ELSEVIER

Artificial Intelligence 81 (1996) 241-271

Artificial Intelligence

A probabilistic analysis of propositional STRIPS planning ☆

Tom Bylander *

Division of Computer Science, The University of Texas at San Antonio, San Antonio, TX 78249, USA

Received May 1994; revised April 1995

Abstract

I present a probabilistic analysis of propositional STRIPS planning. The analysis considers two assumptions. One is that each possible precondition (likewise postcondition) of an operator is selected independently of other pre- and postconditions. The other is that each operator has a fixed number of preconditions (likewise postconditions). Under both assumptions, I derive bounds for when it is highly likely that a planning instance can be efficiently solved, either by finding a plan or proving that no plan exists. Roughly, if planning instances under either assumption have n propositions (ground atoms) and g goals, and the number of operators is less than an $O(n \ln g)$ bound, then a simple, efficient algorithm can prove that no plan exists for most instances. If the number of operators is greater than an $\Omega(n \ln g)$ bound, then a simple, efficient algorithm can find a plan for most instances. The two bounds differ by a factor that is exponential in the number of pre- and postconditions. A similar result holds for plan modification, i.e., solving a planning instance that is close to another planning instance with a known plan. Thus it appears that propositional STRIPS planning, a PSPACE-complete problem, exhibits a easy-hard-easy pattern as the number of available operators increases with a narrow range of hard problems. An empirical study demonstrates this pattern for particular parameter values. Because propositional STRIPS planning is PSPACE-complete, this extends previous phase transition analyses, which have focused on NP-complete problems. Also, the analysis shows that surprisingly simple algorithms can solve a large subset of the planning problem.

Keywords: Planning; STRIPS; Average-case analysis; PSPACE-complete

☆ This paper is a revised and extended version of [3].

* E-mail: bylander@ringer.cs.utsa.edu.

SSDI 0004-3702(95)00055-0

1. Introduction

Lately, there has been a number of worst-case complexity results for planning, showing that the general problem is hard and that several restrictions are needed to guarantee polynomial time [1,2,4,5,10,11,21]. A criticism of such worst-case analyses is that they do not apply to the average case [7,17]. Recent work in AI has shown that this criticism has some merit. Several experimental and theoretical results have shown that *specific* NP-complete problems are hard only for narrow ranges [6,9,18,19,26] and suggests that even the instances within these ranges can usually be efficiently solved [24,25].

This paper extends these results by providing a probabilistic analysis of propositional STRIPS planning [22]. In contrast to the above work, propositional STRIPS planning is a PSPACE-complete problem [4], a much harder complexity class [13]. PSPACE problems are those that can be solved by algorithms with space requirements that are bounded by a polynomial on the size of the input. PSPACE-complete problems are the hardest problems in this class. Satisfiability of quantified Boolean expressions belongs to this class. Naturally, because PSPACE-complete problems are harder than NP-complete problems, [1] they should also have regions of hard problems. See [13] for an introduction to NP- and PSPACE-completeness.

One way to think of the difference between PSPACE-complete and NP-complete problems is that NP-complete problems are restricted to a polynomial number of nondeterministic choice points, while PSPACE-complete problems can have in effect an exponential number of choice points. This is because nondeterministic algorithms restricted to polynomial space (NPSPACE algorithms) can be reduced to PSPACE algorithms, and these nondeterministic algorithms can require exponential time. In propositional STRIPS planning, this means that the shortest solution plan might have exponential length. Thus, each choice in a PSPACE-complete problem is potentially a much smaller component of a solution as compared to a NP-complete problem. Similarly, each modification to an instance of a PSPACE-complete problem might have a small effect. This might explain the smooth transitions between easy and hard instances in our empirical study.

In common with previous work on NP-complete problems, I make strong independence assumptions about the distribution of instances. I assume that the probability that a given operator is in a planning instance is independent of what other operators are in the instance. Two variations of this theme are explored. One is that each possible precondition (likewise postcondition) is selected independently of the other pre- and postconditions in the operator. The other is that each operator has a fixed number of preconditions (likewise postconditions). Under these assumptions, I derive bounds for when it is highly likely that specific algorithms will efficiently solve a planning instance, either by finding a plan or proving that no plan exists.

Musick and Russell [20] also analyze a problem similar to planning. They approximate a restricted kind of search (each operator has one postcondition) with Markov chains, which in turn leads to polynomial-time solutions on average under certain con-

[1] Actually, $P \neq PSPACE$ has not yet been proven, but I shall make the standard assumptions that $P \neq NP$ and $NP \neq PSPACE$.

ditions. In contrast, my analysis provides rigorous probabilistic bounds in terms of five parameters: the number of propositions, the number of operators, the number of pre- and postconditions per operator, and the number of goals. These parameters characterize a full range of planning problems.

The algorithms analyzed by this paper are all variations of simple hill climbing, i.e., hill climbing with no backtracking. Clearly, this type of search is far removed from the sophisticated partial planning algorithms that are the subject of much current study, e.g., TWEAK [5], SNLP [16], and UCPOP [23], and is impoverished compared to almost any other planning algorithm ever proposed. There are three reasons to consider algorithms of such simplicity: they permit formal analysis, they are efficient, and, as noted below, they cover a very large portion of the planning problem. The results for these algorithms should provide a baseline for analyzing and empirically comparing more sophisticated algorithms on random planning instances.

Specifically, given that randomly-generated planning instances have n propositions (ground atoms) and g goals, and that operators have r preconditions and s postconditions on average, I derive the following results. If the number of operators is at most

$$((2n-s)/s)(\ln g - \ln\ln 1/\delta),$$

then a simple, efficient algorithm can prove that no plan exists for at least $1-\delta$ of the instances. If the number of operators is at least

$$e^r e^{sg/n}(2n/s+1)(\ln g/\delta),$$

then a simple, efficient algorithm can find a plan for at least $1-\delta$ of the instances. If r, s, and δ are held constant as n and g increase, then the bounds are $O(n\ln g)$ and $\Omega(n\ln g)$, respectively. They differ by a factor that is exponential in the number of pre- and postconditions.

A similar result holds for plan modification. If the initial state or goals are different by one condition from that of another planning instance with a known plan, and if there are at least $e^{r+s}(2n/s)(\ln 1/\delta)$ operators, then it is likely $(1-\delta)$ that a single operator converts the old plan into a solution for the new instance.

Thus, it appears that propositional STRIPS planning is easy if the number of operators is below one threshold or above a somewhat higher threshold. Conjecturing that some range of problems between the thresholds are hard, then propositional STRIPS planning exhibits a easy-hard-easy pattern similar to NP-complete problems. An empirical study demonstrates this pattern for particular parameter values. However, the empirical study shows smooth transitions between easy and hard instances, and so would not normally be considered a phase transition. Despite this, the theoretical analysis can be said to demonstrate an "asymptotic" phase transition. Larger random planning instances are hard only if the number of operators is $\Theta(n\ln g)$. Outside this asymptote, larger instances become easy. In any case, future work is needed to narrow the gap between the bounds and to analyze more realistic distributional assumptions and more sophisticated algorithms.

The rest of the paper is organized as follows. First, definitions and key inequalities are presented. Then, the results of the analysis are derived. Finally, empirical results are displayed.

2. Preliminaries

This section defines propositional STRIPS planning, describes the distributions of instances to be analyzed, and presents key inequalities.

2.1. Propositional STRIPS planning

An instance of *propositional* STRIPS *planning* is specified by a tuple $\langle \mathcal{P}, \mathcal{O}, \mathcal{I}, \mathcal{G} \rangle$, where:

- $\mathcal{P}$ is a finite set of ground atomic formula, called the *propositions*; a proposition is also called a *positive condition*; its negation is called a *negative condition*; a *state* is a satisfiable set of conditions of which each proposition or its negation is a member;
- $\mathcal{O}$ is a finite set of *operators*; the preconditions and postconditions of each operator are satisfiable sets of conditions;
- $\mathcal{I}$ is the *initial state*; and
- $\mathcal{G}$, the *goals*, is a satisfiable set of conditions.

If the preconditions of an operator are satisfied by a state, then the operator can be applied to that state, and the resulting state is determined by adding the postconditions, deleting those conditions that conflict with the postconditions (cf. [12]). A solution plan is a sequence of operators that transforms the initial state into a goal state, i.e., a state that satisfies the goals.

For example, a blocks-world instance can be represented using propositions like $clear(A)$ to represent "block A has a clear top", and $on(A,B)$ to represent "block A is on top of block B". The set of preconditions of an operator to move A from on top of B to on top of C can be represented as:

$$\{clear(A),\ clear(C),\ on(A,B)\}.$$

That is, blocks A and C are clear, and block A is on top of block B. Its postconditions are:

$$\{clear(B),\ \neg clear(C),\ \neg on(A,B),\ on(A,C)\}.$$

If the preconditions are true before the operator is applied, then after the operator is applied, block B becomes clear, block C is no longer clear, block A is no longer on block B, and block A is now on block C.

2.2. Distributional assumptions

Let n be the number of propositions. Let o be the number of operators. Let r and s respectively be the expected number of pre- and postconditions within an operator. Let g be the number of goals.

For given n, o, r, s, and g, I assume that *random planning instances* under the *variable model* are distributed by generating each operator as follows:

- For each proposition $p \in \mathcal{P}$, p is a precondition of the operator with probability $r/(2n)$; alternatively $\neg p$ is a precondition with probability $r/(2n)$. These probabilities are independent of other pre- and postconditions. For postconditions, $s/(2n)$ is the relevant probability.

- For each proposition $p \in \mathcal{P}$, $p \in \mathcal{I}$ (the initial state) is as likely as $\neg p \in \mathcal{I}$.

For the goals, g propositions are selected at random and are set to positive or negative so that no goal is satisfied in the initial state. This latter restriction is made for ease of exposition.

The only difference between the variable model and the *fixed model* is that:

- Each operator has exactly r preconditions and s postconditions. Any legal set of r preconditions or s postconditions is equally likely.

It must be admitted that these assumptions do not approximate certain aspects of planning domains very well. For example, there are only b *clear* conditions for a blocks-world instance of b blocks compared to $\Theta(b^2)$ *on* conditions. However, every blocks-world operator refers to one or more *clear* conditions, i.e., a given *clear* condition appears more often within the set of ground operators than a given *on* condition. Also, there are correlations between the conditions, e.g., $clear(A)$ is more likely to appear with $on(A,B)$ than with $on(C,D)$. Similar violations can be found for any of the standard toy domains.

