Inapproximability of *STRIPS* Planning

Xing Tan[1], **Alban Grastien**[2]

[1]Department of Computer Science, Lakehead University, Canada
[2]Université Paris-Saclay, CEA, List, F-91120, Palaiseau, France
xing.tan@lakeheadu.ca, alban.grastien@cea.fr

Abstract

Automated planning involves finding a sequence of actions that changes the world from an initial state to a final state with goals satisfied. The general problem is *PSPACE*-hard. Nevertheless, many restricted variants are *NP*-complete or even in *P*. Existing complexity work focuses mostly on plan existence, or plan with minimal plan length. Little is known about optimization variants that aim to satisfy as many goal conditions as possible. In this paper, we aim to fill this gap by providing a first inapproximability study of goal-maximization using the classical *STRIPS* formalism. For *MAX-PLANSAT* and its length-bounded counterpart *MAX-PLANSAT*(K), we prove tight constant-factor lower bounds. More specifically, through performing L-reductions from *MAXE3SAT* and *MAX3DM*, we show several of these problems are inapproximable by a constant factor, unless $P = NP$.

1 Introduction

Automated planning seeks a sequence of actions that moves the world from an initial state to one that satisfies specified goals. Since the seminal *STRIPS* system (Fikes and Nilsson 1971, 1993), the field has expanded to adopt richer action languages such as ADL (Pednault 1989) and the PDDL standard (McDermott et al. 1998; Haslum et al. 2019); incorporate temporal and resource reasoning (Allen 1984; Dean and Boddy 1988; Wilkins 2014); and exploit hierarchical task-network formalisms (Tate 1976, 1977; Erol, Hendler, and Nau 1994, 1996; Bercher, Alford, and Höller 2019). These advances are complemented by heuristic and constraint-based planners that scale to real-world problems, from factory scheduling to spacecraft control (McDermott 1996; Bonet and Geffner 1999, 2000; Hoffmann 2001; Hoffmann and Nebel 2001; Helmert 2006; Kautz and Selman 1996; Blum and Furst 1997; Muscettola et al. 1998; Blazewicz, Pesch, and Schmidt 2019).

On the theoretical side, substantial effort over the past decades has been devoted to characterizing the computational complexity of planning tasks (Bercher, Haslum, and Muise 2024). Bylander's results established that propositional *STRIPS* planning is *PSPACE*-complete in general, with many syntactic fragments remaining *NP*-complete, and several highly restricted subclasses solvable in polynomial time (Bylander 1991, 1994). Work by Bäckström, Nebel, and others further refined this landscape by identifying additional tractable subclasses, providing parameterized complexity classifications, and exploiting structural restrictions such as causal graphs (Nebel and Bäckström 1994; Bäckström and Nebel 1995; Bäckström et al. 2013). While optimization aspects have been studied extensively for minimizing plan length, comparatively little attention has been given to objectives aimed at maximizing the number of goals achieved. Consequently, the approximability of these goal-maximization variants is essentially unexplored to this end. We aim to fill this gap by proving, to the best of our knowledge, the first set of constant-factor hardness results for goal-maximization in *STRIPS* planning.

We investigate the approximability of propositional *STRIPS* planning, focusing on two natural optimization variants: *MAX-PLANSAT*, which aims to find a plan that satisfies the maximum number of goal conditions, and its length-bounded counterpart *MAX-PLANSAT*(K), which restricts plans to a maximum length of K. We show that even highly restricted fragments, where operators have a bounded number of preconditions and effects, and/or are limited to only positive or only negative literals, remain inapproximable within any factor better than certain constants, unless $P = NP$. The proofs employ approximation-preserving L-reductions (Papadimitriou and Yannakakis 1991) from canonical hard problems *MAXE3SAT* and *MAX3DM*, thereby transferring lower bounds to the planning domain. Our results particularly expose fundamental barriers to computing near-optimal plans, even in fragments whose decision versions are polynomial-time solvable, and underscore the continued need for structural restrictions and powerful heuristics in scalable planning systems.

Our work aligns closely with oversubscription (partial satisfaction) planning, in which a planner must select a subset of soft goals to achieve under limited resources or plan-length bounds (Smith 2004; van den Briel et al. 2004; Katz and Mirkis 2016; Olaya, de la Rosa, and Borrajo 2021). Recent work on oversubscription planning relies on exact search rather than approximation algorithms, e.g., symbolic search (Speck and Katz 2021) and bound-sensitive A* heuristics (Katz and Keyder 2022). Surveys of plan optimization and bounded plan existence, however, emphasize the lack of worst-case guarantees for such goal-maximization objectives (Lin et al. 2024; Bercher, Haslum, and Muise 2024). Our inapproximability bounds explain this situation formally. That is, even for simple *STRIPS* fragments whose decision versions are in P, no polynomial-time algorithm can guarantee solutions that are close to optimal, unless *P* = *NP*.

The remainder of the paper is structured as follows. Section 2 provides background and necessary preliminaries. In Section 3, we present our main technical contributions. Finally, Section 4 concludes the paper.

2 Background

In this section, we review the formal definitions of *STRIPS* planning, key concepts from inapproximability theory, and the two canonical *NP*-hard optimization problems, *MAXE3SAT* and *MAX3DM*, that we later reduce from.

2.1 Propositional *STRIPS* Planning

We follow the notation of Bylander (Bylander 1994), allowing our inapproximability results to be compared line-for-line with his decision-complexity map. With this model, *conditions* are unsigned propositional atoms, and *literals* are signed occurrences (positive or negated) of conditions.

Definition 1 (Planning Problem). *A planning problem* Π_p *is described by a 4-tuple* $\langle \mathcal{P}, \mathcal{O}, \mathcal{I}, \mathcal{G} \rangle$*, where* $\mathcal{P}$ *is a finite set of propositional atoms, called conditions; The set* $\mathcal{O}$ *is a finite collection of operators. Each operator* $o \in \mathcal{O}$ *is defined by a pair of two sets:*

- *The set of preconditions: a conjunction of literals (positive and negative, written as* o^+ *and* o^-*, respectively) that must be satisfied to apply* o*;*
- *The set of postconditions: a conjunction of literals indicating which facts become true (add list, written as* o_+*) or false (delete list, written as* o_-*) upon execution.*

The set $\mathcal{I} \subseteq \mathcal{P}$ *denotes the initial state, listing which conditions are initially true. All other conditions (i.e., the conditions in* $\mathcal{P} \setminus \mathcal{I}$*) are false in the initial state. The 2-tuple* $\mathcal{G} = \langle \mathcal{G}_+, \mathcal{G}_- \rangle$ *specifies the goal, which consists of the desired positive and negative occurrences of conditions (specified in* $\mathcal{G}_+$ *and* $\mathcal{G}_-$*, respectively).*

A *state* is represented as a subset $\mathcal{S} \subseteq \mathcal{P}$. A condition p is *true in state* $\mathcal{S}$ if and only if $p \in \mathcal{S}$, and *false in* $\mathcal{S}$ otherwise. The goal is a pair $\mathcal{G} = \langle \mathcal{G}_+, \mathcal{G}_- \rangle$ with $\mathcal{G}_+, \mathcal{G}_- \subseteq \mathcal{P}$, where $\mathcal{G}_+$ contains the conditions that must be true, and $\mathcal{G}_-$ those that must be false in a solution state. A state $\mathcal{S}$ satisfies $\mathcal{G} = \langle \mathcal{G}_+, \mathcal{G}_- \rangle$, if $\mathcal{G}_+ \subseteq \mathcal{S}$ and $\mathcal{G}_- \cap \mathcal{S} = \emptyset$.

