Published in *Proceedings of the 13th International Joint Conference on Artificial Intelligence (IJCAI'93)*, Chambery, France, August 1993, 1430–1435.

Complexity Results for SAS$^+$ Planning*

Christer Bäckström
Department of Computer
and Information Science,
Linköping University
S-581 83 Linköping, Sweden
email: cba@ida.liu.se

Bernhard Nebel
German Research Center for
Artificial Intelligence (DFKI)
Stuhlsatzenhausweg 3
D-6600 Saarbrücken 11, Germany
email: nebel@dfki.uni-sb.de

Abstract

We have previously reported a number of tractable planning problems defined in the SAS$^+$ formalism. This paper complements these results by providing a complete map over the complexity of SAS$^+$ planning under all combinations of the previously considered restrictions. We analyse the complexity both of finding a minimal plan and of finding any plan. In contrast to other complexity surveys of planning we study not only the complexity of the existence problems but also of the search problems. We prove that the SAS$^+$-PUS problem is the maximal tractable problem under the restrictions we have considered if we want to find a minimal plan. If we are satisfied with finding any plan, then we can generalize further to the SAS$^+$-US problem, which we prove to be the maximal tractable problem in this case.

1 Introduction

We have previously reported a number of tractable planning problems using the SAS$^+$ and SAS formalisms. We started by identifying the SAS-PUBS problem [Bäckström and Klein, 1991b] and prove this problem to be tractable. Then we applied a bottom-up strategy, generalizing the results by removing restrictions, which resulted in the more expressive, but still tractable, SAS-PUS [Bäckström and Klein, 1991a] and SAS$^+$-PUS [Bäckström, 1992a, 1992b] problems. The overall goal of this research has been to identify successively more and more expressive, tractable planning problems, with the hope of ultimately finding problems which are relevant to practical applications, especially in *sequential control*.[1]

We found the tractable problems mentioned above by studying a test problem in sequential control and identifying a number of inherent restrictions (denoted P, U, B and S) on that problem. We have proven [Bäckström, 1992a, 1992b] that the SAS$^+$ formalism is expressively equivalent to 'standard' variants of propositional STRIPS and ground TWEAK. Yet, it was essential to use the SAS$^+$ formalism since the P, U, B and S restrictions cannot be naturally expressed in any of the other formalisms [Bäckström, 1992a Bäckström and Nebel, 1992] and it is unlikely that we would have managed to isolate them if using a standard formalism.

Having started from one tractable problem and generalized twice to new tractable problems it is interesting to ask how much further we can generalize and stay tractable by simply removing restrictions. This paper answers that question by providing an exhaustive map over the complexities for finding both optimal and non-optimal plans for all possible combinations of the P, U, B and S restrictions. It turns out that the SAS$^+$-PUS problem is the maximally expressive tractable problem if we insist on finding optimal (*ie.* minimal length) plans. Whichever of the three restrictions on this problem we drop, the resulting problem is intractable. On the other hand, if we do not require the solutions to be minimal, then we can generalize somewhat further. We prove that if we remove the P restriction, resulting in the SAS$^+$-US problem, we can still plan in polynomial time, although we are no longer guaranteed to find optimal plans.

2 The SAS$^+$ Formalism

This section briefly recasts the main differences between the SAS$^+$ formalism and the STRIPS formalism and also gives a somewhat simplified account of the formal definition. Due to the page limit we have to refer the reader to previous publications [Bäckström and Klein, 1991a, 1991b, Bäckström, 1992a, 1992b] for background, motivation and examples.

2.1 World Modelling

There are basically two important differences between the SAS$^+$ formalism and the propositional STRIPS formalism. The first one is that the SAS$^+$ formalism uses partial, multi-valued state variables instead of propositional atoms. The second difference is that the operators also have a *prevail-condition* in addition to the usual pre- and post-conditions. This makes it possible to distinguish easily between those parts of the (tra-

*This research was supported by the Swedish National Board for the Engineering Sciences (TFR) under grant Dnr. 92-143 and by the German Ministry for Research and Technology (BMFT) under grant ITW 8901 8.

[1] Sequential control is a subdisciplin of *automatic control* and includes many planning-like problems.

ditional STRIPS) pre-condition that is changed by the operator and the part that remains unchanged.[2] That is, the (SAS^+) pre-condition of an operator specifies those state variables which must have a certain defined value in order to execute the operator and that will also be *changed* to some other value by the operator. The prevail-condition, on the other hand, specifies those state variables that must have a certain value but will remain *unchanged* after executing the operator.