Ultimately, the usefulness of these assumptions will depend on how well the threshold bounds of the analysis classify easiness and hardness of real planning domains. Even so, it is worth noting that the fixed model provides a uniform distribution over the set of instances defined by the parameters. Because the results show that such planning instances are usually easy except for a narrow range of the number of operators o, it follows that the vast majority of planning instances are indeed easy.

2.3. *Algorithm characteristics*

Each algorithm in this paper is incomplete but sound, i.e., each algorithm returns correct answers when it returns "yes" or "no", but might answer "don't know". Specifically, "success" is returned if the algorithm finds a solution plan, "failure" is returned if the algorithm determines that no plan exists, and "don't know" is returned otherwise.

The performance of a given algorithm is characterized by an accuracy parameter δ, $0 < \delta < 1$. Each result below shows that if the number of operators o is greater than (or less than) a formula on n, r, s, g, and δ, then the accuracy of the algorithm on the corresponding distribution (see Section 2.2 on distributional assumptions) will be at least $1 - \delta$.

2.4. *Inequalities*

I freely use the following inequalities. For nonnegative x and y:

$$e^{-x/(1-x)} \leqslant 1 - x \quad \text{for } 0 \leqslant x < 1, \tag{1}$$

$$1 - x \leqslant e^{-x}, \tag{2}$$

$$\frac{xy}{1+xy} \leqslant 1 - (1-x)^y \quad \text{for } 0 \leqslant x < 1. \tag{3}$$

Inequalities (1) and (2) are easily derivable from [8]. Inequality (3) is derivable from inequalities (1) and (2). The logarithmic forms of these inequalities are sometimes used.

A particular form of Bonferroni's inequality shall be used. If $E_1, E_2, \ldots, E_m$ are m events and the probability of each event is greater than or equal to $1 - \delta/m$, then:

$$P(E_1 \wedge E_2 \wedge \cdots \wedge E_m) \geqslant 1 - \delta. \tag{4}$$

Finally, the following recurrence relation is useful in analyzing the fixed model. Its justification is described in the proof for Lemma 4. Let $f(s,n,k)$ be the probability that s conditions that are randomly generated from n propositions are consistent with some particular set of k conditions. Then, for nonnegative integers n, $s \leqslant n$ and $k \leqslant n$:

$$f(s,n,k) = \frac{n-k}{n} f(s-1,n-1,k) + \frac{k}{2n} f(s-1,n-1,k-1). \tag{5}$$

The base cases are $f(0,n,k) = 1$, $f(s,n,0) = 1$, $f(s,n,n) = 2^{-s}$ and $f(n,n,k) = 2^{-k}$. In the appendix, the following inequalities are demonstrated:

$$\mathrm{e}^{-sk/n} \leqslant f(s,n,k) \leqslant \frac{2n}{2n+sk}. \tag{6}$$

3. Efficiently proving plan non-existence

If there are few operators, it becomes unlikely that the postconditions of the operators cover all the goals, i.e., it is likely that some goal is not a postcondition of any operator. Recall that random planning instances are defined so that no goal is true of the initial state. So if some goal is not a postcondition of any operator, then the instance has no solution plan. This leads to the following simple algorithm:

```
POSTS-COVER-GOALS
for each goal
   if the goal is not in the postconditions of any operator
      then return failure
return don't know
```

While POSTS-COVER-GOALS might be considered a trivial algorithm, the following two theorems show that POSTS-COVER-GOALS works for a substantial range of the planning problem.

Theorem 1. *For random planning instances under the variable model, if*

$$o \leqslant \frac{2n-s}{s}(\ln g - \ln\ln 1/\delta),$$

then POSTS-COVER-GOALS *will determine that no plan exists for at least* $1-\delta$ *of the instances.*

Proof. The probability that there exists a goal that is not a postcondition of any operator can be developed as follows. Consider a particular goal and operator:

$s/2n$	probability that the goal is a postcondition of the operator;[2]
$1 - s/2n$	probability that the goal is not a postcondition of the operator;
$(1 - s/2n)^o$	probability that the goal is not a postcondition of any operator;
$1 - (1 - s/2n)^o$	probability that the goal is a postcondition of some operator;
$(1 - (1 - s/2n)^o)^g$	probability that every goal is a postcondition of some operator.

The inequality of the theorem implies that the above probability is less than or equal to δ. Suppose that the inequality of the theorem is true:

$$o \leqslant \frac{2n - s}{s}(\ln g - \ln\ln 1/\delta).$$

This is equivalent to:

$$\frac{os}{2n - s} \leqslant \ln \frac{g}{\ln 1/\delta}$$

and:

$$-\frac{os}{2n - s} \geqslant \ln \frac{\ln 1/\delta}{g}.$$

$\ln(1 - s/2n) \geqslant -s/(2n - s)$ by inequality (1), which implies:

$$o \ln(1 - s/2n) \geqslant \ln \frac{\ln 1/\delta}{g}.$$

This is equivalent to:

$$(1 - s/2n)^o \geqslant \frac{\ln 1/\delta}{g}$$

and:

$$-(1 - s/2n)^o \leqslant \frac{\ln \delta}{g}.$$

$\ln(1 - (1 - s/2n)^o) \leqslant -(1 - s/2n)^o$ by inequality (2), which implies:

$$\ln(1 - (1 - s/2n)^o) \leqslant \frac{\ln \delta}{g}.$$

This is equivalent to:

$$g \ln(1 - (1 - s/2n)^o) \leqslant \ln \delta$$

[2] For arithmetic expressions within this paper, multiplication has highest precedence, followed by division, logarithm, subtraction, and addition. E.g., $1 - s/2n$ is equivalent to $1 - (s/(2n))$.

and finally:

$$(1-(1-s/2n)^{o})^{g} \leqslant \delta,$$

which is the desired inequality.

Thus, if the inequality of the theorem is satisfied, then the probability that some goal is not a postcondition of any operator is at least $1-\delta$. □

Theorem 2. *For random planning instances under the fixed model, if*

$$o \leqslant \frac{2n-s}{s}(\ln g - \ln\ln 1/\delta),$$

then POSTS-COVER-GOALS *will determine that no plan exists for at least* $1-\delta$ *of the instances.*

Proof. The derivation in the previous proof holds for the fixed model to the point where $1-(1-s/2n)^{o}$ is the probability that a particular goal is a postcondition of some operator. However, if this goal is a postcondition of some operator, then this reduces the probability that other goals will be postconditions of that operator, i.e., the number of "available" postconditions is reduced from s to $s-1$. Although $(1-(1-s/2n)^{o})^{g}$ is not the probability that every goal is a postcondition of some operator, this expression does remain an upper bound. Thus, the same inequality holds for the fixed model. □

For fixed δ and increasing n and g, the above bound approaches $(2n-s)(\ln g)/s$. If s is also fixed, the bound is $O(n \ln g)$. In general then, planning instances with a number of operators linear in n (or linear in n times logarithmic in g times a small constant) will not have plans. Fortunately though, it is usually easy in such cases to prove that a plan does not exist.

Naturally, more complex properties that are efficient to evaluate and imply plan non-existence could and should be used, e.g., the above algorithm does not look at preconditions or consider how postconditions conflict with the goals. Nevertheless, the analysis of POSTS-COVER-GOALS provides a strong bound on when it is easy to prove plan non-existence.

4. Efficiently finding plans

With a sufficient number of operators, it becomes likely that some operator will make progress towards the goal. In this section, I consider four algorithms. One is a simple forward search from the initial state to a goal state, at each state searching for an operator that decreases the number of goals to be achieved. The second is a backward search from the goals to the initial state. The third is also a backward search from the goals, but tries finds a plan that will reach the goals from any initial state. The fourth is a very simple algorithm for when the initial state and goals differ by just one condition.

To illustrate the algorithms the following instance is used:

$$
\begin{aligned}
\mathcal{P} &= \{\ a_1, a_2, a_3, a_4\},\\
\mathcal{O} &= \{\ a_1 \wedge a_2 \Rightarrow \neg a_3 \wedge a_4,\\
&\qquad a_2 \wedge a_4 \Rightarrow a_3,\\
&\qquad \neg a_1 \wedge a_2 \Rightarrow a_3 \wedge a_4,\\
&\qquad a_2 \wedge \neg a_4 \Rightarrow \neg a_1,\\
&\qquad \neg a_2 \Rightarrow a_3 \wedge a_4,\\
&\qquad \Rightarrow \neg a_2\},\\
\mathcal{I} &= \{\ a_1, a_2, \neg a_3, \neg a_4\},\\
\mathcal{G} &= \{\ a_3, a_4\}.
\end{aligned}
$$

The notation $pre \Rightarrow post$ is used for operators; the preconditions are represented by a conjunction of conditions before the arrow; the postconditions are after the arrow. This instance would be possible under the variable model for $n = 4$, $o = 6$, $g = 2$, and r and s set to any positive number between 1 and 3, inclusive, though it would be an especially unlikely instance for $r = 3$ or $s = 3$.

4.1. Forward search

Consider the following algorithm:

```
PLAN-FORWARD(S)
if G is satisfied by S
then return success
else if some operator can be applied to S and satisfies more goals
    then let S' be the result of applying the operator to S
      return PLAN-FORWARD(S')
    else return don't know
    end if
end if
```

If PLAN-FORWARD($\mathcal{I}$) is called, then it searches for an operator that increases the number of satisfied goals. If there is such an operator, the current state S is updated. PLAN-FORWARD succeeds if it reaches a goal state and is noncommittal otherwise.

For the example instance, a plan of the first two operators might be generated. The first operator achieves the goal a_4 from the initial state, leaving the other propositions unchanged. The second operator achieves the remaining goal a_3.

PLAN-FORWARD just performs simple hill climbing. I do not claim that this is a practical algorithm for planning in general, but the analysis is greatly simplified by avoiding backtracking and partial plans. The probability that the algorithm will succeed can be bound by considering the probability that an additional goal can be satisfied by some operator. Certainly, a more systematic search algorithm that efficiently includes PLAN-FORWARD would exceed its probability of success.[3]

[3] One such algorithm would be A* search from the initial state using a heuristic equal to the number of unsatisfied goals times any constant greater than 1. The constraint on the constant ensures that the number of goals achieved are considered more valuable than the number of operators applied. However, this heuristic might not lead to optimal plans.

Despite its handicaps, PLAN-FORWARD is surprisingly robust under certain conditions. First, I demonstrate two lemmas for the number of operators that need to be considered to increase the number of satisfied goals. One lemma is for the variable model, and the other for the fixed model.