The result of applying a single operator o to a state $\mathcal{S}$, denoted by "Rst($\mathcal{S}, o$)", is defined as

$$\text{Rst}(\mathcal{S}, o) = \begin{cases} (\mathcal{S} \cup o_+) \setminus o_- & o^+ \subseteq \mathcal{S};\ o^- \cap \mathcal{S} = \emptyset \\ \mathcal{S} & \text{otherwise,} \end{cases}$$

where o^+ and o^- denote positive and negative preconditions respectively, while o_+ and o_- are the sets of conditions added and removed from the state, respectively. An operator can be applied to any state $\mathcal{S}$, however, it will have effects on $\mathcal{S}$ only if its preconditions are satisfied in $\mathcal{S}$.

The state transition is extended by applying a sequence of operators $(o_1, \ldots, o_n)$ on state $\mathcal{S}$:

$$\text{Rst}(\mathcal{S}, (o_1, \ldots, o_n)) = \text{Rst}(\text{Rst}(\cdots \text{Rst}(\mathcal{S}, o_1), o_2), \ldots, o_n).$$

Definition 2 (Plan). *A sequence of operators* $(o_1, \ldots, o_n)$ *is a plan.*

Definition 3 (Solution). *A plan is a solution to the planning problem* Π_p *if the sequence transforms the initial state* $\mathcal{I}$ *in planning problem* Π_p *into a state* $\mathcal{S}$*, where* $\mathcal{G}_+ \subseteq \mathcal{S}$*, and* $\mathcal{G}_- \cap \mathcal{S} = \emptyset$*.*

Having formalized what it means for a plan to be a solution, we now turn to several important computational problems that ask whether such a solution exists, whether it can be achieved within a bounded number of steps, or how many goal conditions can be satisfied under various constraints.

Definition 4 (*PLANSAT*).
Instance: *A planning problem instance* $\Pi_p = \langle \mathcal{P}, \mathcal{O}, \mathcal{I}, \mathcal{G} \rangle$.
Question: *Is there a sequence of operations that transforms the initial state into a state that satisfies the goal condition?*

To account for time and resource limits, a natural variant of *PLANSAT* is to constrain the maximum length of the plan.

Definition 5 (*PLANSAT*(K)).
Instance: *A planning problem instance* $\Pi_p = \langle \mathcal{P}, \mathcal{O}, \mathcal{I}, \mathcal{G} \rangle$ *and an integer* $K \geq 0$.
Question: *Is there a sequence of at most* K *operations that transforms the initial state into a state that satisfies* $\mathcal{G}$*?*

In practical scenarios, meanwhile, it may be infeasible to satisfy all goal conditions. A natural optimization variant of *PLANSAT* therefore seeks to maximize the number of goal conditions that are achieved.

Definition 6 (*MAX-PLANSAT*).
Instance: *A planning problem instance* $\Pi_p = \langle \mathcal{P}, \mathcal{O}, \mathcal{I}, \mathcal{G} \rangle$.
Objective: *Find a sequence of operations that transforms the initial state into a state* $\mathcal{S}$ *such that the number of goal conditions from* $\mathcal{G}$ *satisfied in* $\mathcal{S}$ *is maximized.*

To combine both resource constraints and partial goal satisfaction, we define further the following problem.

Definition 7 (*MAX-PLANSAT*(K)).
Instance: *A planning problem instance* $\Pi_p = \langle \mathcal{P}, \mathcal{O}, \mathcal{I}, \mathcal{G} \rangle$ *and an integer* $K \geq 0$.
Objective: *Find a sequence of at most* K *actions that transforms the initial state into a state* $\mathcal{S}$ *such that the number of goal conditions from* $\mathcal{G}$ *satisfied in* $\mathcal{S}$ *is maximized.*

For all the planning problems defined above, we may also consider their restricted variants, which impose structural limits on the number of preconditions and postconditions allowed in each operator. We provide below a formal definition for the restricted variant of *PLANSAT*. The restricted versions of other problems, such as *PLANSAT*(K), *MAX-PLANSAT*, and *MAX-PLANSAT*(K), can be defined similarly.

Definition 8 (*PLANSAT*$_{\varepsilon}^{\rho}$).
Instance: *A planning problem instance* Π_p *satisfying the following restrictions:*

- *Restriction ρ on the number or type of preconditions (e.g., at most one precondition per action),*
- *Restriction ε on the number or type of effects (i.e., post-conditions).*

Question: *Is there a sequence of operations that transforms the initial state into a state that satisfies the goal condition?*

Notation convention: Symbol $+$ indicates that only positive conditions are permitted. Symbol $-$ indicates that only negative conditions are permitted. A number (e.g., 1, 2) specifies a bound on the number of conditions allowed. If left unspecified, it means that the component is *unbounded*. For example, $PLANSAT^{1}_{2+}$ denotes the class of planning problems where each operator has at most one precondition and at most two positive postconditions. $PLANSAT^{+}_{1}$ denotes problems where each operator may have an unbounded number of positive preconditions and at most one postcondition, either positive or negative.

2.2 Inapproximability Theory

This section provides standard preliminaries on inapproximability theory (Papadimitriou and Yannakakis 1991).

Definition 9. *Let α be a ratio, called an optimization ratio. An α-approximation algorithm for an optimization problem is an algorithm that returns a solution within an α factor of the optimal solution in polynomial time.*

We use the convention that $\alpha \geq 1$ for minimization problems, and $\alpha \leq 1$ for maximization problems. For example, a $\frac{2}{3}$-approximation algorithm for a maximization problem is a polynomial-time algorithm that returns a solution for any instance of the problem, and the solution is at least $\frac{2}{3}$ of the optimal solution for the instance. A $\frac{3}{2}$-approximation algorithm for a minimization problem is a polynomial-time algorithm that returns a solution at most $\frac{3}{2}$ optimal.

