

Oversubscription Planning: Complexity and Compilability

Meysam Aghighi and Peter Jonsson

Department of Computer and Information Science

Linköping University

Linköping, Sweden

{meysam.aghighi, peter.jonsson} at liu.se

Abstract

Many real-world planning problems are oversubscription problems where all goals are not simultaneously achievable and the planner needs to find a feasible subset. We present complexity results for the so-called *partial satisfaction* and *net benefit* problems under various restrictions; this extends previous work by van den Briel et al. Our results reveal strong connections between these problems and with classical planning. We also present a method for efficiently compiling oversubscription problems into the ordinary plan existence problem; this can be viewed as a continuation of earlier work by Keyder & Geffner.

1 Introduction

Classical propositional planning is the problem of finding a sequence of operators that achieves a set of goals from a given initial state. An important feature of this problem is that a solution plan must achieve all goals simultaneously. Unfortunately, this is not possible in many real-world problems. For instance, Smith (2004) notes that many NASA planning problems have a large number of possible goals and the planning system has to find a feasible subset of the goals. This kind of planning is known as *oversubscription planning*. Ordinary planning systems cannot handle oversubscription planning and, in response to this, several custom-made planners and heuristics have been suggested, cf. (Benton, Do, and Kambhampati 2009; Mirkis and Domshlak 2013; van den Briel et al. 2004). Compared to classical planning, it is fair to say that the algorithmics of oversubscription planning is not very well-understood and the number of available planners is quite limited. It is also fair to say that the computational complexity of oversubscription planning is not very well-understood either: besides the fact that oversubscription planning is PSPACE-complete (van den Briel et al. 2004), very little is known. The aim of this paper is two-fold:

1. to study the computational complexity of oversubscription planning under various restrictions, and
2. to present a new way of compiling oversubscription planning into classical planning.

We study two different problems: the *partial satisfaction problem* (PSP) where the number of achieved goals is to be maximized, and the *net benefit problem* (NBP) where the total weight of the achieved goals minus the cost of the plan is to be maximized. These problems are analyzed in Sections 3 and 4, respectively, and the results are summarized in Section 5. Our complexity analysis considers two types of restrictions: syntactically restricted pre- and postconditions (Bylander 1994) and the P,U,B,S restrictions (Bäckström and Klein 1991). We concentrate on these restrictions since they (or slight variations of them) are quite popular when studying different aspects of planning complexity, cf. (Bäckström et al. 2012; Giménez and Jonsson 2012; Katz and Domshlak 2008).

A small number of cases are left open in both cases and this reflects that we do not have a full understanding of the complexity of the plan existence problem (PE) under neither Bylander's nor the P,U,B,S restrictions. Despite this, there are several observations to be made. One important observation is that, in many cases, oversubscription planning is not substantially harder than classical planning. For instance, if the PE problem for some set of instances X (under mild additional assumptions) is NP-complete, then PSP for X is NP-complete, too. Another observation is that (in the cases where we can exactly pinpoint the complexity), PSP and NBP have the same complexity. This is, of course, not always the case and we give an example of instances that strongly separates the two problems. However, our results indicate that PSP, NBP, and classical planning are more closely related than one may initially suspect.

We demonstrate this close relationship in a different way in Section 6. We present a polynomial-time reduction from the decision version of NBP to PE such that the number of variables of the resulting instance is slowly growing in the size of the original instance or, put differently, the number of nodes in the corresponding search space increases only moderately. Such a reduction is interesting since algorithms for PE can be used for solving NBP problems with limited slowdown. One should additionally note that PE, cost-optimal planning, and PSP can be trivially reduced to NBP. Thus, these four problems are tightly connected, and algorithms for PE, i.e. ordinary classical planners, may be a viable alternative for performing both oversubscription planning and cost-optimal planning. This result is inspired by Keyder &

Geffner (2009) who have presented a method for compiling the optimization version of NBP into cost-optimal planning.