Definition 2.1 *A* **SAS^+-structure** $\Phi = \langle \mathcal{M}, \mathcal{S}, \mathcal{H} \rangle$ *is defined by:*

- *a set* $\mathcal{M} = \{i_1, \ldots, i_m\}$ *of* **state variable indices***;*
- *a* **space** $\mathcal{S} = \mathcal{S}_{i_1} \times \ldots \times \mathcal{S}_{i_m}$ **of total states***, where for each* $j \in \mathcal{M}$*,*
  - $\mathcal{S}_j$ *is a* **domain** *of mutually exclusive values for the jth state variable and*
  - $\mathcal{S}_j^+ = \mathcal{S}_j \cup \{\mathsf{u}\}$ *is the* **extended domain** *for the jth state variable,where* u *denotes the* **undefined value***,*

  and the **space** $\mathcal{S}^+ = \mathcal{S}_{i_1}^+ \times \ldots \times \mathcal{S}_{i_m}^+$ **of partial states** *is implicitly defined;*
- *a set* $\mathcal{H}$ *of* **operators** *(action types), each* $h \in \mathcal{H}$ *being of the form* $h = \langle \mathsf{b}, \mathsf{e}, \mathsf{f} \rangle \in \mathcal{S}^+ \times \mathcal{S}^+ \times \mathcal{S}^+$ *where* b*,* e *and* f*, denote the pre-, post- and prevail-condition respectively of* h*, and where the set* $\mathcal{H}$ *is subject to the following restrictions:*
  - *for all* $h \in \mathcal{H}$ *and for all* $i \in \mathcal{M}$*, if* $\mathsf{b}(h)[i] \neq \mathsf{u}$*, then* $\mathsf{b}(h)[i] \neq \mathsf{e}(h)[i] \neq \mathsf{u}$
  - *for all* $h \in \mathcal{H}$ *and* $i \in \mathcal{M}$*, either* $\mathsf{e}(h)[i] = \mathsf{u}$ *or* $\mathsf{f}(h)[i] = \mathsf{u}$*.*

The first of the two restrictions on $\mathcal{H}$ expresses that all state variables having a defined value in the pre-condition of some operator to have a defined but different value in the post-condition. That is, an operator cannot change a state variable from a defined value to the undefined value and it must either change a variable or not define it at all in the pre- and post-conditions; Variables that are defined but not changed should go into the prevail-condition. The second restriction expresses that the post-condition and the prevail-condition of an operator must not define the same state variables since an operator cannot both change a state variable and require it to be constant. Neither of these is a restriction in practice.

We write $s[i]$ to denote the value of the ith state variable in a state s. We also write $s \sqsubseteq t$ if the state s is subsumed (or satisfied) by state t, *ie.*,

$$s \sqsubseteq t \text{ iff } \forall i \in \mathcal{M}(s[i] = \mathsf{u} \text{ or } s[i] = t[i]).$$

2.2 Plans

To simplify matters we only define linear plans and do not distinguish between operators and their instantiations (actions).

[2]This distinction could be implicitly derived from the ordinary STRIPS pre- and post-conditions but we find it both formally and conceptually clearer to make it explicit. Furthermore, the distinction is important if we also consider parallel execution of operators [Bäckström and Klein, 1991b, 1991a, Bäckström, 1992a].

Definition 2.2 *A* **plan** *over a SAS^+-structure* $\Phi = \langle \mathcal{M}, \mathcal{S}, \mathcal{H} \rangle$ *is a sequence of operators* $\langle h_1, \ldots, h_n \rangle$ *s.t.* $h_k \in \mathcal{H}$ *for* $1 \leq k \leq n$*.*

Given two plans $\bar{\alpha} = \langle h_1, \ldots, h_m \rangle$ and $\bar{\beta} = \langle h'_1, \ldots, h'_n \rangle$ we define $(\bar{\alpha};\bar{\beta}) = \langle h_1, \ldots, h_m, h'_1, \ldots, h'_n \rangle$. The result of executing a plan is defined recursively as follows, using the *update* function $\oplus$.