Definition 10 (L-reduction with parameters a and b)**.** *Two positive numbers $a > 0$ and $b > 0$, and two maximization problems Π and Π' are given. A reduction from Π to Π' is an L-reduction (also called linear reduction, or approximation-preserving reduction), if the following three conditions are satisfied*

1. *For any instance I of Π, an instance I' of Π' can be computed in polynomial time through this reduction.*
2. *Let $OPT(I)$ be the value of a maximum solution to I of Π, and $OPT(I')$ the one for the constructed I' of Π', the inequality $OPT(I') \leq a \cdot OPT(I)$ holds.*
3. *For any solution of value V' to I', a solution of value V to I can be computed accordingly in polynomial time such that $|OPT(I) - V| \leq b \cdot |OPT(I') - V'|$.*

Theorem 1. *(Papadimitriou and Yannakakis 1991) Given two maximization problems Π and Π', if 1) there exists an L-reduction (with parameters a and b) from Π to Π', and 2) there is an α-approximation algorithm for Π', then there exists an $(a \cdot b \cdot (\alpha - 1) + 1)$-approximation algorithm for Π. In other words, if it is NP-hard to approximate Π within a factor of $(a \cdot b \cdot (\alpha - 1) + 1)$, there does not exist an α-approximation algorithm for Π', unless P = NP.*

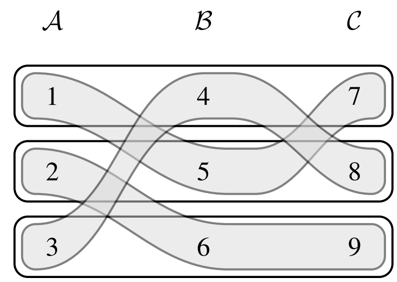

Figure 1: A $3DM^{=2}$ instance. Three disjoint sets in three columns, each containing $m = 3$ elements: $\mathcal{A} = \{1, 2, 3\}$, $\mathcal{B} = \{4, 5, 6\}$, $\mathcal{C} = \{7, 8, 9\}$. A set of $n = 6$ triples $\mathcal{T} = \{T_1, T_2, \ldots, T_6\}$, where: $T_1 = \{1, 4, 7\}$, $T_2 = \{2, 5, 8\}$, $T_3 = \{3, 6, 9\}$, $T_4 = \{1, 5, 7\}$, $T_5 = \{2, 6, 9\}$, and $T_6 = \{3, 4, 8\}$. Each element occurs in exactly two triples. The set $\mathcal{S} = \{T_1, T_2, T_3\}$ forms a matching of $\mathcal{T}$.

2.3 Maximum SAT and Maximum 3DM

We will perform some reductions from the *MAXE3SAT*, the maximum satisfiability problem where each clause has exactly three literals. An example (with $n = 4$ variables and $m = 5$ clauses) is $(x_1 \vee x_3 \vee \neg x_4) \wedge (\neg x_2 \vee \neg x_3 \vee \neg x_4) \wedge (x_1 \vee \neg x_2 \vee \neg x_4) \wedge (x_1 \vee x_2 \vee x_4) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_3)$. Of course, the example is satisfiable.

We have the following well-known result for *MAXE3SAT*.

Theorem 2. *(Johnson 1974; Yannakakis 1994; Håstad 2001) For any instance of MAXE3SAT, there exists a truth assignment that satisfies at least $\frac{7}{8}$ of the clauses. If there exists a $(\frac{7}{8} + \epsilon)$-approximation algorithm for the MAXE3SAT problem for any constant $\epsilon > 0$, then P = NP.*

Some reductions in the paper need to use the *NP*-complete 3-dimensional matching (*3DM*) problem (Karp 1972). Figure 1 presents an example of *3DM*.

Definition 11 (*3DM* problem)**.** *A 3DM problem instance is a tuple $\langle \mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{T} \rangle$ where $\mathcal{A} = \{a_1, a_2, \ldots, a_m\}$, $\mathcal{B} = \{b_1, b_2, \ldots, b_m\}$, and $\mathcal{C} = \{c_1, c_2, \ldots, c_m\}$ are three disjoint sets of cardinality m and $\mathcal{T}$ is a set of n triples $\mathcal{T} = \{T_1, T_2, \ldots, T_n\} \subseteq (\mathcal{A} \cup \mathcal{B} \cup \mathcal{C})$ such that each triple $T_i \in \mathcal{T}$ contains exactly one element from $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$, respectively. A partial matching of $\mathcal{T}$ is a subset $\mathcal{S} \subseteq \mathcal{T}$ of disjoint elements (i.e., $\forall\, T_i, T_j \in \mathcal{S},\ T_i \neq T_j \Rightarrow T_i \cap T_j = \emptyset$). The matching is complete if $|\mathcal{S}| = m$ or, equivalently, $\bigcup_{T \in \mathcal{S}} T = \mathcal{A} \cup \mathcal{B} \cup \mathcal{C}$. The 3DM problem asks the question: Does $\mathcal{T}$ contain a (complete) matching?*

Definition 12 ($3DM^{\leq 2}$ and $3DM^{=2}$)**.** *$3DM^{\leq 2}$ is a 3DM problem, in which no element occurs in more than two triples. In a $3DM^{=2}$ problem, each element occurs in exactly two triples.*

Theorem 3. *(Karp 1972; Garey and Johnson 1979) The 3DM problem is NP-complete, and remains so even if no element occurs in more than three triples. $3DM^{\leq 2}$ however, is poly-time solvable.*

MAX3DM is an optimal variant of*3DM*, defined below. $MAX3DM^{\leq 2}$ and $MAX3DM^{=2}$ can be defined similarly.

Definition 13 (*MAX3DM*). *Given an instance of 3DM, find $\mathcal{S}$, which is a maximum partial matching of $\mathcal{T}$, and $|\mathcal{S}| = k$. That is, among all partial matchings of $\mathcal{T}$, $\mathcal{S}$ has maximum set size.*

Theorem 4. *(Cygan 2013) For any instance of MAX3DM, there exists a polynomial-time $\frac{3}{4}$-approximation algorithm for the instance.*

Theorem 5. *(Chlebík and Chlebíková 2006) It is NP-hard to approximate MAX3DM within a factor of $\frac{94}{95}$, even for instances where each element has exactly two occurrences in the triples (thus the result is extended to MAX3DM$^{=2}$).*

In fact, Figure 1 presents an example of *3DM*$^{=2}$. We will use it as one running example to explain our L-reductions in Sections 3.2 and 3.3.

3 Inapproximability Results

In this section, we present a series of inapproximability results for optimization variants of *STRIPS* planning, established via approximation-preserving reductions from classical *NP*-hard optimization problems.

3.1 *MAX-PLANSAT*

We begin with a result due to Bylander, which establishes *NP*-completeness of some restricted planning problems.

Theorem 6. *(Bylander 1994, Theorem 3.5 and Corollary 3.6) PLANSAT$_+$ and PLANSAT$^1_{1+}$ are NP-complete.*

We will prove that *MAX-PLANSAT*$^1_{1+}$, an optimization version of *PLANSAT*$^1_{1+}$, where the goal is to maximize the number of goal conditions, admits no polynomial-time approximation algorithm with ratio exceeding a certain threshold, unless $P = NP$. We first propose a reduction from *MAXE3SAT* to *MAX-PLANSAT*$^1_{1+}$.

