (as we can not disprove validity with this trace). If $s^P \models pre(\sigma)$, then using existentially quantified decision nodes, guess a plan of cost at most c_M, check its applicabiltiy (else return false) and whether it reaches $s^P[[\sigma]]$. If so, return true, else false.

Hardness: We reduce from the respective restricted QBF satisfiability problem – which are formulae of the form $\forall x_i \exists y_j \phi$. We can assume that ϕ is in 3-CNF. We define the state variables

$$V = \{B\} \cup \{T_i^x, F_i^x, S_i^x \mid x_i\} \cup \{T_j^y, F_j^y, S_j^y \mid y_j\} \\ \cup \{cl_i \mid \text{for every clause } i \text{ in } \phi\}$$

The initial state is $\{B\}$. We then define actions

- $sel_i^x\text{-}T$ with $pre(sel_i^x\text{-}T) = \{\neg S_i^x, B\}$ and $add(sel_i^x\text{-}T) = \{S_i^x, T_i^x\}$
- $sel_i^x\text{-}F$ with $pre(sel_i^x\text{-}F) = \{\neg S_i^x, B\}$ and $add(sel_i^x\text{-}F) = \{S_i^x, F_i^x\}$
- $do\text{-}block$ with $pre(do\text{-}block) = \{B\} \cup \{S_i^x \mid x_i\}$ and $del(do\text{-}block) = \{B\}$
- $sel_j^y\text{-}T$ with $pre(sel_j^y\text{-}T) = \{\neg S_j^y, \neg B\}$ and $add(sel_j^y\text{-}T) = \{S_j^y, T_j^y\}$
- $sel_j^y\text{-}F$ with $pre(sel_j^y\text{-}F) = \{\neg S_j^y, \neg B\}$ and $add(sel_j^y\text{-}F) = \{S_j^y, F_j^y\}$
- $val_{cl_i}^j$ with $add(val_{cl_i}^j) = \{cl_i\}$. Let l_j be the jth literal in the clause i.
  - If it is positive literal then $pre(val_{cl_i}^j) = \{\neg B, T_j^l\}$
  - If it is a negative literal, then $pre(val_{cl_i}^j) = \{\neg B, F_j^l\}$
- $re\text{-}block$ with $pre(re\text{-}block) = \{\neg B\} \cup \{S_j^y \mid y_j\}$, $add(re\text{-}block) = \{B\}$, and $del(re\text{-}block) = \{T_j^y, F_j^y \mid y_j\}$

All actions have cost 1.

We then ask, whether the meta operator σ with $pre(\sigma) = \{B\} \cup \{S^x_i \mid x_i\} \cup \{\neg S^y_j \mid y_j\}$ and $add(\sigma) = \{cl_i \mid \text{for every clause} i \text{ in } \phi\} \cup \{S^y_j \mid y_j\}$ is valid under the cost limits $c_R = |\{x_i \mid x_i\}|$ and $c_M = |\{y_j \mid y_j\}| + |\{i \mid \text{for every clause} i \text{ in } \phi\}| + 2$

We claim that the meta operator σ is valid if and only if the formula ϕ is satisfiable. To validate σ, we have to consider any reachable state s^P (with cost at most c_R) in which B, all the S^x_i, but none of the S^y_j are true. Since the block variable B has to be true in this state, we cannot have executed $do\text{-}block$ – otherwise we would also require a $re\text{-}block$ which exceeds together with the necessary sel^x action the cost limit c_R. Thus in any such state s^P, we have enforced that truth values for all the x_i variables have been selected, but for none of the y_j variables.