Definition 2.3 *Given two states* $s, t \in \mathcal{S}^+$*, we define for all* $i \in \mathcal{M}$*,*

$$(s \oplus t)[i] = \begin{cases} t[i] & \textit{if } t[i] \neq \mathsf{u}, \\ s[i] & \textit{otherwise.} \end{cases}$$

The function result gives the state resulting from executing a plan and is defined recursively as:

$$\begin{aligned} &\mathit{result}(s, \langle\rangle) = s, \\ &\mathit{result}(s, (\bar{\alpha};\langle h \rangle)) = \\ &\quad \begin{cases} \mathit{result}(s, \bar{\alpha}) \oplus \mathsf{e}(h) & \text{if } \mathsf{b}(h) \sqsubseteq \mathit{result}(s, \bar{\alpha}) \\ & \text{ and } \mathsf{f}(h) \sqsubseteq \mathit{result}(s, \bar{\alpha}), \\ \langle \mathsf{u}, \ldots, \mathsf{u} \rangle & \text{otherwise.} \end{cases} \end{aligned}$$

An operator h is **admissible** wrt. a state s iff $\mathsf{b}(h) \sqsubseteq s$ and $\mathsf{f}(h) \sqsubseteq s$. A plan $\bar{\alpha} = \langle h_1, \ldots, h_n \rangle$ is admissible wrt. a state s iff h_1 is admissible wrt. s and for $1 < k \leq n$, h_k is admissible wrt. $\mathit{result}(s, \langle h_1, \ldots, h_{k-1} \rangle)$. In addition, the empty plan $\langle\rangle$ is admissible wrt. any state.

Definition 2.4 *An* **instance of the SAS^+ planning problem** *is a tuple* $\Pi = \langle \Phi, s_0, s_\star \rangle$ *s.t.* Φ *is a SAS^+-structure and* $s_0, s_\star \in \mathcal{S}^+$ *denote the* **initial state** *and* **goal state** *respectively. A plan* $\bar{\alpha}$ *over* Φ **solves** Π *iff*

1. $\bar{\alpha}$ *is admissible wrt.* s_0 *and*
2. $s_\star \sqsubseteq \mathit{result}(s_0, \bar{\alpha})$*.*

More specifically we distinguish four different problems. The **SAS^+plan existence problem** *is: given an instance* Π*, decide whether there exists some plan* $\bar{\alpha}$ *over* Φ *s.t.* $\bar{\alpha}$ *solves* Π*. The* **SAS^+plan search problem** *is: given an instance* Π*, find a plan* $\bar{\alpha}$ *over* Φ *that solves* Π*, or answer that there is no such plan. The corresponding* **bounded** *plan existence (search) problem takes an extra parameter* K *and asks only for plans of length* K *or shorter.*

3 Restrictions

We have previously identified four restrictions on the SAS^+ planning problem that together result in tractability. An instance of the SAS^+ problem is *post-unique* (P) iff no two distinct operators can change the same state variable to the same value and it is *unary* (U) iff each operator changes exactly one state variable. The instance is *binary* (B) iff all state variable domains are two-valued. Finally, the instance is *single-valued* (S) iff any two operators that both require the same state variable to have some specific value during their respective occurrences must require the *same* defined value. For example, single-valuedness prevents us from having two operators such that one requires a certain room to be lit during its occurrence while the other requires the same room to be dark during its occurrence.

Definition 3.1 *A SAS$^+$ structure $\Phi = \langle \mathcal{M}, \mathcal{S}, \mathcal{H} \rangle$ is*

- **binary** *iff $|\mathcal{S}_i| = 2$ for all $i \in \mathcal{M}$,*
- **post-unique** *iff for all $h, h' \in \mathcal{H}$, if $\mathsf{e}(h)[i] = \mathsf{e}(h')[i] \neq \mathsf{u}$ for some $i \in \mathcal{M}$, then $h = h'$;*
- **unary** *iff for all $h \in \mathcal{H}$, there is exactly one $i \in \mathcal{M}$ s.t. $\mathsf{e}(h)[i] \neq \mathsf{u}$;*
- **single-valued** *iff there exists some state s s.t. $\mathsf{f}(h) \sqsubseteq s$ for all $h \in \mathcal{H}$. In particular, we define the* **global prevail-condition** *$\hat{\mathsf{f}}_{\mathcal{H}}$ for $\mathcal{H}$ as the minimal such s (wrt. the number of defined values).*

All these restrictions were identified by studying a test example in automatic control, thus complementing the usual problems from the AI world. For a somewhat more elaborate discussion of the restrictions, see Bäckström and Klein [1991a] or Bäckström [1992a].

We name subproblems of the SAS$^+$ problem satisfying combinations of the above restrictions by appending the letters denoting these restrictions. For example, the SAS$^+$-PUS problem is the SAS$^+$ problem restricted to instances that are post-unique, unary and single-valued while the SAS$^+$-B problem is only restricted to binary state variables.