Reduction[1] Let I be a *MAXE3SAT* instance with n variables $\{x_1, x_2, \ldots, x_n\}$ and m clauses $\{c_1, c_2, \ldots, c_m\}$. Consequently, a *MAX-PLANSAT*$^1_{1+}$ instance I' is reduced, which has the following three types of conditions in $\mathcal{P}$: T_i: variable x_i is true; F_i: variable x_i is false; and C_j: clause c_j is satisfied. Hence, the size of $\mathcal{P}$ is $2n + m$.

For each variable x_i in I, two operators[2] are introduced in the reduced I': $\overline{F_i} \Rightarrow T_i$ and $\overline{T_i} \Rightarrow F_i$. That is, the condition T_i can be achieved from a state $\mathcal{S}$ if $F_i \notin \mathcal{S}$. Similarly, F_i can be achieved if $T_i \notin \mathcal{S}$. In this formulation, starting from an empty state $\emptyset$, it follows that for any variable x_i from the original instance I, at most one of T_i or F_i in the reduced instance I' can appear in any future state reachable from $\emptyset$. Totally, $2n$ operators introduced.

[1]The reduction closely follows Bylander's construction for proving the *NP*-hardness of *PLANSAT*$_+$ (Bylander 1994, Theorem 3.5). However, correctness must be independently established within our optimization framework.

[2]The symbol $\Rightarrow$ is used as a compact notation for *STRIPS* operators. For instance, $\overline{A} \Rightarrow B$ denotes an operator whose precondition is the absence of condition A in the current state, and whose effect is to add condition B to the current state. That is, if $A \notin \mathcal{S}$, then applying this operator to state $\mathcal{S}$ yields a successor state $\mathcal{S}' = (\mathcal{S} \cup \{B\})$.

Further, for each case where a clause c_j in I contains a variable x_i, we introduce in I' the operator $T_i \Rightarrow C_j$. For a negated variable $\overline{x_i}$ in c_j, the operator $F_i \Rightarrow C_j$ is introduced instead. Thus additionally $3m$ operators are introduced. Hence the size of $\mathcal{O}$ is $2n + 3m$.

The initial state is empty (i.e., $\mathcal{I} = \emptyset$) and the goal $\mathcal{G}$ is

$$\langle \mathcal{G}^+ = \{C_1, C_2, \cdots, C_m\}, \mathcal{G}^- = \emptyset \rangle.$$

Each operator introduced in the reduction has exactly one precondition and one positive postcondition. Thus, the reduced I' is an instance of *MAX-PLANSAT*$^1_{1+}$. We now establish that this reduction satisfies the following property.

Proposition 1. *Let I be a MAXE3SAT instance with m clauses and n variables. The instance has optimal solution OPT$(I) = k^\star$ (i.e., totally $k^\star$ clauses satisfied) iff the reduced MAX-PLANSAT$^1_{1+}$ instance I' has optimal solution of OPT$(I') = k^\star$ goal conditions from G satisfied.*

Proof. Suppose an assignment in I satisfies exactly $k^\star$ out of the m clauses. This assignment can be used to construct a plan for the corresponding *MAX-PLANSAT*$^1_{1+}$ instance I'. Starting from the initial state $\mathcal{I} = \emptyset$, the plan applies a sequence of operators that achieve exactly $k^\star$ of the m goal conditions from $\mathcal{G}_+$ of $\mathcal{G}$. Each of these goal conditions corresponds to a clause in the original I. Conversely, assume that the reduced instance I' has an optimal solution satisfying $k^{\star\star} > k^\star$ goal conditions. The sequence of applied operators corresponds to a variable assignment that satisfies $k^{\star\star}$ clauses in I, contradicting the assumption that $k^\star$ is optimal. Therefore, *OPT*(I') = *OPT*$(I) = k^\star$. ☐

Proposition 2. *From MAXE3SAT to MAX-PLANSAT$^1_{1+}$, there exists an L-reduction, where $a = b = 1$.*

Proof. The reduction introduced above, is indeed an L-reduction from *MAXE3SAT* to *MAX-PLANSAT*$^1_{1+}$, with parameters $a = 1$ and $b = 1$.

It is straightforward to show that all the three requirements on L-reduction (see Definition 10) are satisfied here: a) The construction involves a constant number of conditions and operators per variable and clause, and can be performed in polynomial time. b) From Proposition 1, we have *OPT*(I') = *OPT*(I). Thus, *OPT*$(I') \leq a \cdot$ *OPT*(I) holds with $a = 1$. c) Suppose we are given a solution to I' that achieves V' goals. From the structure of the reduction, the sequence of selected variable-assignment operators corresponds to a truth assignment over the variables in I. Since each goal C_j is only achievable if at least one literal in the corresponding clause is satisfied under that assignment, we have $V \geq V'$, where V is the number of satisfied clauses. That is, we have $|OPT(I) - V| = |OPT(I') - V| \leq b \cdot |OPT(I') - V'|$, where $b = 1$. ☐

Theorem 7. *There does not exist an α-approximation algorithm for MAX-PLANSAT$^1_{1+}$ for any $\alpha > \frac{7}{8}$, unless P = NP.*

Proof. From Proposition 2, we know that there exists an L-reduction from *MAXE3SAT* to *MAX-PLANSAT*$^1_{1+}$ with parameters $a = 1$ and $b = 1$. By Theorem 1, if there exists an α-approximation algorithm for *MAX-PLANSAT*$^1_{1+}$, then there exists an $(a \cdot b \cdot (\alpha - 1) + 1) = \alpha$-approximation algorithm for *MAXE3SAT*. However, it is also known that no such algorithm exists for *MAXE3SAT* when $\alpha > \frac{7}{8}$, unless *P* = *NP* (Theorem 2). Therefore, no α-approximation algorithm exists for *MAX-PLANSAT*$^1_{1+}$ for any constant $\alpha > \frac{7}{8}$, unless *P* = *NP*. □

3.2 Restricted *MAX-PLANSAT*: Case One

We know that *PLANSAT*$^+_1$, a planning problem which, as shown in the following theorem, is tractable:

Theorem 8. *(Bylander 1994, Theorem 3.7) The decision problem PLANSAT*$^+_1$ *can be solved in polynomial time.*

However, in this section, we will show that *MAX-PLANSAT*$^+_1$, which is an optimization variant of *PLANSAT*$^+_1$, is inapproximable, unless *P* = *NP*. We begin by presenting a reduction from *MAX3DM*$^{=2}$ to *MAX-PLANSAT*$^+_1$. To illustrate the reduction, we refer to the *3DM*$^{=2}$ instance depicted in Figure 1.

Reduction Given an instance I of *MAX3DM*$^{=2}$ problem with $3m$ elements and $n = 2m$ triples, we construct an instance I' of *MAX-PLANSAT*$^+_1$. In particular, for each triple $T \in \mathcal{T}$ in I, a condition τ is introduced in $\mathcal{P}$ of I'. Thus in the resulting I', $n = 2m$ conditions are introduced.