For σ to be valid, for any such s^P, we have to find a plan that reaches $s^P[[\sigma]]$. Given the effects of σ, this means that we have to select a value for all y_j variables and satisfy all clauses (via the cl_i variables). As the first action of any such plan, we have to perform $do\text{-}block$ – as all other actions (except the sel^x which we can't execute anyhow) require $\neg B$. We then have to select truth values for the variables y_j using the sel^y actions. At this point a single, non-modifiable valuation of the x_i and y_j has been chosen. Executing the appropriate selection of val_{cl_i} actions then marks all clauses as satisfied (if this is indeed the case). Lastly, the plan has use the $re\text{-}block$ action to clear the information on how we set the truth values for the y_j variables and to make the variable B true again. This is required as we have to reach $s^P[[\sigma]]$ exactly. In essence, the $re\text{-}block$ action allows us not to "leak" any information on how we selected the truth values of the y_j variables out of the execution of the meta operator.

If σ is valid then every valuation of the x_i corresponds to a reachable state s^P and the fact that σ is valid means that for every such valuation we can find a plan that sets the y_j in a way that all clauses in the formula are satisfied. If σ is not valid, we can on the other hand find a valuation of the x_i for which we cannot achieve the target state of σ thus it is impossible to set the y_j to satisfy the formula. Thus σ is valid if and only if the original formula is true. □

We want to point out that the duality between METAOPVER and STACKELSAT can be exploited further, showing analogous results for Bylander's (1994) task classes. Contrary to Stackelberg planning, however, the identification of tractable fragments is less useful for meta-operator verification due to the lack of the monotonicity invariance of the meta-operator condition. An action being a meta-operator in a task abstraction does not imply that the action is a meta-operator in the original task, and vice versa. We hence do not further explore this analysis here.

Related Work

Stackelberg planning is related to conformant and conditional planning, extensions of classical planning by state and/or action outcome uncertainty. A conformant plan is a sequence of actions that will reach a goal state – for any possible initial state and action outcome. In contrast, a conditional plan, is a tree-shaped structure that allows for different plans, depending on observations. Under the restriction to deterministic actions, both can be seen as a special case of Stackelberg planning using the leader-reachable states as an encoding of the initial belief. With this interpretation, STACKELSAT is false iff the conditional planning task is solvable. If the follower is restricted to use the same plan independent of the leader actions, we would have a model for conformant planning.

In the general case, conditional planning under partial observability and with conditional effects is EXPSPACE complete (Rintanen 2004). Both conformant and conditional planning have been investigated under the restriction to only polynomially long plans, like we did here. Rintanen (1999) showed that polynomially-length-bounded conditional STRIPS planning Π^P_2 complete, the co-result to our Thm. 3. His hardness proof uses a similar proof idea as ours, with technical differences owed to the different planning formalism. Bonet (2010) studied conditional planning with non-determinstic actions, proving that polynomially bounded plan existence for conditional plans with at most k branching points is Σ^P_{2k+k}-complete. Stackelberg planning corresponds $k = 1$, the difference between determinism and non-determinism causing the Σ^P_2 vs. Σ^P_4 complexity results.

For conformant planning, Baral, Kreinovich, and Trejo (2000) showed that plan existence is Σ^2_P-complete, if conditional effects are allowed. Turner (2002) considered conditional and conformant planning, but his formalism supported arbitrary boolean formulae as conditions, making length-1 plan existence already NP-complete.

No prior work on conformant/conditional planning considered any of Bylander's syntactical restrictions. Further, Stackelberg planning differs from conditional/conformant planning in using a more complex compact description of the "relevant" states through reachability.

Conclusion

Stackelberg planning remains PSPACE-complete like classical planning in general, but is Σ^P_2 complete under a polynomial plan-length bound. Hence, unless the polynomial hierarchy collapses at its first level, it is not possible to compile Stackelberg planning into classical planning without exponential blow-up. We showed that Stackelberg planning remains intractable even under various syntactical restrictions. Lastly, we have proven similar results for meta-operator verification, showing that it is PSPACE-complete in general and Π^P_2-complete for the polynomial plan-length bounded case, implying the same type of results for it.