Since we have previously studied also the slightly more restricted SAS formalism and it could, thus, be interesting to view also SAS problems as restricted versions of the corresponding SAS$^+$ problems. It turns out, however, that all complexity results presented in this paper hold regardless of whether we consider SAS$^+$-structures or SAS-structures. The complexity figures depend only on the restrictions in Definition 3.1.

4 Existence of Optimal Plans

We already know [Bäckström, 1992a, 1992b] that we can find optimal (in the sense of minimal length) plans for the SAS$^+$-PUS problem in polynomial time.

Theorem 4.1 *Bounded SAS$^+$-PUS plan existence and plan search is solvable in polynomial time.*

Since this problem is a generalization of the previously studied, tractable SAS-PUBS [Bäckström and Klein, 1991b] and SAS-PUS [Bäckström and Klein, 1991a] planning problems, it is interesting to ask whether we can generalize even further, staying tractable. Unfortunately, it turns out that we cannot remove any of the three restrictions (P, U and S) and still find optimal plans tractably, as we will see in this section. The complexity results for bounded plan existence are summarized in Figure 1 for all possible combinations of the restrictions in Definition 3.1.

The following theorems together establish that the SAS$^+$-PUS problems is the maximal tractable problem wrt. the P, U and S restrictions. Due to space limitations we will sometimes omit proofs or provide only proof sketches, but the full proofs of all theorems appear in Bäckström and Nebel [1993].

We first note that all problems which are both unary and single-valued have polynomially-sized minimal plans.

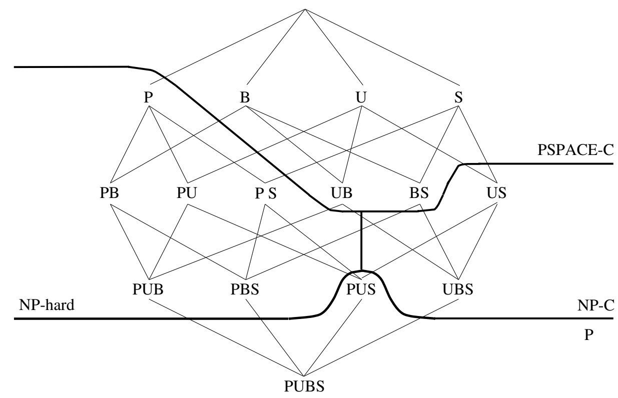

Figure 1: Complexity of bounded plan existence for the SAS$^+$ problem and its subproblems.

Theorem 4.2 *All solvable instances of the SAS$^+$-US problem have minimal solutions which are of polynomially bounded length.*

Proof: Appears in Bäckström and Nebel [1993]. It is also an immediate consequence of Theorem 6.1 in this paper. □

We can now prove the NP-hardness results.

Theorem 4.3 *Bounded plan existence is NP-complete in the strong sense for SAS$^+$-UBS and SAS$^+$-US.*

Proof: We prove NP-hardness of SAS$^+$-UBS plan existence by reduction from *minimum cover* [Garey and Johnson, 1979, page 222] which is defined as follows. Let $X = \{x_1, \ldots, x_m\}$ be a set, let $C = \{C_1, \ldots, C_n\}$ be a set of subsets of X and let K be an integer. Does there exist a cover for X, *ie.*, a subcollection $C' \subseteq C$ s.t. $\bigcup_{C_k \in C'} C_k = X$ and $|C'| \leq K$.

Define one binary state variable, x_k, for each $x_k \in X$ and one, c_l, for each $C_l \in C$. Further define an operator h_l for each c_l s.t. h_l sets c_l and also define an operator $h_{l,k}$ for each C_l and each $x_k \in C_l$ s.t. $h_{l,k}$ can set x_k iff c_l is set. Let all state variables be reset initially and require all x_k to be set in the goal state. It is obvious that X has a cover C' s.t. $|C'| \leq K$ iff there is a plan of size $|X| + K$ or less solving Π.

Both problems are in NP since the minimal solutions are always of polynomial length (Theorem 4.2) and can, thus, be verified in polynomial time. □

Theorem 4.4 *Bounded plan existence is NP-hard in the strong sense for SAS$^+$-PUB.*