Lemma 3. *Consider random planning instances under the variable model except that d of the g goals are not satisfied. If*

$$o \geqslant e^r e^{s(g-d)/n} \left(\frac{2n}{sd} + 1\right) \ln \frac{1}{\delta},$$

then, for at least $1-\delta$ of the instances, applying some operator will increase the number of satisfied goals.

Proof. The expression for the probability that some operator will increase the number of satisfied goals can be developed as follows:

$(1 - r/2n)^n$	probability that a state satisfies the preconditions of an operator, i.e., each of n propositions is not a precondition with probability $1 - r/n$; alternatively, a proposition is a matching precondition with probability $r/2n$;
$(1 - s/2n)^{g-d}$	probability that the postconditions of an operator are consistent with the $g - d$ goals already achieved;
$(1 - s/2n)^d$	probability that the postconditions do not achieve any of the d remaining goals, i.e., for each goal, it is not a postcondition with probability $1 - s/2n$;
$1 - (1 - s/2n)^d$	probability that the postconditions achieve at least one of the d remaining goals.

Thus, the probability p that a particular operator can be applied, will not clobber any satisfied goals, and will achieve at least one more goal, is:

$$p = \left(1 - \frac{r}{2n}\right)^n \left(1 - \frac{s}{2n}\right)^{g-d} \left(1 - \left(1 - \frac{s}{2n}\right)^d\right).$$

$1 - p$ is the probability that the operator is missing one or more of these properties, and $(1 - p)^o$ is the probability that o operators are unsatisfactory.

If $(1-p)^o \leqslant \delta$, then there will be some satisfactory operator with probability at least $1 - \delta$. This inequality is satisfied if $o \geqslant (1/p)(\ln 1/\delta)$ because in such a case:

$$(1 - p)^o \leqslant e^{-po} \leqslant e^{-\ln 1/\delta} = \delta.$$

All that remains then is to determine an upper bound on $1/p$, i.e., a lower bound on p. For each term of p:

$$(1 - r/2n)^n \geqslant e^{-rn/(2n-r)} \geqslant e^{-r},$$

$$(1 - s/2n)^{g-d} \geqslant e^{-s(g-d)/(2n-s)} \geqslant e^{-s(g-d)/n},$$

$$1-(1-s/2n)^d \geqslant \frac{sd}{2n+sd}.$$

Inverting these terms leads to the bound of the lemma. □

Lemma 4. *Consider random planning instances under the fixed model except that d of the g goals are not satisfied. If at least*

$$e^r e^{s(g-d)/n}\left(\frac{2n}{sd}+1\right)\ln\frac{1}{\delta}$$

operators are considered, then, for at least $1-\delta$ of the instances, PLAN-FORWARD *will find an operator that increases the number of satisfied goals.*

Proof. Just as in the previous lemma, the approach is to show that the probability p that a particular operator can be applied, will not clobber any satisfied goals, and will achieve at least one more goal, satisfies:

$$p \geqslant e^{-r}e^{-s(g-d)/n}\frac{sd}{2n+sd}.$$

The probability that a particular operator can be applied is $2^{-r} \geqslant e^{-r}$.

The probability that the postconditions of an operator are consistent with the $g-d$ goals already achieved can be described with a recurrence equation. Let $f(s,n,k)$ be the probability that s conditions can be randomly generated from n propositions so that they are consistent with some particular set of k conditions. If a condition is randomly generated from the n propositions, there is a $(n-k)/n$ probability that it is neither identical to nor the negation of one of the k conditions; this leaves $s-1$ conditions to be generated from $n-1$ propositions and to be consistent with k conditions. Alternatively, there is a $k/2n$ probability that it is identical to one of the k conditions; this leaves $s-1$ conditions to be generated from $n-1$ propositions to be consistent with $k-1$ conditions. The remaining $k/2n$ probability is when it conflicts with one of the k conditions. This leads to the following recurrence equation:

$$f(s,n,k)=\frac{n-k}{n}f(s-1,n-1,k)+\frac{k}{2n}f(s-1,n-1,k-1),$$

which was introduced as Eq. (5).

In the base cases, $f(0,n,k)=1$, $f(s,n,0)=1$ (the probability is 1 if there are no conditions to select or no conditions to be consistent with), $f(s,n,n)=2^{-s}$ (each condition to select must have a particular sign), and $f(n,n,k)=2^{-k}$ (each condition to be consistent with must be selected). For this recurrence equation and these base cases, inequality (6) holds, in this case, $f(s,n,g-d) \geqslant e^{-s(g-d)/n}$.

The probability of achieving at least one more of the d remaining goals is one minus the probability that none of the d goals are achieved, i.e., $1-f(s,n,d)$. Inequality (6) implies that $f(s,n,d) \leqslant 2n/(2n+sd)$, from which $1-f(s,n,d) \geqslant sd/(2n+sd)$ follows.

The probability that the postconditions of an operator are consistent with the $g-d$ goals already achieved is not independent of the probability of achieving at least one more of the d remaining goals. Fortunately, the two events have an additive dependency relationship, i.e., either event "increases" the probability of the other event. For example, if the postconditions of an operator are consistent with the goals already achieved, then this increases the probability of achieving one of the remaining goals because there are fewer conditions to choose from, and because any of the goals can still be chosen.

This completes the proof. □

The maximum value of the expression in the lemmas can be used to describe PLAN-FORWARD, which leads to the following theorem:

Theorem 5. *For random planning instances under either the variable or the fixed model, if*

$$o \geqslant e^r e^{sg/n} \left(\frac{2n}{s} + 1\right) \ln \frac{g}{\delta},$$

then PLAN-FORWARD *will find a plan for at least* $1 - \delta$ *of the instances.*

Proof. For g goals, the number of satisfied goals will be increased at most g times. If each increase occurs with probability at least $1 - \delta/g$, then g increases (the most possible) will occur with probability at least $1 - \delta$ (this follows from Bonferroni's inequality).

Thus, Lemmas 3 and 4 can be applied using δ/g instead of δ. Maximizing over the g goals leads to:

$$\max_{d=1}^{g} \left[e^r e^{s(g-d)/n} \left(\frac{2n}{sd} + 1\right) \ln \frac{g}{\delta}\right] \leqslant e^r e^{sg/n} \left(\frac{2n}{s} + 1\right) \ln \frac{g}{\delta}. \quad \square$$

The bound is exponential in the expected numbers of pre- and postconditions. Naturally, as operators have more preconditions, it becomes exponentially less likely that they can be applied. Similarly, as operators have more postconditions, it becomes exponentially less likely that the postconditions are consistent with the goals already achieved. Note though that if $g \leqslant n/s$, then $e^{sg/n} \leqslant e$, so the expected number of postconditions s is not as important a factor if the number of goals is small.

For fixed δ, r, and s, and increasing n and g, the above bound is $\Omega(n \ln g)$. Taking into account the result for POSTS-COVER-GOALS, it is clear that two sides of the random planning problem are easy. Below an $O(n \ln g)$ bound on the number of operators, it is usually easy to prove that a plan does not exist; above an $\Omega(n \ln g)$ bound on the number of operators, it is usually easy to find a plan. Remaining is a gap of a constant between the two bounds, which is exponential in the number of pre- and postconditions. It is a safe conjecture that some range of instances within the gap is hard, so I conclude that random planning instances exhibit the easy-hard-easy pattern of other NP-hard problems, with the hard problems occupying a narrow range of the number of operators. The empirical study in a following section displays the results of using this algorithm.

4.2. Backward search

One could also perform a backward search from the goals to the initial state. Consider the following algorithm:

PLAN-BACKWARD(G)
if G is consistent with $\mathcal{I}$,
then return success
else if there is a an operator with R and S
as its pre- and postconditions such that
G is consistent with S,
R is consistent with $G - S$, and
$|G - \mathcal{I}| > |((G - S) + R) - \mathcal{I}|$
then return PLAN-BACKWARD($(G - S) + R$)
else return don't know

Initially, PLAN-BACKWARD($\mathcal{G}$) is invoked. PLAN-BACKWARD then chooses an operator if it can achieve a goal state from another state that has fewer conflicts with the initial state $\mathcal{I}$. The postconditions S of such an operator must be consistent with the current set of goals G, its preconditions R must be consistent with the goals not achieved by the postconditions, and the new set of goals $(G - S) + R$ must have fewer conditions that are not in the initial state.

In the example instance, the third operator $\neg a_1 \wedge a_2 \Rightarrow a_3 \wedge a_4$ achieves the goals $\{a_3, a_4\}$, leaving $\{\neg a_1, a_2\}$ as the new goals. a_2 is consistent with the initial state, so this reduces the number of unsatisfied goals by one. Now the fourth operator $a_2 \wedge \neg a_4 \Rightarrow \neg a_1$ achieves the unsatisfied goal $\neg a_1$, and the new set of goals will be $\{a_2, \neg a_4\}$, which is consistent with the initial state.

The disadvantage of PLAN-BACKWARD is that the number of current goals G can increase steadily to the number of propositions n because each new set of goals $(G - S) + R$ can be $r - 1$ larger than the previous set of goals (under the fixed model). First, I present an analysis for the general case, then I consider two special cases in which the performance of PLAN-BACKWARD will be more comparable to PLAN-FORWARD.

Lemma 6. *Consider random planning instances under the variable or fixed model except that d goals are not satisfied. If*

$$o \geqslant \mathrm{e}^{r+s}\left(\frac{2n}{sd} + 1\right)\ln\frac{1}{\delta},$$

then, for at least $1 - \delta$ of the instances, some operator will achieve a goal state from another state that has fewer conflicts with the initial state.

Proof. The expression for the probability for the variable model can be developed as follows:

$(1-r/2n)^n$	a lower bound on the probability that the preconditions of an operator are consistent with the goals minus the postconditions, and all preconditions not in the goals minus the postconditions are consistent with the initial state;
$1-(1-s/2n)^d$	probability that the postconditions achieve at least one of the d remaining goals;
$(1-s/2n)^n$	a lower bound on the probability that the postconditions of an operator are consistent with the goals, i.e., each of at most n goals is not negated with probability $1-s/2n$.

The last two probabilities are not independent, but their interaction is additive, i.e., an operator that is consistent with the remaining goals is more likely to achieve one of the remaining goals. Thus, the probability p that some operator will reduce the number of goals that are not true of the initial state is bounded by

$$p \geqslant (1-r/2n)^n(1-s/2n)^n(1-(1-s/2n)^d).$$

$1-p$ is the probability that the operator is unsatisfactory, and $(1-p)^o$ is the probability that o operators are unsatisfactory. If $o \geqslant (1/p)(\ln 1/\delta)$, then there will be some satisfactory operator with probability at least $1-\delta$.