Initially all conditions are true. That is, for any condition $\tau \in \mathcal{P}$, $\tau \in \mathcal{I}$ holds. The goal $\mathcal{G}$ asks for negating all conditions. In the given example, this results in

$$\langle \mathcal{G}_+ = \emptyset, \quad \mathcal{G}_- = \{\tau_1, \tau_2, \tau_3, \tau_4, \tau_5, \tau_6\}\rangle.$$

For each condition τ in $\mathcal{G}_-$ (thus each triple T in I), an operator o_τ is introduced correspondingly in $\mathcal{O}$:

$$o_\tau: \ \tau_a \wedge \tau_b \wedge \tau_c \Rightarrow \overline{\tau},$$

where τ_a, τ_b, and τ_c, correspond to the three[3] triples T_a, T_b and T_c in I that share an element with the triple T, on Layers $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$, respectively. An application of operator o_τ in I' corresponds to a selection of a triple T in I, where all the triples (at most three of them) sharing an element with T, are not selected.

For example, an application of $\tau_5 \wedge \tau_4 \wedge \tau_6 \Rightarrow \overline{\tau_2}$ in I' corresponds to a selection of the triple $T_2 = \{2, 5, 8\}$ in I, provided that the triples $T_5 = \{\mathbf{2}, 6, 9\}$, $T_4 = \{1, \mathbf{5}, 7\}$, and $T_6 = \{3, 4, \mathbf{8}\}$ are not selected (actual elements shared with T_2 in these triples are highlighted in bold-font, respectively).

Proposition 3. *Let I be a MAX3DM$^{=2}$ instance with $3m$ elements and $n = 2m$ triples. The instance has optimal value OPT$(I) = k^\star$ (i.e., a matching of $k^\star$ triples) iff the reduced MAX-PLANSAT$^+_1$ instance I' has optimal value OPT$(I') = k^\star$ satisfied goal literals.*

Proof. Assume *OPT*$(I) = k^\star$, an optimal matching of size $k^\star$ in I, and *OPT*$(I') = k^{\star\star}$, a plan in I' satisfying optimally $k^{\star\star}$ goal literals. For each one of the selected $k^\star$ triples (say T), trigger the corresponding operator o_τ in I' that achieves the negative literal $\overline{\tau}$. No two chosen triples in the solution for I have any common element, which ensures that in I' the applicability of o_τ achieving $\overline{\tau}$. We have secured a plan in I' that satisfies $k^\star$ goal literals in $\mathcal{G}_-$, obtained through applying exactly $k^\star$ corresponding operators. We know $k^{\star\star} \geq k^\star$.

Meanwhile, note that $k^{\star\star} \leq m$ holds. That is, at most m triples can be selected in I. Thus, at most half of the goals can be achieved in I'. These $k^{\star\star}$ goal literals are achieved in I' through a sequence of applications of operators in $\mathcal{O}$, which corresponds a sequence of selection of triples in I. Along the way, none of the requirement that no two selected triples sharing one same element in I, is violated. Hence, from this sequence, we can construct a solution in I where $k^{\star\star}$ triples are selected. Hence, $k^{\star\star} \leq k^\star$. Combining both directions, *OPT*(I') = *OPT*$(I) = k^\star$ is concluded. □

[3]It is possible that only two relevant triples exist (say, T_a and T_b), where T_a shares two elements with T, and T_b shares the remaining element with T. In this case, we define the operator as $\tau_a \wedge \tau_b \Rightarrow \overline{\tau}$ instead. Since each element appears exactly two times in an instance of *MAX3DM*$^{=2}$, there can be at most three triples relevant (in the sense of element sharing) to a given triple T.

Proposition 4. *From MAX3DM$^{=2}$ to MAX-PLANSAT$^+_1$, there exists an L-reduction, where $a = b = 1$.*

Proof. The reduction above is an L-reduction, with parameters $a = 1$ and $b = 1$. The construction involves a constant number of conditions and operators per variable and triple, and can be performed in polynomial time. From Proposition 3, we have *OPT*(I') = *OPT*(I). Thus *OPT*$(I') \leq a \cdot$ *OPT*(I) holds, with $a = 1$. Suppose we are given a solution to I' that achieves V' goals. From the structure of the reduction, the sequence of applied operators corresponds to a sequence of selections of triples in I. Since each goal $\tau \in \mathcal{G}_-$ is only achievable if the corresponding triple is selected in I, we have $V \geq V'$, where V is the number of selected triples in I. Thus, we have $b = 1$, from the inequality of $|OPT(I) - V| \leq b \cdot |OPT(I') - V'|$. □

Theorem 9. *For any $\alpha > \frac{94}{95}$, an α-approximation algorithm for MAX-PLANSAT$^+_1$ does not exist, unless P = NP.*

Proof. From Theorem 5, we know that there does not exist a better than $\frac{94}{95}$-approximation algorithm for *MAX3DM*$^{=2}$. Now that we have an L-reduction here with $a = b = 1$, we can extend this bound to *MAX-PLANSAT*$^+_1$. By Theorem 5, there exists no polynomial-time approximation algorithm for *MAX3DM*$^{=2}$ with a ratio better than $\frac{94}{95}$. Since our reduction to *MAX-PLANSAT*$^+_1$ is an L-reduction with parameters $a = b = 1$, this inapproximability bound carries over, establishing that *MAX-PLANSAT*$^+_1$ also admits no $\frac{94}{95}$-approximation algorithm unless *P* = *NP*. □

Each operator in $\mathcal{O}$ has precondition size bounded 3. This allows us to obtain the following refined result.

Corollary 10. *For any $\alpha > \frac{94}{95}$, an α-approximation algorithm for MAX-PLANSAT$^{3+}_1$ does not exist, unless P = NP.*

3.3 Restricted *MAX-PLANSAT*: Case Two

We now consider another planning problem, *PLANSAT*^{0}, which is unconditional. That is, actions have no preconditions, but may include an arbitrary number of postconditions, either positive or negative. We know this problem is also tractable:

Theorem 11. *(Bylander 1994, Theorem 3.9) The decision problem PLANSAT^0 can be solved in polynomial time.*

In this section, we will show that *MAX-PLANSAT*^{0}, which is an optimization variant of *PLANSAT*^{0}, is also inapproximable, unless $P = NP$. We begin by presenting a reduction from *MAX3DM*$^{=2}$ to *MAX-PLANSAT*^{0}. The reduction is quite similar to the one presented in Section 3.2.