Proof sketch: NP-hardness in the strong sense follows by a reduction from the *clique problem* [Garey and Johnson, 1979, p. 194]. Assuming a graph $G = \langle V, E \rangle$, the underlying idea of the proof is to define a SAS$^+$-PUB instance as follows. For each vertex $v_i \in V$ we use five state variables A_i, B_i, C_i, D_i and V_i and eight operators a_i^+, b_i^+, c_i^+, c_i^-, d_i^+, d_i^-, v_i^+ and v_i^- s.t. operators denoted + (-) set (reset) their corresponding state variables. The prevail-conditions are chosen s.t. a_i^+ can be executed only if C_i is set and D_i is reset, while b_i^+ can be executed only if C_i is reset and D_i is set. We let all state variables be reset initially and require all variables but V_i to be set in the goal state. Obviously, any plan must include either of the operator sequences $\langle c_i^+, a_i^+, c_i^-, d_i^+, b_i^+, c_i^+\rangle$ and $\langle d_i^+, b_i^+, d_i^-, c_i^+, a_i^+, d_i^+\rangle$. We further add prevail-conditions s.t. c_i^- be executable only if V_i is set and d_i^- is executable only if V_j is set for all v_j not incident to v_i in G. It can be proven that G has a clique of size k iff the above SAS$^+$-PUB instance has a plan of length $8|V| - 2k$ or less. □

Theorem 4.5 *Bounded plan existence is NP-hard in the strong sense for SAS$^+$-PBS.*

Proof sketch: SAS$^+$-PUB plan existence can be reduced to SAS$^+$-PBS plan existence as follows. For each state variable x add a new state variable x' and redefine the operators s.t. x' is set when x is reset, reset when x is set and undefined when x is undefined. This is possible since the new instance need not be unary. Furthermore, we can now test whether x is reset by instead testing whether x' is set, so the operators can be redefined to be single-valued. □

If we further drop post-uniqueness, then the last two problems become PSPACE-complete.

Theorem 4.6 *Both bounded and unbounded SAS$^+$plan existence is PSPACE-complete.*

Proof: Immediate from the formalism equivalence result [Bäckström, 1992a, Corollary 5.18] and PSPACE-completeness for propositional STRIPS [Bylander, 1991, Theorem 1]. □

We also sharpen the PSPACE-completeness results presented by Bylander [1991] and Erol *et al.* [1992].

Theorem 4.7 *Both bounded and unbounded SAS$^+$-UB and SAS$^+$-BS plan existence is PSPACE-complete.*

5 Finding Optimal Plans

We are ultimately interested in finding a solution, not only finding out whether one exists. Hence, it is interesting to also analyse the complexity of the bounded plan *search* problem. Obviously, a search problem can never be easier than its corresponding existence problem, so the results in the previous section provide lower bounds for the bounded plan search problems. Furthermore, if we can solve the bounded plan existence problem, then we can also solve the bounded plan search problem by using *prefix search* [Balcázar *et al.*, 1988, Garey and Johnson, 1979, pp. 116–117]. It is important to note, however, that this method does not provide a polynomial reduction of the bounded search problem to the bounded existence problem. The reason for this is that some of the problems we consider do not have polynomially bounded minimal solutions, as we will see below. Hence, the prefix search method provides only a pseudo-polynomial reduction, *ie.*, the reduction is polynomially bounded in K, but not necessarily in the size of the representation of K.

Theorem 5.1 *The SAS$^+$-PUB and SAS$^+$-PBS planning problems have instances with exponentially sized minimal solutions.*

Proof sketch: Given $m > 0$, define a SAS$^+$-PUB instance $\Pi = \langle\langle\mathcal{M}, \mathcal{S}, \mathcal{H}\rangle, s_0, s_\star\rangle$ s.t. $\mathcal{M} = \{1, \ldots, m\}$; all $\mathcal{S}_i = \{0, 1\}$; $\mathcal{H} = \{h_1^+, h_1^-, \ldots, h_m^+, h_m^-\}$, where for $1 \leq k \leq m$,

$$\mathsf{b}(h_k^+)[i] = \mathsf{e}(h_k^-)[i] = \begin{cases} 0 & \text{if } i = k, \\ u & \text{otherwise}, \end{cases}$$

$$\mathsf{e}(h_k^+)[i] = \mathsf{b}(h_k^-)[i] = \begin{cases} 1 & \text{if } i = k, \\ u & \text{otherwise}, \end{cases}$$

$$\mathsf{f}(h_k^+)[i] = \mathsf{f}(h_k^-)[i] = \begin{cases} 0 & \text{if } i < k-1, \\ 1 & \text{if } i = k-1, \\ u & \text{otherwise}; \end{cases}$$

$s_0 = \langle 0, \ldots, 0\rangle$ and $s_\star = \langle 0, \ldots, 0, 1\rangle$. This instance has a unique minimal solution of length $2^m - 1$ corresponding to a Hamilton path in the state space.