For each term of p:

$$(1-r/2n)^n \geqslant e^{-rn/(2n-r)} \geqslant e^{-r},$$
$$(1-s/2n)^n \geqslant e^{-sn/(2n-s)} \geqslant e^{-s},$$
$$1-(1-s/2n)^d \geqslant \frac{sd}{2n+sd}.$$

Inverting these terms leads to the bound under the variable model.

Under the fixed model, let f be recurrence equation (5). Then, the probability of suitably consistent pre- and postconditions is at least $f(r,n,n) = 2^{-r} \geqslant e^{-r}$ and $f(s,n,n) = 2^{-s} \geqslant e^{-s}$, respectively. The probability that the postconditions achieve at least one of the d remaining goals is $1-f(s,n,d) \geqslant sd/(2n+sd)$. The probability that the postconditions are consistent with the goals is not independent of the probability that the postconditions achieve at least one of the d remaining goals, but their interaction is additive. Thus, the probability p that some operator reduces the number of goals not true of the initial state under the fixed model is also bounded by:

$$p \geqslant e^{-r}e^{-s}\frac{sd}{2n+sd},$$

which leads to the inequality of the theorem. □

As for PLAN-FORWARD, to determine a bound for PLAN-BACKWARD, the maximum value of the expression in the above lemma needs to be determined. This is done to prove the following theorem:

Theorem 7. *For random planning instances under either the variable or the fixed model, if*

$$o \geqslant e^{r+s}\left(\frac{2n}{s}+1\right)\ln\frac{g}{\delta},$$

then PLAN-BACKWARD *will find a plan for at least* $1-\delta$ *of the instances.*

Proof. For g goals, the number of unsatisfied goals will be decreased at most g times. If each decrease occurs with probability at least $1-\delta/g$, then g decreases (the most possible) will occur with probability at least $1-\delta$.

Thus, the expression in the previous lemma can be used substituting δ/g instead of δ. Maximizing over the g decreases leads to:

$$\max_{d=1}^{g}\left[e^{r+s}\left(1+\frac{2n}{sd}\right)\ln\frac{g}{\delta}\right] \leqslant e^{r+s}\left(1+\frac{2n}{s}\right)\ln\frac{g}{\delta}. \qquad \square$$

The difference between this bound for PLAN-BACKWARD and the bound for PLAN-FORWARD is that the PLAN-BACKWARD bound has an e^{s} term instead of $e^{sg/n}$. This makes the PLAN-BACKWARD bound larger by a factor of $e^{s(n-g)/n}$, so if the number of goals g is small relative to the number of propositions n, and if the expected number of postconditions s is large, then the increase is substantial.

This suggests that PLAN-FORWARD will outperform PLAN-BACKWARD when g is small relative to n. This performance difference should become more pronounced for larger s. However, if r, s, and δ remain constant as g and n increase, the order of the PLAN-BACKWARD bound will be $\Omega(n\ln g)$, which is identical to the order of the PLAN-FORWARD bound. Also, the following analysis suggests that the difference will be smaller than suggested by Theorem 7.

4.3. Backward search with few goals

The above analysis of PLAN-BACKWARD assumes the worst case regarding the size of the current set of goals G, namely that $|G|$ is always close to n. However, if the number of goals is small enough, specifically $g \leqslant n/r$, then $|G|$ can never reach n under the fixed model, and is very unlikely to reach n under the variable model. The analysis for the fixed model is much more tractable, and is given below.

Lemma 8. *Consider random planning instances under the fixed model except that d goals are not satisfied, and there are no more than $gr-dr+d \leqslant n$ goals. If*

$$o \geqslant e^{r}e^{s(gr-dr+d)/n}\left(\frac{2n}{sd}+1\right)\ln\frac{1}{\delta},$$

then, for at least $1-\delta$ *of the instances, some operator will achieve a goal state from another state that has fewer conflicts with the initial state.*

Proof. The only difference from the proof for Lemma 6 is that the postconditions must be consistent with at most $gr-dr+d$ goals rather than at most n goals. The probability of this is bounded by $f(s,n,gr-dr+d) \geqslant e^{-s(gr-dr+d))/n}$, where f is recurrence equation (5), and the inequality follows from inequality (6). Substituting $e^{-s(gr-dr+d)/n}$ for e^{-s} in the proof for Lemma 6 leads to the bound of this lemma. □

Theorem 9. *For random planning instances under the fixed model, with $gr \leqslant n$ and $r \geqslant 1$, if*

$$o \geqslant e^r e^{sgr/n}\left(\frac{2n}{s}+1\right)\ln\frac{g}{\delta},$$

then PLAN-BACKWARD *will find a plan for at least $1-\delta$ of the instances.*

Proof. For g goals, the number of unsatisfied goals will be decreased at most g times. If each decrease occurs with probability at least $1-\delta/g$, then g decreases (the most possible) will occur with probability at least $1-\delta$.

Also, satisfying a goal results in increasing the total number of goals by at most $r-1$. Thus, when there are d goals left to achieve, there are at most $g+(g-d)(r-1) = gr-dr+d$ goals.

Thus, the expression in the previous lemma can be used substituting δ/g instead of δ. Maximizing over the g decreases leads to:

$$\max_{d=1}^{g}\left[e^r e^{s(gr-dr+d)/n}\left(\frac{2n}{sd}+1\right)\ln\frac{g}{\delta}\right] \leqslant e^r e^{sgr/n}\left(\frac{2n}{s}+1\right)\ln\frac{g}{\delta}. \qquad \square$$

In the case where $r=1$, this bound for PLAN-BACKWARD is the same as for PLAN-FORWARD. For fixed r and s and increasing g and n, if sgr remains small relative to n, then there is little difference between the two bounds, but the PLAN-FORWARD bound will still be smaller for $r>1$, though it should be noted that both bounds are $\Omega(n \ln g)$.

4.4. Backward search independent of the initial state

Consider the following variation of PLAN-BACKWARD:

```
PLAN-BACKWARD2(G)
if G = ∅
then return success
else if there is a an operator with R and S
        as its pre- and postconditions such that
        G is consistent with S, and
        |(G − S) + R| < |G|
    then return PLAN-BACKWARD2((G − S) + R)
    else return don't know
```

Like the previous algorithm, PLAN-BACKWARD2 looks for operators that reduce the number of goals, but unlike PLAN-BACKWARD, PLAN-BACKWARD2 does not depend on the initial state, and it repeatedly looks for an operator that reduces the number of goals until there are no goals left. If PLAN-BACKWARD2 succeeds, then it will have discovered a sequence of operators that achieves a goal state from any initial state, although note that the first operator in this sequence (the last operator selected by PLAN-BACKWARD) must not have any preconditions; otherwise $|(G - S) + R|$ would be nonzero. Having such an operator is probably unrealistic; it is impossible under the fixed model if $r \geqslant 1$. Nevertheless, the analysis below suggests that reducing a set of goals into a much smaller set of goals is often possible, which, of course, can then be followed by forward search or a more general backward search.

In the example instance, the fifth operator $\neg a_2 \Rightarrow a_3 \wedge a_4$ achieves the goals leaving the single new goal $\neg a_2$. The sixth operator $\Rightarrow \neg a_2$ achieves the new goal without any preconditions.

I first introduce a lemma for the number of operators needed to find one operator that reduces the number of goals.

Lemma 10. *For random planning instances under the variable model, with $r \leqslant n/2$ and $s \leqslant n/2$, if*

$$o \geqslant e^{2r} e^{sg/n} \left(\frac{3n}{sg} + 1 \right) \ln \frac{1}{\delta},$$

then, for $1 - \delta$ of the instances, some operator reduces the number of goals.

Proof. The preconditions should not refer to any condition that is not a goal or the negation of a non-goal. This has probability $(1 - r/n)^{n-g}$. It does not matter what the postconditions do to these conditions.

The preconditions and postconditions should be consistent with the g goals. This has probability

$$\left(1 - \frac{r}{2n}\right)^g \left(1 - \frac{s}{2n}\right)^g = \left(1 - \frac{r}{2n} - \frac{s}{2n} + \frac{rs}{4n^2}\right)^g.$$

However, the case in which every goal equal to a postcondition is also equal to a precondition must be avoided. The probability that this occurs for a given goal is a sum of the following:

$(1 - r/n)(1 - s/n)$	probability that the goal and its negation is not in the pre- and postconditions;
$(r/2n)(1 - s/n)$	probability that the goal is in the preconditions, but it (as well as its negation) is not in the postconditions;
$(r/2n)(s/2n)$	probability that the goal is in both the pre- and postconditions.

Using the sum of these probabilities, the probability of this case happening for all g goals is:

$$\left(1 - \frac{r}{2n} - \frac{s}{n} + \frac{3rs}{4n^2}\right)^g.$$

Thus, the probability p that a random operator will satisfy the stated requirements is:

$$p = (1 - r/n)^{n-g} \left(\left(1 - \frac{r}{2n} - \frac{s}{2n} + \frac{rs}{4n^2}\right)^g - \left(1 - \frac{r}{2n} - \frac{s}{n} + \frac{3rs}{4n^2}\right)^g \right).$$

Now recall from the previous proofs that if the number of operators considered exceeds $(1/p)(\ln 1/\delta)$, then the probability that some operator is satisfactory is at least $1-\delta$. What remains then is to demonstrate an upper bound on $1/p$ (lower bound on p).