Reduction Given an instance I of *MAX3DM*$^{=2}$ problem with $3m$ elements and $n = 2m$ triples, we construct an instance I' of *MAX-PLANSAT*^{0}. The set $\mathcal{P}$ in I' is the same: For each triple $T \in \mathcal{T}$ in I, a condition τ is introduced. In the resulting I' of *MAX-PLANSAT*^{0}, thus $n = 2m$ conditions are introduced. Also, the goal $\mathcal{G}$ asks for negating all conditions. In the given example, this results in, again,

$$\langle \mathcal{G}_+ = \emptyset, \quad \mathcal{G}_- = \{\tau_1, \tau_2, \tau_3, \tau_4, \tau_5, \tau_6\}\rangle.$$

For each condition τ in $\mathcal{G}_-$ (thus each triple T in I), an operator o_τ is introduced correspondingly in $\mathcal{O}$:

$$o_\tau : \quad \Rightarrow \overline{\tau} \wedge \tau_a \wedge \tau_b \wedge \tau_c,$$

where τ_a, τ_b, and τ_c, correspond to the three[4] triples T_a, T_b and T_c, in the original problem instance I, that share an element with the triple T on Layers $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$, respectively. An application of such an operator will achieve $\overline{\tau}$, thus achieve a condition in $\mathcal{G}_-$ of $\mathcal{G}$. Meanwhile, three additional conditions (i.e., τ_a, τ_b, and τ_c), will be removed from $\mathcal{G}_-$ (if already there).

The operation corresponds to a selection of the triple T in I. Meanwhile, it requires that, all the triples (maximally three of them), who share an element with T, must not be selected. Or if already selected, they need to be de-selected.

For example, applying the operator $\Rightarrow \overline{\tau_2} \wedge \tau_5 \wedge \tau_4 \wedge \tau_6$ in I' corresponds to selecting the triple $T_2 = \{2, 5, 8\}$ in I. Meanwhile, triples $T_5 = \{\mathbf{2}, 6, 9\}$, $T_4 = \{1, \mathbf{5}, 7\}$, and $T_6 = \{3, 4, \mathbf{8}\}$ must not be selected (or removed if already selected).

Proposition 5. *Let I be a MAX3DM$^{=2}$ instance with $3m$ elements and $n = 2m$ triples. The instance has optimal value OPT$(I) = k^\star$ (i.e., a matching of $k^\star$ triples) iff the reduced MAX-PLANSAT^0 instance I' has optimal value OPT$(I') = k^\star$ satisfied goal literals.*

Proof. The proof is similar to the one for proving Proposition 3. In essence, an optimal solution *OPT*$(I) = k^\star$ ensures that *OPT*$(I') \geq k^\star$. And if we assume that *OPT*$(I') = k^{\star\star}$, $k^{\star\star} \leq k^\star$ holds, implying *OPT*$(I) =$ *OPT*(I'). □

It is thus evident that we have once again established an L-reduction. Moreover, using the convenient fact that $a = b = 1$, we obtain the same inapproximability bound with $\alpha = \frac{94}{95}$. We obtain the following results.

Proposition 6. *From MAX3DM$^{=2}$ to MAX-PLANSAT^0, there exists an L-reduction, where $a = b = 1$.*

Theorem 12. *For any $\alpha > \frac{94}{95}$, an α-approximation algorithm for MAX-PLANSAT^0 does not exist, unless P = NP.*

[4]It is again possible that only two relevant triples exist.

Each operator in $\mathcal{O}$ has at most four postconditions. This allows us to derive the following refined result.

Corollary 13. *For any $\alpha > \frac{94}{95}$, an α-approximation algorithm for MAX-PLANSAT$^0_4$ does not exist, unless P = NP.*

3.4 *MAX-PLANSAT*(K)

We now consider the bounded-length setting, where the plan length is restricted to at most K. To do so, we construct a reduction that satisfies the conditions of an L-reduction, thereby establishing the desired inapproximability result.

Reduction We propose in particular a reduction from *MAXE3SAT* to *MAX-PLANSAT*$(K)_{1+}$, where K is a bound on plan length. In the reduction, we will simply use the size of the condition set in the reduced problem to bound the plan length, i.e., $K = |\mathcal{P}|$.

Let I be a *MAXE3SAT* instance with n variables $\{x_1, x_2, \ldots, x_n\}$ and m clauses $\{c_1, c_2, \ldots, c_m\}$. A *MAX-PLANSAT*$(K)_{1+}$ instance I' is reduced, which has the following five types of conditions in $\mathcal{P}$ of I': T_i: variable x_i is true, F_i: variable x_i is false, V_i: a value has been assigned for x_i, *AllVars*: values of all variables have been assigned C_j: clause c_j is satisfied.

The size of $|\mathcal{P}|$ is thus $3n + 1 + m$. Accordingly we have $K = 3n + 1 + m$.

The initial state is empty (i.e., $\mathcal{I} = \emptyset$) and the goal $\mathcal{G}$ is

$$\langle \mathcal{G}^+ = \{C_1, C_2, \cdots, C_m\}, \mathcal{G}^- = \emptyset\rangle.$$

For each variable x_i, where $1 \leq i \leq n$, two operators for assigning either *T* or *F* value to it, provided the variable has not been assigned a value. As $\mathcal{I} = \emptyset$, it is impossible to assign a variable to both *T* and *F*.

$$\overline{V_i} \wedge \overline{F_i} \Rightarrow T_i \quad \text{and} \quad \overline{V_i} \wedge \overline{T_i} \Rightarrow F_i$$

When a variable x_i is assigned with a value (regardless true or false), the condition V_i is achieved to record the fact. That is, for all $1 \leq i \leq n$, $T_i \Rightarrow V_i$ and $F_i \Rightarrow V_i$. Note that once a V_i condition is achieved, it remains permanently true. In other words, once a variable is assigned a value, the assignment becomes irrevocable. The following operator is used to achieve the condition *AllVars*, corresponding to the fact in I that each variable has been assigned a value at least once: $\big(V_1 \wedge V_2 \wedge \cdots \wedge V_n\big) \Rightarrow$ *AllVars*.

In I, if a clause c_j contains a variable x_i (respectively $\overline{x_i}$), the following operators are introduced in the reduced I':

$$T_i \wedge \mathit{AllVars} \Rightarrow C_j \quad \text{and} \quad F_i \wedge \mathit{AllVars} \Rightarrow C_j.$$

We observe that the reduced instance belongs to the class of *MAX-PLANSAT*$(K)_{1+}$. That is, the number of preconditions is unbounded, while each operator has at most one positive postcondition. We now establish that this reduction satisfies the following property:

Proposition 7. *Let I be a MAXE3SAT instance with m clauses and n variables. The MAXE3SAT instance has optimal solution OPT$(I) = k^\star$ (i.e., a total of $k^\star$ clauses would be satisfied) iff the reduced MAX-PLANSAT$(K)_{1+}$ instance I' has optimal solution of OPT$(I') = k^\star$ goal conditions from $\mathcal{G}$ satisfied, in K steps, where $K = |\mathcal{P}|$, and $\mathcal{P}$ is the set of conditions in I'.*

Proof. Suppose that an optimal assignment for the instance I of *MAXE3SAT* satisfies exactly $OPT(I) = k^\star$ clauses. This assignment can be used to guide the application of operators in the reduced $MAX\text{-}PLANSAT(K)_{1+}$ instance I', starting from the initial state $\emptyset$, to satisfy exactly $k^\star$ of the m goal conditions, in $2n + 1 + k^\star$ steps.