The above SAS$^+$-PUB instance can be encoded as a SAS$^+$-PBS instance, using the same technique as in the proof of Theorem 4.5, which, thus, has a unique minimal plan of length $2^m - 1$. □

An immediate consequence of this theorem is that both bounded and unbounded SAS$^+$-PUB and SAS$^+$-PBS plan search is provably intractable since we might have to output a solution which is exponentially larger than the problem instance itself. Some care should be taken in interpreting these results, though. What we see here is the second cause for intractability as defined by Garey and Johnson [1979, pp. 11–12] and we should hardly regard instances with exponentially sized solutions as realistic. However, even if we look only for solutions with a specified length the problems are still NP-hard in the length parameter[3], which follows from the strong NP-hardness of SAS$^+$-PBS and SAS$^+$-PUB bounded plan existence. A further discussion on this topic appears in Bdckstrvm [1992a, pp. 142–147].

Finally, the following result is immediate from Theorem 4.

Theorem 5.2 *Bounded plan search is NP-equivalent for SAS$^+$-UBS and SAS$^+$-US*

6 Non-optimal Planning

While an algorithm for the bounded plan existence problem can be used to find plans, this is not the case for the unbounded existence problem. The reason is that we no longer have a measure of the distance to the goal. A polynomial-time algorithm used to prove that an existence problem is tractable can often be modified to also find a solution in polynomial time. This is not always the case, however. Hence, it seems that it is not of much use to know the complexity of the plan existence problem since we are ultimately interested in the complexity of finding a plan. Therefore, we focus only on the search problem in this section.

Figure 2 summarizes the complexity results for both the bounded and the unbounded plan search problems. The only difference is that the problems in the grey area are tractable for unbounded plan search but NP-equivalent for bounded plan search.

[3] That is, NP-hard in the value of the length parameter, not only in the size of its representation.

While the SAS$^+$-PUS problem was found to be the maximal tractable problem for optimal planning (wrt. the restrictions in Definition 3.1), this is no longer the case if we consider also non-optimal solutions. It turns out that we can find a solution in polynomial time even if we remove the P restriction, *ie.* if we have alternative ways of achieving an effect.

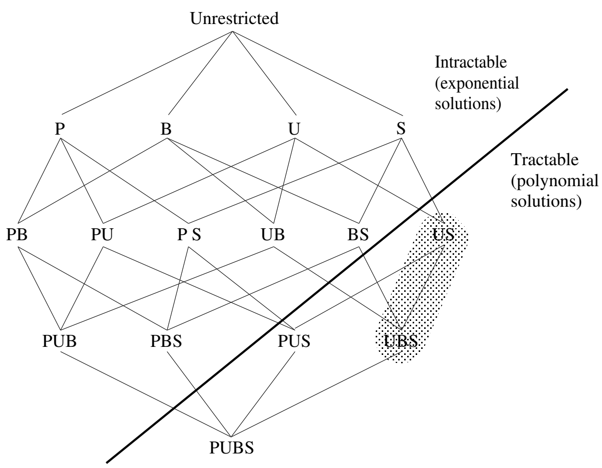

Figure 2: Complexity of plan search for the SAS$^+$ problem and its subproblems. The results hold also for bounded plan search, except for the problems in the grey area which are then NP-equivalent.

Theorem 6.1 *Plan search is solvable in polynomial time for SAS$^+$-US .*

Proof sketch: The following algorithm, *Plan*, finds plans for the SAS$^+$-US problem in polynomial time.[4]

Plan$(\mathcal{M}, \mathcal{S}, \mathcal{H}, s_0, s_\star)$
  $N \leftarrow \emptyset$
  loop
    $\langle s, \bar{\alpha} \rangle \leftarrow$ *AchievePrevail*(N)
    $\langle t, \bar{\beta} \rangle \leftarrow$ *AchieveGoal*(s)
    if $t \sqsubseteq s$ **then return** $(\bar{\alpha};\bar{\beta})$
    elsif $\exists i \in N (t[i] \not\sqsubseteq s[i])$ **then reject**
    else $N \leftarrow N \cup \{i \in \mathcal{M} | s[i] \neq \hat{\mathsf{f}}_{\mathcal{H}}[i]\}$
  end loop

AchievePrevail returns a state s and a plan $\bar{\alpha}$ from s_0 to s s.t. $s[i] = \hat{\mathsf{f}}_{\mathcal{H}}[i]$ for as many $i \in \mathcal{M} - N$ as possible and $s[i] = s_0[i]$ otherwise. Unariness and single-valuedness together guarantee that a unique such state exists, since achieving the global prevail-condition, $\hat{\mathsf{f}}_{\mathcal{H}}$, for one state variable cannot make $\hat{\mathsf{f}}_{\mathcal{H}}$ unachievable for another state variable. An iterated greedy strategy is used to find s and $\bar{\alpha}$, which is not guaranteed to be minimal.