For the first term of p,

$$(1-r/n)^{n-g} \geqslant \mathrm{e}^{-r(n-g)/(n-r)}.$$

Because $r \leqslant n/2$,

$$\mathrm{e}^{-r(n-g)/(n-r)} \geqslant \mathrm{e}^{-2r(n-g)/n}.$$

Regarding the second term of p:

$$\begin{aligned}
&\left(\left(1-\frac{r}{2n}-\frac{s}{2n}+\frac{rs}{4n^2}\right)^g-\left(1-\frac{r}{2n}-\frac{s}{n}+\frac{3rs}{4n^2}\right)^g\right)\\
&=\left(1-\frac{r}{2n}-\frac{s}{2n}+\frac{rs}{4n^2}\right)^g\left(1-\left(\frac{4n^2-2rn-4sn+3rs}{4n^2-2rn-2sn+rs}\right)^g\right)\\
&=\left(1-\frac{2rn+2sn-rs}{4n^2}\right)^g\left(1-\left(1-\frac{2sn-2rs}{4n^2-2rn-2sn+rs}\right)^g\right)\\
&\geqslant \exp\left\{-\frac{2rng+2sng-rsg}{4n^2-2rn-2sn+rs}\right\}\frac{2sng-2rsg}{4n^2-2rn-4sn+rs+2sng-2rsg}\\
&=\exp\left\{-\frac{(2n-s)rg+(2n-r)sg+rsg}{(2n-r)(2n-s)}\right\}\frac{2sg(n-r)}{(n-r)(2n+2sg-s)+n(2n-s)}.
\end{aligned}$$

The first term of this expression can be further simplified using $r \leqslant n/2$ and $s \leqslant n/2$:

$$\begin{aligned}
&\exp\left\{-\frac{(2n-s)rg+(2n-r)sg+rsg}{(2n-r)(2n-s)}\right\}\\
&=\exp\left\{-\frac{rg}{2n-r}-\frac{sg}{2n-s}-\frac{rsg}{(2n-r)(2n-s)}\right\}\\
&\geqslant \exp\left\{-\frac{rg}{n}-\frac{2sg}{3n}-\frac{4rsg}{9n^2}\right\}\\
&\geqslant \exp\left\{-\frac{rg}{n}-\frac{2sg}{3n}-\frac{2sg}{9n}\right\}\\
&\geqslant \mathrm{e}^{-rg/n}\mathrm{e}^{-sg/n}.
\end{aligned}$$

At this point, the following lower bound for p has been derived:

$$p \geqslant \mathrm{e}^{-2r(n-g)/n}\mathrm{e}^{-rg/n}\mathrm{e}^{-sg/n}\frac{2sg(n-r)}{(n-r)(2n+2sg-s)+n(2n-s)}.$$

This can be further simplified using $\mathrm{e}^{-2r(n-g)/n}\mathrm{e}^{-rg/n} \geqslant \mathrm{e}^{-2r}$.

Finally, an upper bound for $(1/p)(\ln 1/\delta)$ is derived:

$$\begin{aligned}\frac{1}{p}\ln\frac{1}{\delta} &\leqslant e^{2r}e^{sg/n}\left(\frac{2n+2sg-s}{2sg}+\frac{n(2n-s)}{2sg(n-r)}\right)\ln\frac{1}{\delta}\\ &\leqslant e^{2r}e^{sg/n}\left(\frac{n+sg}{sg}+\frac{2n}{sg}\right)\ln\frac{1}{\delta}\\ &= e^{2r}e^{sg/n}\frac{3n+sg}{sg}\ln\frac{1}{\delta},\end{aligned}$$

which proves the lemma. □

Similar to previous theorems, the maximum of this expression needs to be determined. This is done to prove the next theorem.

Theorem 11. *For random planning instances under the variable model, with $r \leqslant n/2$ and $s \leqslant n/2$, if*

$$o \geqslant e^{2r}\left(\frac{5n}{s}+\frac{3e^s}{s}+e^{sg/n}\right)\ln\frac{g}{\delta},$$

then PLAN-BACKWARD2 *will find a plan that achieves the goals from any initial state for at least $1-\delta$ of the instances.*

Proof. For g goals, the number of remaining goals will be decreased at most g times. If each decrease occurs with probability $1-\delta/g$, then g decreases (if necessary) will occur with probability at least $1-\delta$.

Thus, Lemma 10 can be applied using δ/g instead of δ. Maximizing over the g decreases leads to:

$$\max_{d=1}^{g}\left[e^{2r}e^{sd/n}\left(\frac{3n}{sd}+1\right)\ln\frac{g}{\delta}\right] \leqslant e^{2r}\ln\frac{g}{\delta}\left(e^{sg/n}+3\max_{d=1}^{g}\left[e^{sd/n}\frac{n}{sd}\right]\right),$$

$e^{sd/n}n/sd$ has one minimum for positive d, i.e., when $d=n/s$. So:

$$\max_{d=1}^{g}\left[e^{sd/n}\frac{n}{sd}\right] = \max\left\{e^{s/n}\frac{n}{s}, e^{sg/n}\frac{n}{sg}\right\} \leqslant \max\left\{e^{1/2}\frac{n}{s}, \frac{e^s}{s}\right\} \leqslant \frac{5n}{3s}+\frac{e^s}{s}.$$

As a result:

$$e^{2r}\ln\frac{g}{\delta}\left(e^{sg/n}+3\max_{d=1}^{g}\left[e^{sd/n}\frac{n}{sd}\right]\right) \leqslant e^{2r}\ln\frac{g}{\delta}\left(e^{sg/n}+\frac{5n}{s}+\frac{3e^s}{s}\right),$$

which proves the theorem. □

Comparing the two bounds for PLAN-FORWARD and PLAN-BACKWARD2, the bound for PLAN-BACKWARD2 is worse in that it has a larger constant and has an e^{2r} term as opposed to an e^r term for the PLAN-FORWARD bound. Because PLAN-BACKWARD2 does not use the initial state, some increase would be expected. However, the PLAN-BACKWARD2 bound is better in that one component is additive, i.e., $O(e^s+n/s)$; whereas the corresponding subexpression for the PLAN-FORWARD bound is $O(e^{sg/n}n/s)$. The reason is that in the maximization of the PLAN-BACKWARD2 bound, $e^{sd/n}$ is maximum when d is at its maximum, while the maximum value for n/sd is when d is at its minimum. In the maximization for the PLAN-FORWARD bound, $e^{s(g-d)/n}$ attains its maximum at the same time as n/sd does, when d is at its minimum. However, for fixed r, s, and δ, and increasing n and g, both bounds are $\Omega(n \ln g)$.

4.5. Plan modification

So far I have considered the problem of generating a plan from scratch. In many cases, however, the current planning instance is close to a previously solved instance, e.g., [14,15].

Consider a simplified version of plan modification, specifically, when the initial state or set of goals of the current planning instance differs by one condition from a previously solved instance. In this case, the new instance can be solved by showing how the new initial state can reach the old initial state, or how the old goal state can reach a new goal state. Within the framework of random planning instances then, I shall analyze the problem of reaching one state from another when the two states differ by one condition, i.e., there are n goals, and all but one goal is true of the initial state.

The worst-case complexity of this problem, like the problem of planning from scratch, is PSPACE-complete [21]. However, the following theorem shows that efficient plan modification does not appear to require as many operators as efficient planning from scratch.

Theorem 12. *For random planning instances under either the variable or the fixed model in which there are n goals, where $n-1$ goals are true of the initial state, if*

$$o \geqslant e^r e^s \frac{2n}{s} \ln \frac{1}{\delta},$$

then, for at least $1-\delta$ of the instances, some operator solves the instance in one step.

Proof. First, I develop the probability p that a random operator solves a random instance for the variable model. The probability that the preconditions are consistent with the initial state is $(1-r/2n)^n$. The probability that the postconditions are consistent with the $n-1$ goals already achieved is $(1-s/2n)^{n-1}$. In addition, the probability that the goal is achieved by a postcondition is $s/2n$. Thus:[4]

$$p = (1-r/2n)^n(1-s/2n)^{n-1}\frac{s}{2n}.$$

Lower bounds for p are:

$$p \geqslant e^{-rn/(2n-r)}e^{-sn/(2n-s)}\frac{s}{2n} \geqslant e^{-r}e^{-s}\frac{s}{2n}.$$

[4] This does not scale up to the case of attaining g goals by a single operator. The probability that the postconditions of a random operator contain the g goals is $(s/2n)^g$, i.e., exponentially small in the number of goals.

The probability that none of o operators solves the instance is $(1-p)^o$. If o satisfies the inequality stated in the theorem, then:

$$(1-p)^o \leqslant e^{-po} \leqslant e^{-\ln 1/\delta} = \delta,$$

which proves the bound for the variable model.

For the fixed model, the probability that the preconditions are true of the initial state is $2^{-r} \geqslant e^{-r}$. The probability that the goal is achieved by a postcondition is $s/2n$ (it is selected with probability s/n and has the right sign with probability $1/2$). The probability that the remaining $s-1$ postconditions are consistent with the $n-1$ goals already achieved is $f(s-1, n-1, n-1) \geqslant 2^{-s+1} \geqslant e^{-s}$, where f is defined by recurrence equation (5). Thus, the probability p that a random operator solves a random instance in the fixed model has a lower bound of $e^{-r}e^{-s}s/2n$, and the inequality of the theorem follows. □

Thus, for fixed r, s, and δ, $\Omega(n)$ operators suffice to solve planning instances that differ by one condition from previously solved instances. So, for at least the distributions of planning instances considered here, the number of operators needed for efficient plan modification appears to be a factor of $O(\ln g)$ lower than that needed for efficient planning from scratch.

As a corollary, consider any state sequence of length $g+1$ from the initial state to a goal state, where each successive state in the sequence differs by just one condition. If g/δ is substituted for δ in the theorem, and the inequality is satisfied, then any particular transition in the sequence can be accomplished with probability $1-\delta/g$. Consequently, all transitions can be performed with probability $1-\delta$. Thus, $\Omega(n \ln g)$ planning can be accomplished even if the transitions from the initial state to a goal state are chosen in advance.

5. Empirical study

The formal analysis provides rigorous probabilistic bounds on when random planning instances can be efficiently solved, either by proving that no plan exists or by finding a solution plan. However, the derivation of the bounds in the above theorems depends on rather crude inequalities. In this section, I display the empirical results when two of the above algorithms, POSTS-COVER-GOALS and PLAN-FORWARD, are applied to randomly-generated planning instances.

Note that there are five parameters to choose (the number of operators o, the number of propositions n, the number of pre- and postconditions r and s, and the number of goals g) as well as the choice of variable model (on average, operators have r and s pre- and postconditions) or fixed model (each operator has exactly r and s pre- and postconditions). As a result, it is not feasible to empirically cover many of the possibilities. Any choice of values will be arbitrary to some extent.