More specifically, the first n steps assign truth values to the n variables using T_i or F_i operators. The next n steps are used to activate the corresponding V_i conditions. Then, a single step is used to achieve the *AllVars* condition, now that all V_i conditions are satisfied. After this, we can use the $k^\star$ steps to achieve exactly the $k^\star$ clause conditions, chosen according to the satisfied clauses in the original assignment *MAXE3SAT*. We know $2n + 1 + k^\star \leq 2n + 1 + m \leq K$. Hence, the plan is bounded by K. Suppose $OPT(I') = k^{\star\star}$, we have $k^{\star\star} \geq k^\star$.

Conversely, from $OPT(I') = k^{\star\star}$, we know that the reduced instance I' admits an optimal plan that satisfies $k^{\star\star}$ goal conditions within $K = 3n + 1 + m$ steps. Note that all goal conditions are clause conditions. Hence, for $k^{\star\star}$ to be greater than 0, at least one clause condition must be satisfied, which requires condition *AllVars* to be achieved first. Achieving *AllVars* necessarily requires setting all V_i conditions, which in turn requires all variable conditions to be achieved at least once. However, from the way the operators are designed, an assignment of a variable cannot be revoked. We have two observations here: a) At least $2n + 1$ steps are needed before any clause goal can be achieved. b) Right before a clause goal is achieved, all variable conditions are assigned *uniquely*, and the assignments cannot be revoked in the future steps. Next, we may use at most m steps to achieve clause goals. However, since the already achieved conditions (which cannot be revoked) correspond to a variable assignment satisfying at most $k^\star$ clauses, it follows that at most $k^\star$ goal conditions can be achieved. To this end, there are at least n steps remaining before reaching the plan length bound $K = 3n + 1 + m$. However, it is impossible for any operator to achieve any additional goal conditions beyond those already satisfied. We conclude that $k^{\star\star} \leq k^\star$. Hence, $OPT(I') = OPT(I) = k^\star$. □

As direct consequences from the reduction above, we derive the following two results.

Proposition 8. *There exists an L-reduction from MAXE3SAT to* $MAX\text{-}PLANSAT(K)_{1+}$*, where* $a = b = 1$.

Theorem 14. *For any* $\alpha > \frac{7}{8}$*, an* α*-approximation algorithm for* $MAX\text{-}PLANSAT(K)_{1+}$ *does not exist, unless P = NP.*

In the previous reduction, the only operator with more than two preconditions is the one used to achieve *AllVars*. However we can decompose this operator into a chain of n binary operators using n auxiliary conditions: *ZeroVar*, *OneVar*, *TwoVars*, ..., *NMinusOneVars*, which collectively lead to *AllVars*. The resulting sequence of n operators is: $ZeroVar \wedge V_1 \Rightarrow OneVar$, $OneVar \wedge V_2 \Rightarrow TwoVars$, ..., $NMinusOneVars \wedge V_n \Rightarrow AllVars$. All preconditions sizes are now bounded by 2. We obtain the following results.

Proposition 9. *There exists an* $a = b = 1$ *L-reduction from MAXE3SAT to* $MAX\text{-}PLANSAT(K)^2_{1+}$.

Theorem 15. *For any* $\alpha > \frac{7}{8}$*, an* α*-approximation algorithm for* $MAX\text{-}PLANSAT(K)^2_{1+}$ *does not exist, unless P = NP.*

In the following corollary, we establish *NP*-completeness for the corresponding decision variants, which are cases that were not explicitly settled in Bylander's original classification. These results serve as a baseline when contrasting the inapproximability bounds derived in this section.

Corollary 16. *The decision problems* $PLANSAT(K)_{1+}$ *and* $PLANSAT(K)^2_{1+}$ *are NP-complete.*

4 Conclusion

This paper provides, to the best of our knowledge, a first inapproximability study of *STRIPS* planning problems whose objective is to maximize the number of goal conditions achieved by a plan[5]. In short, our results show that when the objective is to satisfy as many goals as possible, no efficient algorithm in *STRIPS* planning can guarantee solutions close to optimal, unless *P* = *NP*. Consequently, our hardness thresholds not only delineate some exact boundaries that future approximation algorithms must respect, but also underscore the need for alternative strategies (e.g., parameterized, heuristic, or learning-guided methods) when pursuing practical performance improvements.

The inapproximability thresholds established in several of our theorems (Theorems 7, 14 and 15, in Sections 3.1 and 3.4) are tight: They admit no polynomial α-approximation for any α greater than $7/8$, mirroring exactly the celebrated bound for *MAXE3SAT* (Håstad 2001). In other words, any improvement beyond $7/8$ would immediately contradict the optimal lower bound for propositional satisfiability.

Another key contribution of this work is the demonstration of a sharp separation between decision and optimization complexities in classical planning, mirroring the well-known contrast between *2SAT* (which is in *P*) and its optimization counterpart *MAX-2SAT* (which is *NP*-hard). Specifically, we have established two such cases:

- In Section 3.2, we show that, although the decision fragments $PLANSAT_1^+$ (and by implication the more restrictive $PLANSAT_1^{3+}$) are solvable in polynomial time, their corresponding maximization variants ($MAX\text{-}PLANSAT_1^+$, or even $MAX\text{-}PLANSAT_1^{3+}$) are inapproximable (Theorem 9 and Corollary 10).
- In Section 3.3, we show that, although $PLANSAT^0$ (and by implication $PLANSAT_4^0$) are solvable in polynomial time, their maximization counterparts ($MAX\text{-}PLANSAT^0$, or even $MAX\text{-}PLANSAT_4^0$) are again inapproximable (Theorem 12 and Corollary 13).

Finally, we remark that these inapproximability bounds can be carried over, with minor adaptations expected, to other classical planning formalisms, such as PDDL (McDermott et al. 1998; Haslum et al. 2019).

[5]Inapproximability results in optimal multi-agent pathfinding (MAPF) are also obtained recently (Tan and Grastien 2025).

Acknowledgments

The authors thank all the reviewers for their valuable feedback. Xing Tan gratefully acknowledges the Discovery Grant from the Natural Sciences and Engineering Research Council of Canada (NSERC), and the research grant from the Faculty of Science & Environmental Studies, Lakehead University.

References

Allen, J. F. 1984. Towards a general theory of action and time. *Artificial intelligence*, 23(2): 123–154.

Bäckström, C.; Jonsson, P.; Ordyniak, S.; and Szeider, S. 2013. Parameterized complexity and kernel bounds for hard planning problems. In *International Conference on Algorithms and Complexity*, 13–24. Springer.