Using a similar strategy, *AchieveGoal* returns a state t and a plan $\bar{\beta}$ from t to $s_\star$ s.t. $t[i] = s[i]$ for as many $i \in \mathcal{M}$ as possible and $t[i] = s_\star[i]$ otherwise.

[4]The algorithm is a descendant from an algorithm used by Bylander [1991, Theorem 7].

The first time around the loop in *Plan* the global prevail-condition is achieved for as many $i \in \mathcal{M}$ as possible in the state s, and t is the state 'closest' to s from which the goal can be achieved. If $s = t$, then s is a subgoal s.t. $\bar{\alpha}$ is a plan from s_0 to s and $\bar{\beta}$ is a plan from s to $s_\star$, and we are done. Otherwise, if $s[i] \neq t[i]$ for some $i \in \mathcal{M} - N$, then $s_\star[i]$ could not be achieved from $s[i]$ so $\hat{\mathsf{f}}_{\mathcal{H}}[i]$ must not be a subgoal. Hence, i is put in N to guarantee that $\hat{\mathsf{f}}_{\mathcal{H}}[i]$ is not achieved the next time around the loop. Furthermore, if there is some $i \in N$ s.t. $s[i] \neq t[i]$, then there cannot be any plan at all since $s_\star[i]$ cannot be achieved from $s_0[i]$ and not from $\hat{\mathsf{f}}_{\mathcal{H}}[i]$ (or $\hat{\mathsf{f}}_{\mathcal{H}}[i]$ itself could never be achieved).

Plan terminates since N grows strictly. To see that *Plan* is correct, note that N is empty the first time around the loop, so if $s_\star[i]$ could not be achieved from $s[i]$ because such a plan would require some operator h s.t. $\mathsf{f}(h) \not\sqsubseteq s$, then it could not be achieved at all. Furthermore, any prevail-condition which is subsequently blocked by N must not be achieved since this would make some other part of the goal unachievable. □

Theorem 5.1 imply provable intractability also for the unbounded SAS$^+$-PUB and SAS$^+$-PBS plan search problems. Hence, we can find plans in polynomial time exactly for those problems that have polynomially bounded minimal solutions. One should not try to draw any generalized conclusions from this observation, however.

7 Discussion and Conclusions

Recently a number of results have been published on the computational complexity of propositional STRIPS planning under various restrictions [Bylander, 1991, 1992a, Erol *et al.*, 1992]. In addition to this we have previously presented a number of tractable planning problems using the SAS$^+$ formalism [Bäckström and Klein, 1991a, 1991b, Bäckström, 1992a, 1992b]. All of these results concerns the complexity of planning in various restricted versions of certain formalisms. One might also investigate the complexity of planning for specific problems instead of specific formalisms. This has been done for some variants of the blocks-world problem by Gupta and Nau [1992]. Furthermore, the complexity of temporal projection and its relationship to planning has been investigated by Dean and Boddy [1988] and by ourselves [Bäckström and Nebel, 1992; Nebel and Bäckström, 1993].

Our previous publications on SAS$^+$ planning have concentrated on finding tractable subproblems and trying to extend these while retaining tractability. This paper complements these results by presenting a complete investigation of the complexity for each of the possible combinations of the previously considered restrictions. We already know [Bäckström, 1992a, 1992b] that the SAS$^+$ formalism is expressively equivalent to a number of 'standard' propositional STRIPS formalisms, including those analysed by Bylander [1991] and Erol *et al.* [1992]. One might wonder, then, why we have bothered doing such a complexity analysis for the SAS$^+$ formalism. However, there are at least two important differences between our analysis and the previous ones.

Firstly, Bylander and Erol *et al.* have only studied *local* restrictions on operators. Unariness is such a local restriction, while post-uniqueness and single-valuedness are *global* restrictions on the whole set of operators. Korf [1987] has studied some computationally interesting global properties, like independent and serializable subgoals. Unfortunately, finding out whether a problem instance has serializable subgoals is PSPACE-complete [Bylander, 1992b]. In contrast to this, all our restrictions can be tested in low-order polynomial time [Bäckström, 1992a, Theorem 4.8]. Furthermore, we have not derived our restrictions from the formalism *per se* but from studying a test example in the area of automatic control. Using the SAS^+ formalism instead of the STRIPS formalism has facilitated in finding these restrictions, which are interesting from a computational point of view by resulting in tractability. One should also note that the equivalence result applies to the unrestricted versions of the formalisms and it is not guaranteed to hold under arbitrary restrictions. Furthermore, it is not always obvious how to translate a restriction for SAS^+ into a restriction for STRIPS and *vice versa*.