I chose the following values for this study: $n = 100$ and $n = 1000$, $r = 2$ and $s = 2$ under the fixed model, g ranging from low values up to n, and o varying over where the algorithms' performance changes. The two values for n allow g to vary over many

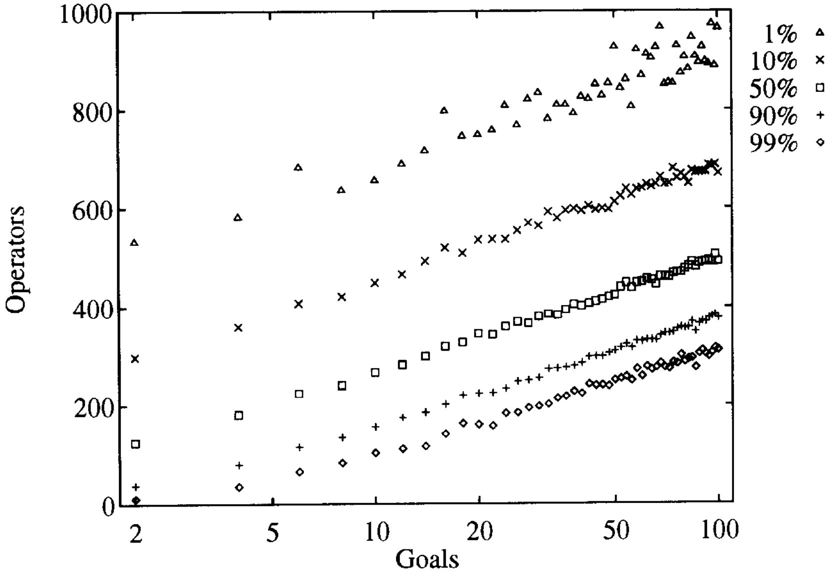

Fig. 1. Empirical effectiveness of POSTS-COVER-GOALS for 100 propositions. The x-axis is logarithmically scaled.

values, and also show how the transitions change from a lower value to a higher value. The values of r and s are the minimum values that make propositional STRIPS planning PSPACE-complete [4]. The fixed model ensures efficiency in generating an operator.[5] Different values for g will test the $\ln g$ asymptote.

For each trial, it is assumed that there is an unbounded stream of randomly-generated operators that can be used. For the POSTS-COVER-GOALS algorithm, it can be determined when the stream of operators covers all the goals. At this number of operators (call the number a), POSTS-COVER-GOALS fails to solve the problem. In my implementation of PLAN-FORWARD, whenever an additional goal is achieved, the algorithm reverts to the beginning of the stream, attempting to achieve remaining goals with previously generated operators. For this implementation, it can be determined how much of the stream was used by PLAN-FORWARD to solve the problem. At this number of operators (call this number b), PLAN-FORWARD solves the problem.

1000 trials were performed for each value of n and g considered. This will, as shown below, give a good indication of where these algorithms solve from 1% to 99% of the instances. I.e., if 99% of the a values are greater than 100, then POSTS-COVER-GOALS empirically solves at least 99% of the instances when $o \leqslant 100$. If 1% of the b values are less than or equal to 1000, then PLAN-FORWARD empirically solves at least 1% of the instances when $o \geqslant 1000$.

[5] It takes constant time to generate two random numbers. For the variable model, generating an operator would take time linear in n because each condition must be independently considered.

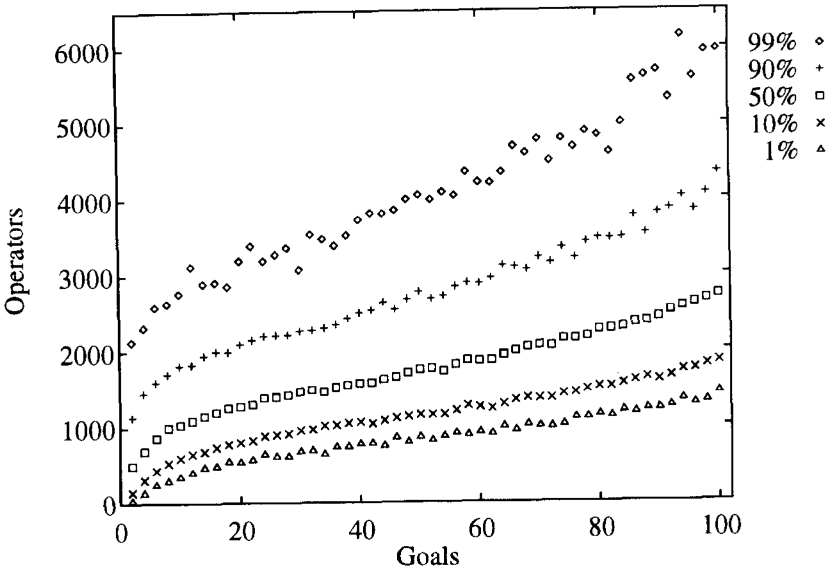

Fig. 2. Empirical effectiveness of PLAN-FORWARD for 100 propositions.

Fig. 1 displays the empirical results for POSTS-COVER-GOALS when $n = 100$. The x-axis is the number of goals (even numbers from 2 to 100) on a logarithmic scale, and the y-axis is the number of operators. Each point on the graph indicates the number of operators where POSTS-COVER-GOALS solves a certain percentage of instances for a given number of goals. The five kinds of points correspond to five isolevels. The diamond points indicate 99% effectiveness; the plus points 90% effectiveness; the square points 50%; the $\times$ points 10%; and the triangle points 1%. The POSTS-COVER-GOALS algorithm has higher effectiveness for lower number of operators. For a given level of effectiveness, the logarithmic scaling makes it clear that the number of operators varies logarithmically with g, which is consistent with the theoretically derived bound. Also, the theoretical bound is remarkably close to the empirical results, e.g., for $g = 100$ and $\delta = 0.01$, the theoretical bound gives 305 operators; the empirical result is 311. This closeness is also true for $n = 1000$.

Fig. 2 displays the results for PLAN-FORWARD when $n = 100$. The x-axis is displayed on a linear scale in this graph. In this case, PLAN-FORWARD has higher effectiveness for higher number of operators. In apparent contradiction to the analysis for PLAN-FORWARD, the effect of the number of goals appears to be linear. For example, the 99% effectiveness level appears to vary linearly from about 2000 for $g = 2$ to about 6000 for $g = 100$. The apparent contradiction can be resolved by looking closer at the theoretical bound of Theorem 5, i.e., $e^r e^{sg/n}(2n/s + 1)(\ln g/\delta)$, where for this data $r = s = 2$ and $n = 100$. As the number of goals g increases from 2 to 100, $\ln g/\delta$ with $\delta = 0.01$ only doubles, but note that $e^{sg/n}$ increases by more than a factor of 7, with most of the increase occurring for $g > 50$. This is more than sufficient to account for the three-fold increase observed for the 99% effectiveness level. Because $e^{sg/n}$ more than doubles when $g > n/2$ while $\ln g$ does not even increment by 1 (no matter what n is),

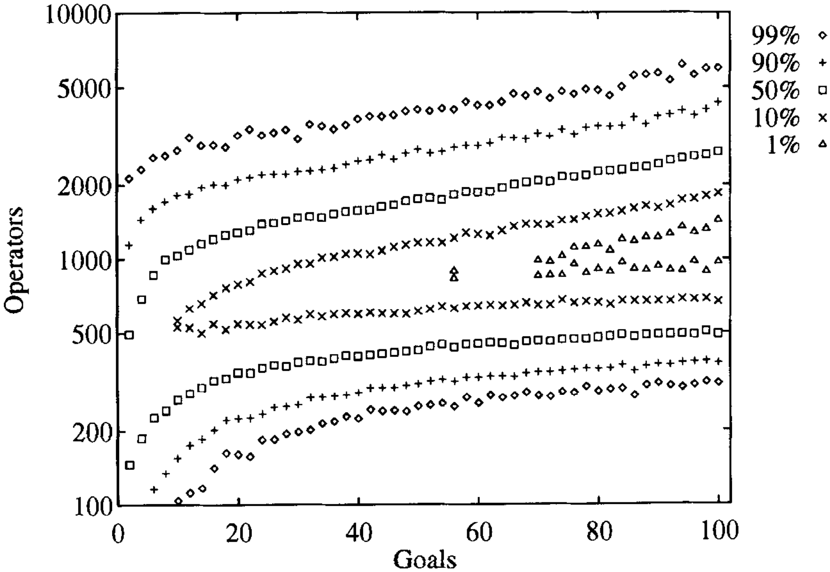

Fig. 3. Combined empirical effectiveness of POSTS-COVER-GOALS and PLAN-FORWARD for 100 propositions. The y-axis is logarithmically scaled.

a similar effect should be present for higher values of n. In this case, the theoretical bound (which is over 50,000 for $g = 100$ and $\delta = 0.01$) exceeds the empirical values by a very pessimistic margin. This is also true for $n = 1000$.

Fig. 3 shows the combined performance of the two algorithms for $n = 100$. For this graph, the y-axis is given a logarithmic scaling and is cut off at 100 to increase readability. There is more than 99% effectiveness at the top and bottom of the graph with minima in the middle. The minimum effectiveness of the algorithms steadily decreases as more goals are added, indicating that the problem becomes progressively harder with additional goals. The middle right of the graph contains a region where the combined effectiveness of the algorithms is less than 1%. There is no point in the empirical data where the combined effectiveness is 0%. The points where 100% empirical effectiveness occurs are similarly spaced on the graph from the 99% points. I.e., the distance on the graph between the 100% points and the 99% points is similar to the distance between the 99% points and the 90% points. The 100% points have high variability, which is why they are not displayed.

The results for $n = 1000$ propositions are displayed in Figs. 4, 5, and 6. This experiment took over one CPU-week on a Sparc 5 running Lucid Common Lisp. The most significant differences are as follows.

- There is about a ten- to twenty-fold increase in the number of operators for high levels of effectiveness. For the 99% effectiveness level for POSTS-COVER-GOALS (comparing Fig. 4 versus Fig. 1), the $n = 1000$ data starts at about 1500 operators for $g = 20$, and goes up to over 5000 operators for $g = 1000$; the $n = 100$ data has about 150 operators for $g = 20$, up to about 300 operators for $g = 100$. For the 99% effectiveness level for PLAN-FORWARD (comparing Fig. 5 versus Fig. 2), the

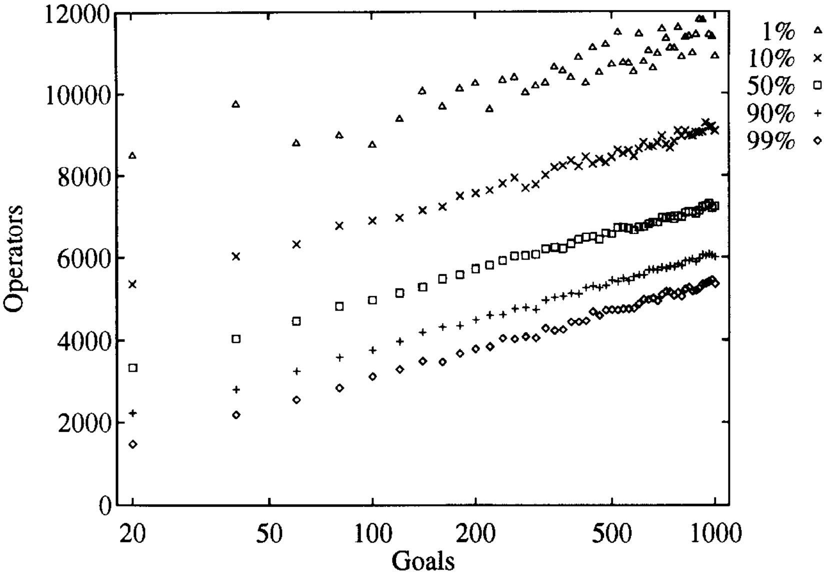

Fig. 4. Empirical effectiveness of POSTS-COVER-GOALS for 1000 propositions. The x-axis is logarithmically scaled.