References

Arora, S.; and Barak, B. 2007. *Computational Complexity: A Modern Approach.* Princeton, Online Draft Version.

Baral, C.; Kreinovich, V.; and Trejo, R. 2000. Computational complexity of planning and approximate planning in the presence of incompleteness. *Artificial Intelligence*, 122(1-2): 241–267.

Bonet, B. 2010. Conformant plans and beyond: Principles and complexity. *Artificial Intelligence*, 174(3-4): 245–269.

Bylander, T. 1994. The Computational Complexity of Propositional STRIPS Planning. *Artificial Intelligence*, 69(1): 165–204.

Di Tizio, G.; Speicher, P.; Simeonovski, M.; Backes, M.; Stock, B.; and Künnemann, R. 2023. Pareto-optimal defenses for the web infrastructure: Theory and practice. *ACM Transactions on Privacy and Security*, 26(2): 1–36.

Domshlak, C.; Hoffmann, J.; and Katz, M. 2015. Red-black planning: A new systematic approach to partial delete relaxation. *Artif. Intell.*, 221: 73–114.

Fikes, R.; and Nilsson, N. 1971. STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving. *Artificial Intelligence*, 2(3–4): 189–208.

Garey, M. R.; and Johnson, D. S. 1979. *Computers and Intractability. A Guide to the Theory of NP-Completeness*. W. H. Freeman.

Hoffmann, J.; and Nebel, B. 2001. The FF Planning System: Fast Plan Generation Through Heuristic Search. *Journal of Artificial Intelligence Research (JAIR)*, 14: 2531–302.

Immerman, N. 1988. Nondeterministic space is closed under complementation. *SIAM Journal on computing*, 17(5): 935–938.

Pham, F.; and Torralba, A. 2023. Can I Really Do That? Verification of Meta-Operators via Stackelberg Planning. In *Proceedings of the 32nd International Joint Conference on Artificial Intelligence (IJCAI 2023)*. IJCAI.

Rintanen, J. 1999. Constructing conditional plans by a theorem-prover. *Journal of Artificial Intelligence Research*, 10: 323–352.

Rintanen, J. 2004. Complexity of Planning with Partial Observability. In *Proceedings of the 14th International Conference on Automated Planning and Scheduling (ICAPS 2004)*, 345–354. AAAI Press.

Sauer, P.; Steinmetz, M.; Künnemann, R.; and Hoffmann, J. 2023. Lifted Stackelberg Planning. In *Proceedings of the 33rd International Conference on Automated Planning and Scheduling (ICAPS 2023)*. AAAI Press.

Savitch, W. J. 1970. Relationships between nondeterministic and deterministic tape complexities. *Journal of computer and system sciences*, 4(2): 177–192.

Speicher, P.; Steinmetz, M.; Backes, M.; Hoffmann, J.; and Künnemann, R. 2018a. Stackelberg planning: Towards effective leader-follower state space search. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI 2018)*, 6286–6293. AAAI Press.

Speicher, P.; Steinmetz, M.; Künnemann, R.; Simeonovski, M.; Pellegrino, G.; Hoffmann, J.; and Backes, M. 2018b. Formally reasoning about the cost and efficacy of securing the email infrastructure. In *2018 IEEE European Symposium on Security and Privacy (EuroS&P)*, 77–91. IEEE.

Szelepcsényi, R. 1987. The method of forcing for nondeterministic automata. *Bulletin European Association for Theoretical Computer Science*, 33: 96–100.

Torralba, Á.; Speicher, P.; Künnemann, R.; Steinmetz, M.; and Hoffmann, J. 2021. Faster Stackelberg Planning via Symbolic Search and Information Sharing. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI 2021)*, 11998–12006. AAAI Press.

Turner, H. 2002. Polynomial-length planning spans the polynomial hierarchy. In *Logics in Artificial Intelligence: 8th European Conference (JELIA 2002)*, 111–124. Springer.