The second difference is that the previous analyses of planning complexity have only considered the *plan existence problem* [Bylander, 1991], and sometimes also the *bounded plan existence problem* [Bylander, 1993, Erol *et al.*, 1992]. In addition to this, we also analyse the complexity of actually *finding* a (possibly optimal) plan, which is what we are ultimately interested in. In many cases there is a strong relationship between a decision (solution existence) problem and its corresponding search problem (finding a solution), but it is not always so.

The result of our analysis shows that we have reached the *tractability borderline* and we cannot continue to remove restrictions and still have tractability. This should not discourage us, however. It means that we have will have to start considering alternative restrictions or less restricted variants of the P, U, B and S restrictions. The proposed research methodology is further described in Bdckstrvm [1992a] where also some suggestions for such alternative restrictions are given. Finally, it seems that one important reason for the disappointing complexity results is that most formalisms allow formulating unrealistic problems, having exponentially sized minimal solutions.

Acknowledgements

Bart Selman and the anonymous reviewers have provided helpful comments on the paper.

References

[AAAI, 1992] *Proc. 10th (US) Nat'l Conf. on Artif Intell. (AAAI-92)*, San José, CA, USA, July 1992.

[Bäckström and Klein, 1991a] Christer Bäckström and Inger Klein. Parallel non-binary planning in polynomial time. In IJCAI [1991], pages 268–273.

[Bäckström and Klein, 1991b] Christer Bäckström and Inger Klein. Planning in polynomial time: The SAS-PUBS class. *Comput. Intell.*, 7(3):181–197, 1991.

[Bäckström and Nebel, 1992] Christer Bäckström and Bernhard Nebel. On the computational complexity of planning and story understanding. In *Proc. 10th Eur. Conf. on Artif. Intell. (ECAI-92)*, pages 349–353, Vienna, Austria, August 1992.

[Bäckström and Nebel, 1993] Christer Bäckström and Bernhard Nebel. Complexity results for SAS^+ planning. Research report, Department of Computer and Information Science, Linköping University, 1993.

[Bäckström, 1992a] Christer Bäckström. *Computational Complexity of Reasoning about Plans.* Doctoral dissertation, Linköping University, Linköping, Sweden, June 1992.

[Bäckström, 1992b] Christer Bäckström. Equivalence and tractability results for SAS^+ planning. In *Proc. 3rd Int'l Conf. on Principles of Knowledge Repr. and Reasoning (KR-92)*, pages 126–137, Cambridge, MA, USA, October 1992.

[Balcázar *et al.*, 1988] José Luis Balcázar, Josep Diaz, and Joaquim Gabarró. *Structural Complexity I.* Springer, 1988.

[Bylander, 1991] Tom Bylander. Complexity results for planning. In IJCAI [1991], pages 274–279.

[Bylander, 1992a] Tom Bylander. Complexity results for extended planning. In *Proc. 1st Int'l Conf. on Artif. Intell. Planning Syst. (AIPS-92)*, pages 20–27, College Park, MD, USA, June 1992.

[Bylander, 1992b] Tom Bylander. Complexity results for serial decomposability. In AAAI [1992], pages 729–734.

[Bylander, 1993] Tom Bylander. The computational complexity of propositional STRIPS planning. *Artif. Intell.* To appear.

[Dean and Boddy, 1988] Thomas Dean and Mark Boddy. Reasoning about partially ordered events. *Artif. Intell.*, 36:375–399, 1988.

[Erol *et al.*, 1992] Kutluhan Erol, Dana S Nau, and V S Subrahmanian. On the complexity of domain-independent planning. In AAAI [1992], pages 381–386.

[Garey and Johnson, 1979] Michael Garey and David Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* Freeman, 1979.

[Gupta and Nau, 1992] Naresh Gupta and Dana S Nau. On the complexity of blocks-world planning. *Artif. Intell.*, 56:223–254, 1992.

[IJCAI, 1991] *Proc 12th Int'l Joint Conf. on Artif. Intell. (IJCAI-91)*, Sydney, Australia, August 1991.

[Korf, 1987] Richard E Korf. Planning as search: A quantitative approach. *Artif. Intell.*, 33:65–88, 1987.

[Nebel and Bäckström, 1993] Bernhard Nebel and Christer Bäckström. On the computational complexity of temporal projection, planning and plan validation. *Artif. Intell.* To appear.