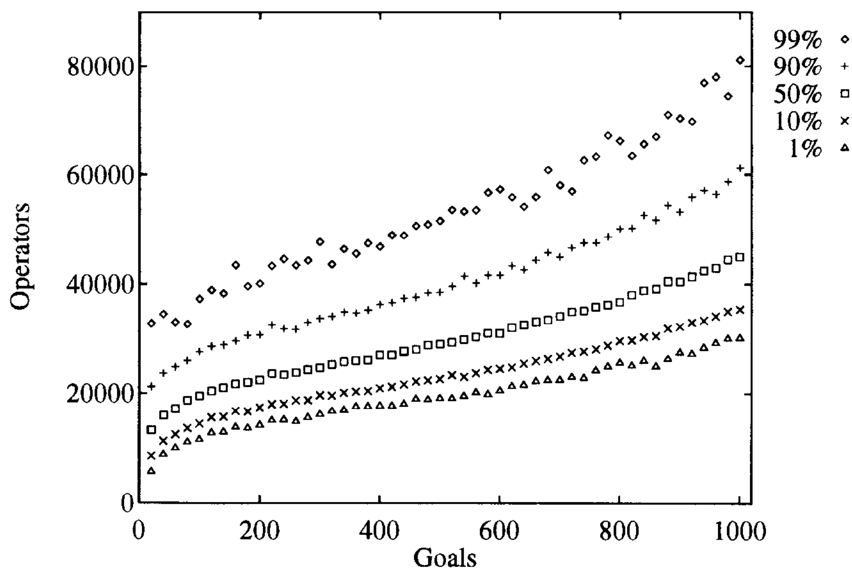

Fig. 5. Empirical effectiveness of PLAN-FORWARD for 1000 propositions.

$n = 1000$ data starts at about 30,000 operators and goes up to 80,000. The $n = 100$ data starts at about 2000 operators, going up to about 6000 operators. This closely corresponds to a combination of a ten-fold increase in the number of propositions and a 50% increase in $\ln g$ from $g = 100$ to $g = 1000$.

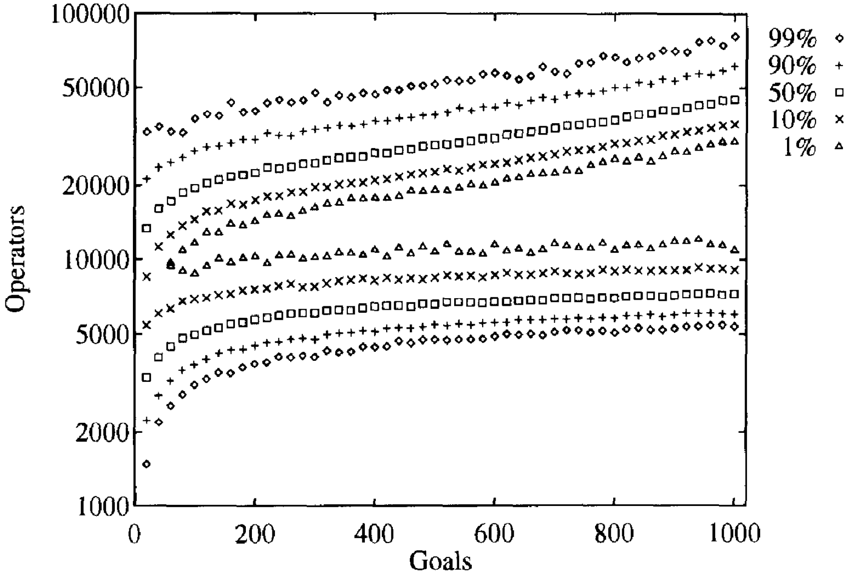

Fig. 6. Combined empirical effectiveness of POSTS-COVER-GOALS and PLAN-FORWARD for 1000 propositions. The y-axis is logarithmically scaled.

- The effectiveness levels for PLAN-FORWARD again appears to be primarily linear in the number of goals, perhaps even mildly exponential for large numbers of goals. This is explained by the $e^{sg/n}$ term in the theoretical analysis as described above.
- Fig. 6 displays a much larger region of hard instances for $n = 1000$, as compared to $n = 100$. The 1% effectiveness region now extends almost to the left side of the graph. Again, the instances become progressively harder as g increases. Roughly, the contour where 0% empirical effectiveness takes hold is similarly spaced from the 1% points on the graph. The 0% points, as might be expected, are highly variable, and so, are not displayed.

For both algorithms and both values of the number of propositions n, the transition from 1% to 99% effectiveness appears smooth; nevertheless, as the number of operators increase, there is an easy-hard-easy pattern. This is shown in Fig. 7 for the parameter values $n = 1000$ and $g = 500$. The x-axis in the figure corresponds to the number of operators and is logarithmically scaled; the y-axis is the empirical effectiveness or probability of a definitive answer by the algorithms. Initially, at least 99% of the instances are solved by POSTS-COVER-GOALS until there are $o \approx 4700$ operators. The effectiveness of POSTS-COVER-GOALS drops to 1% at $o \approx 10,500$. None of the instances are solved by either POSTS-COVER-GOALS and PLAN-FORWARD from $o \approx 13,000$ to $o \approx 16,000$. PLAN-FORWARD solves 1% of the instances at $o \approx 19,000$. Finally at $o \approx 52,000$, 99% of the instances are solved by PLAN-FORWARD.

A valuable addition to Fig. 7 would be a display of the probability of a solution plan dependent on the number of operators, showing the location of the transition from

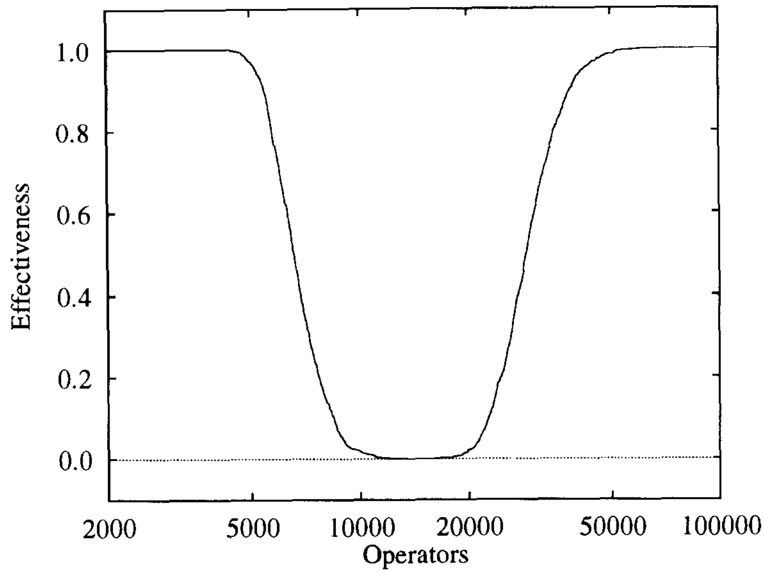

Fig. 7. Empirical effectiveness of POSTS-COVER-GOALS and PLAN-FORWARD for 1000 propositions and 500 goals. The x-axis is logarithmically scaled.

planning instances with no solution plans to those with solution plans. Unfortunately, the empirical data from running the algorithms are clearly insufficient to determine whether the transition is sharp or not and to determine the location of the 50% transition point. Also unknown are the lengths of the shortest solution plan for those instances that have solution plans. Determining this information appears infeasible even for $n = 100$ because it would be slow to use any systematic search to find a possibly very long plan in a search space of size 2^{100}.

6. Remarks

I have shown that determining plan existence for propositional STRIPS planning is usually easy if the number of operators satisfy certain bounds, and if each possible precondition and postcondition is equally likely to appear within an operator, independently of other operators. Assuming that the expected numbers of pre- and postconditions are fixed, then it is usually easy to show that no plan exists for instances with n propositions, g goals, and the number of operators below an $O(n \ln g)$ bound, and it is usually easy to find plans for instances with n propositions, g goals, and the number of operators above an $\Omega(n \ln g)$ bound. In addition, plan modification instances are usually easy to solve if there are $\Omega(n)$ operators. The constants for the latter two results are exponential in the expected numbers of pre- and postconditions.

The $\Omega(n \ln g)$ result was demonstrated for three simple planning algorithms. Searching from the initial state to the goals appears to have a slight advantage over searching backward from the goals. However, it appears possible in many cases to search backward from the goals to find a plan that is largely independent of the initial state. Of course, it should be mentioned that rather crude inequalities are used in the analysis to derive simplified expressions. The empirical behavior of the POSTS-COVER-GOALS and PLAN-FORWARD algorithm corresponded to the theoretical terms; however, the theoretical bounds for PLAN-FORWARD are very conservative.

This work complements and extends previous average-case analyses for NP-complete problems. It complements previous work because it suggests that random planning instances are hard only for a narrow range of a particular parameter, in this case, the number of operators. It extends previous work because the worst-case complexity of propositional STRIPS planning is PSPACE-complete, thus, suggesting that PSPACE-complete problems exhibit threshold phenomena similar to NP-complete problems. The empirical study also resulted in the easy-hard-easy pattern characteristic of random instances of hard problems, though with smooth transitions.

This work also provides theoretical and empirical support for reactive behavior. A main tenet of reactive behavior is that sound and complete planning, besides being too inefficient, is often unnecessary, i.e., states can be mapped to appropriate operators without much lookahead. The analysis of the PLAN-FORWARD algorithm, which only does a one-step lookahead, shows that this tenet is true for a large subset of the planning problem.

Further work is needed to narrow the gap between the bounds derived by this paper and to analyze more realistic distributions. In particular, the assumption that operators are independently selected is clearly wrong. Nevertheless, it would be interesting to empirically test how well the bounds of this paper classify the hardness of realistic planning problems.

Also, more sophisticated algorithms should be analyzed and empirically tested. It will be a challenge to develop planning algorithms that are sufficiently efficient and effective to compare against the empirical results in this paper.