Bäckström, C.; and Nebel, B. 1995. Complexity results for SAS+ planning. *Computational Intelligence*, 11: 625–656.

Bercher, P.; Alford, R.; and Höller, D. 2019. A Survey on hierarchical planning – One abstract idea, many concrete realizations. In *Proc. of the 28th IJCAI*, 6267–6275.

Bercher, P.; Haslum, P.; and Muise, C. 2024. A Survey on plan optimization. In *Proc. of the 33rd IJCAI*, 7941–7950.

Blazewicz, J.; Pesch, E.; and Schmidt, G. 2019. *Handbook on scheduling*. Springer.

Blum, A. L.; and Furst, M. L. 1997. Fast planning through planning graph analysis. *Artificial intelligence*, 90(1-2): 281–300.

Bonet, B.; and Geffner, H. 1999. Planning as heuristic search: New results. In *European Conference on Planning*, 360–372. Springer.

Bonet, B.; and Geffner, H. 2000. Planning with incomplete information as heuristic search in belief space. In *Proc. of the 5th ICAPS*, 52–61.

Bylander, T. 1991. Complexity results for planning. In *Proc. of the 12th IJCAI*, 274–279.

Bylander, T. 1994. The computational complexity of propositional *STRIPS* planning. *Artificial Intelligence*, 69(1-2): 165–204.

Chlebík, M.; and Chlebíková, J. 2006. Complexity of approximating bounded variants of optimization problems. *Theoretical Computer Science*, 354(3): 320–338.

Cygan, M. 2013. Improved approximation for 3-Dimensional matching via bounded pathwidth local search. In *Proc. of the 54th FOCS*, 509–518.

Dean, T. L.; and Boddy, M. S. 1988. An analysis of time-dependent planning. In *Proc. of the 7th AAAI*, volume 88, 49–54.

Erol, K.; Hendler, J.; and Nau, D. S. 1994. HTN planning: complexity and expressivity. In *Proc. of the 12th AAAI*, 1123–1128.

Erol, K.; Hendler, J.; and Nau, D. S. 1996. Complexity results for HTN planning. *Annals of Mathematics and Artificial Intelligence*, 18(1): 69–93.

Fikes, R.; and Nilsson, N. J. 1993. *STRIPS*, a retrospective. *Artificial intelligence*, 59(1-2): 227–232.

Fikes, R. E.; and Nilsson, N. J. 1971. *STRIPS*: A new approach to the application of theorem proving to problem solving. *Artificial intelligence*, 2(3-4): 189–208.

Garey, M.; and Johnson, D. 1979. *Computers and intractability - a guide to NP-completeness*. Cambridge, MA, USA: San Francisco: W.H. Freeman and Company.

Haslum, P.; Lipovetzky, N.; Magazzeni, D.; and Muise, C. 2019. *An introduction to the Planning Domain Definition Language*, volume 13. Springer.

Helmert, M. 2006. The fast downward planning system. *JAIR*, 26: 191–246.

Hoffmann, J. 2001. FF: The fast-forward planning system. *AI magazine*, 22(3): 57–57.

Hoffmann, J.; and Nebel, B. 2001. The FF planning system: Fast plan generation through heuristic search. *JAIR*, 14: 253–302.

Håstad, J. 2001. Some optimal inapproximability results. *J. ACM*, 48(4): 798–859.

Johnson, D. S. 1974. Approximation algorithms for combinatorial problems. *Journal of Computer and System Sciences*, 9(3): 256–278.

Karp, R. M. 1972. Reducibility among combinatorial problems. In Miller, R. E.; Thatcher, J. W.; and Bohlinger, J. D., eds., *Complexity of Computer Computations*, 85–103.

Katz, M.; and Keyder, E. 2022. A* search and bound-sensitive heuristics for oversubscription planning. In *Proc. of the 36th AAAI*, 9813–9820.

Katz, M.; and Mirkis, V. 2016. In search of tractability for partial satisfaction planning. In *Proc. of the 25th IJCAI*, 3154–3160.

Kautz, H.; and Selman, B. 1996. Pushing the envelope: Planning, propositional logic, and stochastic search. In *Proc. of the 13th AAAI*, 1194–1201.

Lin, S.; Olz, C.; Helmert, M.; and Bercher, P. 2024. On the computational complexity of plan verification,(bounded) plan-optimality verification, and bounded plan existence. In *Proc. of the 38th AAAI*, 20203–20211.

McDermott, D.; Ghallab, M.; Howe, A.; Knoblock, C.; Ram, A.; Veloso, M.; Weld, D.; and Wilkins, D. 1998. PDDL — The Planning Domain Definition Language. Technical Report CVC TR-98-003, Yale Center for Computational Vision and Control. Technical Report, AIPS–1998 Planning Competition Committee.

McDermott, D. V. 1996. A Heuristic estimator for means-ends analysis in planning. In *AIPS*, volume 96, 142–149.

Muscettola, N.; Nayak, P. P.; Pell, B.; and Williams, B. C. 1998. Remote agent: To boldly go where no AI system has gone before. *Artificial intelligence*, 103(1-2): 5–47.

Nebel, B.; and Bäckström, C. 1994. On the computational complexity of temporal projection, planning, and plan validation. *Artificial Intelligence*, 66(1): 125–160.

Olaya, A. G.; de la Rosa, T.; and Borrajo, D. 2021. Selecting goals in oversubscription planning using relaxed plans. *Artificial Intelligence*, 291: 103414.

Papadimitriou, C. H.; and Yannakakis, M. 1991. Optimization, approximation, and complexity classes. *Journal of Computer and System Sciences*, 43(3): 425–440.

Pednault, E. P. 1989. ADL: Exploring the middle ground between STRIPS and the situation calculus. In *Proc. of the 1st KR*, 324–332.

Smith, D. E. 2004. Choosing objectives in over-subscription planning. In *Proc. of the 14th ICAPS*, 393–401.

Speck, D.; and Katz, M. 2021. Symbolic search for oversubscription planning. In *Proc. of 35th AAAI*, 11972–11980.

Tan, X.; and Grastien, A. 2025. Inapproximability of optimal multi-agent pathfinding problems. In *Proc. of the 39th AAAI*, 26707–26715.

Tate, A. 1976. *Project planning using a hierarchic nonlinear planner*. Department of Artificial Intelligence, University of Edinburgh.

Tate, A. 1977. Generating project networks. In *Proc. of the 5th IJCAI*, 888–893.

van den Briel, M.; Nigenda, R. S.; Do, M. B.; and Kambhampati, S. 2004. Effective approaches for partial satisfaction (over-subscription) Planning. In *Proc. of the 19th AAAI*, 562–569.

Wilkins, D. E. 2014. *Practical planning: extending the classical AI planning paradigm*. Elsevier.

Yannakakis, M. 1994. On the approximation of maximum satisfiability. *Journal of Algorithms*, 17(3): 475–502.