Acknowledgements

Comments from anonymous reviewers and Tad Hogg led to the empirical study that was performed, which in turn led to improved theoretical results. Jerry Keating pointed out the use of Bonferroni's inequality to me.

Appendix A. Proofs of some inequalities

The proof of Lemma 4 depends on upper and lower bounds for the recurrence relation:

$$f(s,n,k) = \frac{n-k}{n} f(s-1,n-1,k) + \frac{k}{2n} f(s-1,n-1,k-1),$$

for a positive integer n and nonnegative integers s and k, $s \leqslant n$ and $k \leqslant n$, where the base cases are $f(0,n,k) = 1$, $f(s,n,0) = 1$, $f(s,n,n) = 2^{-s}$, and $f(n,n,k) = 2^{-k}$. See the proof of Lemma 4 for a justification of this equation.

Lemma A.1.

$$f(s,n,k) \geqslant \mathrm{e}^{-sk/n}.$$

Proof. In the base cases:

$$\begin{aligned}
f(0,n,k) &= 1 = \mathrm{e}^0,\\
f(s,n,0) &= 1 = \mathrm{e}^0,\\
f(s,n,n) &= 2^{-s} \geqslant \mathrm{e}^{-s},\\
f(n,n,k) &= 2^{-k} \geqslant \mathrm{e}^{-k}.
\end{aligned}$$

Using mathematical induction, assume that (s,n,k) is not a base case, and that the inequality holds for all tuples less than (s,n,k). Then:

$$\begin{aligned}
f(s,n,k) &= \frac{n-k}{n} f(s-1,n-1,k) + \frac{k}{2n} f(s-1,n-1,k-1)\\
&\geqslant \frac{n-k}{n} \exp\left\{-\frac{(s-1)k}{n-1}\right\} + \frac{k}{2n} \exp\left\{-\frac{(s-1)(k-1)}{n-1}\right\}.
\end{aligned}$$

For each exponential:

$$\begin{aligned}
\exp\left\{-\frac{(s-1)k}{n-1}\right\} &= \mathrm{e}^{-sk/n} \exp\left\{\frac{nk-sk}{n(n-1)}\right\},\\
\exp\left\{-\frac{(s-1)(k-1)}{n-1}\right\} &= \mathrm{e}^{-sk/n} \exp\left\{\frac{nk-sk+ns-n}{n(n-1)}\right\}.
\end{aligned}$$

Using $\mathrm{e}^x \geqslant 1 + x$:

$$\begin{aligned}
f(s,n,k) &\geqslant \mathrm{e}^{-sk/n} \left(\frac{n-k}{n} \exp\left\{\frac{nk-sk}{n(n-1)}\right\} + \frac{k}{2n} \exp\left\{\frac{nk-sk+ns-n}{n(n-1)}\right\}\right)\\
&\geqslant \mathrm{e}^{-sk/n} \left(\frac{n-k}{n} \left(1 + \frac{nk-sk}{n(n-1)}\right) + \frac{k}{2n} \left(1 + \frac{nk-sk+ns-n}{n(n-1)}\right)\right)\\
&= \mathrm{e}^{-sk/n} \frac{2n^3 - 2n^2 + kn^2 - kns - knk + ksk}{2n^2(n-1)}\\
&= \mathrm{e}^{-sk/n} \left(1 + \frac{k(n-s)(n-k)}{2n^2(n-1)}\right)\\
&\geqslant \mathrm{e}^{-sk/n}.
\end{aligned}$$

The final inequality follows because $n \geqslant s$ and $n \geqslant k$. $\square$

Lemma A.2.

$$f(s,n,k) \leqslant \frac{2n}{2n+sk}.$$

Proof. In the base cases:

$$f(0,n,k) = 1 = \frac{2n}{2n+0k},$$
$$f(s,n,0) = 1 = \frac{2n}{2n+0s},$$
$$f(s,n,n) = 2^{-s} = \mathrm{e}^{-s\ln 2} \leqslant \mathrm{e}^{-s/2} \leqslant \frac{2}{2+s} = \frac{2n}{2n+sn},$$
$$f(n,n,k) = 2^{-k} \leqslant \frac{2n}{2n+nk}.$$

Using mathematical induction, assume that (s,n,k) is not a base case, and that the inequality holds for all tuples less than (s,n,k). Then:

$$\begin{aligned} f(s,n,k) &= \frac{n-k}{n} f(s-1,n-1,k) + \frac{k}{2n} f(s-1,n-1,k-1) \\ &\leqslant \frac{n-k}{n}\left(\frac{2(n-1)}{2(n-1)+(s-1)k}\right) + \frac{k}{2n}\left(\frac{2(n-1)}{2(n-1)+(s-1)(k-1)}\right). \end{aligned}$$

A lot of tedious algebra leads to:

$$\begin{aligned} &f(s,n,k) \\ &\leqslant \frac{2n}{2n+sk} \cdot \\ &\left(1 + \frac{2kn(s-n)(s-1) + k^3 s(s-1)(1-n) + 2k^2(-sn^2+3sn-n-s^2)}{2n^2(n-1)(2(n-1)+(s-1)k)(2(n-1)+(s-1)(k-1))}\right). \end{aligned}$$

Considering the numerator of the large fraction, $2kn(s-n)(s-1) \leqslant 0$ because $s-n \leqslant 0$ and the other terms are nonnegative. Also, $k^3 s(s-1)(1-n) \leqslant 0$ because $1-n \leqslant 0$ and the other terms are nonnegative. This leaves $2k^2(-sn^2+3sn-n-s^2)$ to consider. If $n \geqslant 3$, then $-sn^2+3sn = sn(3-n) \leqslant 0$, and the remaining $-n-s^2$ is negative. If $n=2$, $s=2$ and $s=0$ are base cases, and if $s=1$, $-sn^2+3sn-n-s^2=0$. $n=1$ and $n=0$ must be base cases, i.e., at least one of $s=0$, $k=0$, $s=n$, or $k=n$ must be true. Thus, the numerator must be nonpositive, which implies that $f(s,n,k) \leqslant 2n/(2n+sk)$. □

References

[1] C. Bäckström and I. Klein, Parallel non-binary planning in polynomial time, in: *Proceedings IJCAI-91*, Sydney, Australia (1991) 268–273.

[2] T. Bylander, Complexity results for planning, in: *Proceedings IJCAI-91*, Sydney, Australia (1991) 274–279.

[3] T. Bylander, An average case analysis of planning, in: *Proceedings AAAI-93*, Washington, DC (1993) 480–485.

[4] T. Bylander, The computational complexity of propositional STRIPS planning, *Artif. Intell.* **69** (1994) 161–204.

[5] D. Chapman, Planning for conjunctive goals, *Artif. Intell.* **32** (3) (1987) 333–377; also in: J. Allen, J. Hendler and A. Tate, eds., *Readings in Planning* (Morgan Kaufmann, San Mateo, CA, 1990).

[6] P. Cheeseman, B. Kanefsky and W.M. Taylor, Where the really hard problems are, in: *Proceedings IJCAI-91*, Sydney, Australia (1991) 331-337.

[7] P.R. Cohen, A survey of the Eighth National Conference on Artificial Intelligence: pulling together or pulling apart?, *AI Mag.* **12** (1) (1991) 17-41.

[8] T.H. Cormen, C.E. Leiserson and R.L. Rivest, *Introduction to Algorithms* (MIT Press, Cambridge, MA, 1990).

[9] J.M. Crawford and L.D. Auton, Experimental results on the crossover point in satisfiability problems, in: *Proceedings AAAI-93*, Washington, DC (1993) 46-51.

[10] K. Erol, D.S. Nau and V.S. Subrahmanian, On the complexity of domain-independent planning, in: *Proceedings AAAI-92*, San Jose, CA (1992) 381-386.

[11] K. Erol, D.S. Nau and V.S. Subrahmanian, When is planning decidable?, in: *Proceedings First International Conference on AI Planning Systems* (1992) 222-227.

[12] R.E. Fikes and N.J. Nilsson, STRIPS: a new approach to the application of theorem proving to problem solving, *Artif. Intell.* **2** (3/4) (1971) 189-208; also in: J. Allen, J. Hendler and A. Tate, eds., *Readings in Planning* (Morgan Kaufmann, San Mateo, CA, 1990).

[13] M.R. Garey and D.S. Johnson, *Computers and Intractability* (Freeman, New York, 1979).

[14] K.J. Hammond, Explaining and repairing plans that fail, *Artif. Intell.* **45** (1-2) (1990) 173-228.

[15] S. Kambhampati and J.A. Hendler, A validation-structure-based theory of plan modification and reuse, *Artif. Intell.* **55** (2-3) (1992) 193-258.

[16] D. McAllester and D. Rosenblitt, Systematic nonlinear planning, in: *Proceedings AAAI-91*, Anaheim, CA (1991) 634-639.

[17] M. Minsky, Logical versus analogical or symbolic versus connectionist or neat versus scruffy, *AI Mag.* **12** (2) (1991) 34-51.

[18] S. Minton, M.D. Johnston, A.B. Philips and P. Laird, Minimizing conflicts: a heuristic repair method for constraint satisfaction and scheduling problems, *Artif. Intell.* **58** (1992) 161-205.

[19] D. Mitchell, B. Selman and H.J. Levesque, Hard and easy distributions of SAT problems, in: *Proceedings AAAI-92*, San Jose, CA (1992) 459-465.

[20] R. Musick and S. Russell, How long will it take?, in: *Proceedings AAAI-92*, San Jose, CA (1992) 466-471.

[21] B. Nebel and J. Koehler, Plan modification versus plan generation: a complexity-theoretic perspective, in: *Proceedings IJCAI-93*, Chambery, France (1993) 1436-1441.

[22] N.J. Nilsson, *Principles of Artificial Intelligence* (Tioga, Palo Alto, CA, 1980).

[23] J.S. Penberthy and D.S. Weld, UCPOP: a sound, complete, partial order planning for ADL, in: *Proceedings Third International Conference on Principles of Knowledge Representation and Reasoning*, Cambridge, MA (1992) 103-114.

[24] B. Selman and H.A. Kautz, An empirical study of greedy local search for satisfiability testing, in: *Proceedings AAAI-93*,, Washington, DC (1993) 46-51.

[25] B. Selman, H. Levesque and D. Mitchell, A new method for solving hard satisfiability problems, in: *Proceedings AAAI-92*, San Jose, CA (1992) 440-446.

[26] C.P. Williams and T. Hogg, Using deep structure to locate hard problems, in: *Proceedings AAAI-92*, San Jose, CA (1992) 472-477.