2 Planning Framework

We use the SAS⁺ planning framework (Bäckström and Nebel 1995) as our basic formalism. A SAS⁺ planning instance is a tuple $\Pi = (V, A, I, G)$ where $V = \{v_1, \dots, v_n\}$ is the set of *variables* over a finite *domain* \mathcal{D} . We add a value \mathbf{u} to \mathcal{D} (where \mathbf{u} stands for *undefined*) resulting in the set \mathcal{D}_+ . \mathcal{D}^n is the set of *total states* and \mathcal{D}_+^n is the set of *partial states*. The value of a variable v in a state $s \in \mathcal{D}_+^n$ is denoted as $s[v]$, and we let $\langle s \rangle = |\{v \in V \mid s[v] \neq \mathbf{u}\}|$. $I \in \mathcal{D}^n$ is the *initial state* and $G \in \mathcal{D}_+^n$ is the *goal state*. A is the set of *actions* where every action $a \in A$ has a *precondition* $\mathbf{pre}(a) \in \mathcal{D}_+^n$ and a *postcondition* $\mathbf{post}(a) \in \mathcal{D}_+^n$. For two states s_1, s_2 , we write $s_1 \sqsubseteq s_2$ if and only if for all $v \in V$, either $s_1[v] = \mathbf{u}$ or $s_1[v] = s_2[v]$. An action a is *applicable* in a state $s \in \mathcal{D}^n$ if and only if $\mathbf{pre}(a) \sqsubseteq s$. The *result of a in s* , if a be applicable in s , is a state $t \in \mathcal{D}^n$ such that for all $v \in V$, $t[v] = s[v]$ if $\mathbf{post}(a)[v] = \mathbf{u}$, otherwise $t[v] = \mathbf{post}(a)[v]$. We say action a *affects* variable v in state s , if $s[v] \neq t[v]$ where t is the result of a in s .

Given two states $I \in \mathcal{D}^n$ and $G \in \mathcal{D}_+^n$, a sequence of actions $\omega = (a_1, \dots, a_m)$ is called a *plan* from I to G if and only if there exists a sequence of total states (s_1, \dots, s_{m-1}) such that s_1 is the result of a_1 in I , s_i is the result of a_i in s_{i-1} for all $2 \leq i \leq m-1$, and $G \sqsubseteq s_G$ where s_G is the result of a_m in s_{m-1} .

Let Θ be an arbitrary set of SAS⁺ instances. The SAS⁺ *plan existence* problem $\text{PE}(\Theta)$ has the following definition:

INSTANCE: A SAS⁺ instance $\Pi = (V, A, I, G) \in \Theta$.

QUESTION: Does Π have a solution, i.e. a plan from I to G ?

We will also consider the *bounded cost plan existence* problem ($\text{BCPE}(\Theta)$):

INSTANCE: A tuple $\Pi = (V, A, I, G, c, K)$ where $(V, A, I, G) \in \Theta$, c is a function assigning a non-negative integer weight to each $a \in A$, and K is an integer.

QUESTION: Does Π have a solution (a_1, \dots, a_n) such that $\sum_{i=1}^n c(a_i) \leq K$?

For historical reasons (but also notational convenience), we will sometimes use a different notation for restricted SAS⁺ instances. Consider the following restrictions (Bäckström and Klein 1991) on instances (V, A, I, G) :

P: (Post-unique) for every $v \in V$ and every $d \in \mathcal{D}$, there is at most one $a \in A$ such that $\mathbf{post}(a)[v] = d$.

U: (Unary) for every $a \in A$, $\langle \mathbf{post}(a) \rangle = 1$.

B: (Binary) $|\mathcal{D}| = 2$.

S: (Single-valued) for every $v \in V$ and every $a, b \in A$, if $\mathbf{pre}(a)[v] \neq \mathbf{u}$, $\mathbf{pre}(b)[v] \neq \mathbf{u}$ and $\mathbf{post}(a)[v] = \mathbf{post}(b)[v] = \mathbf{u}$, then $\mathbf{pre}(a)[v] = \mathbf{pre}(b)[v]$.

We write PE-R to denote the PE problem restricted to instances satisfying the restrictions in R , i.e. PE-PU means $\text{PE}(\Theta)$ where Θ contains all instances satisfying restrictions P and U . We also study another type of restrictions (introduced by Bylander (1994)) which is based on restricting the sign and number of preconditions and postconditions of actions (the sign restriction is only discussed when we already

have restriction B). For example, by $\text{SAS}^+ \text{-B}_{2+}^1$, we mean PE for instances having $|\mathcal{D}| = 2$ and allowing at most one precondition and two positive postconditions per action.

When dealing with two-valued domains, we always assume that $\mathcal{D} = \{0, 1\}$ and we use a simplified way of defining actions. We write \bar{x} to denote that variable x has (or is assigned) value 0. Then we may simply write $x, \bar{y} \rightarrow z$ when referring to the action having preconditions $x = 1, y = 0$ and postcondition $z = 1$. If an action has no preconditions, then this is indicated with the symbol \emptyset . Finally, we let $\bar{Y} = \{\bar{y} \mid y \in Y\}$.

3 The Partial Satisfaction Problem

Let Θ denote a set of SAS⁺ instances. The *partial satisfaction problem* $\text{PSP}(\Theta)$ has the following definition:

INSTANCE: A tuple (V, A, I, G, K) where $(V, A, I, G) \in \Theta$ and K is a positive integer.

QUESTION: Is there a state $G' \sqsubseteq G$ such that $\langle G' \rangle \geq K$ and (V, A, I, G') has a solution?

We define the PSP problem as a decision problem which allow us to simplify the forthcoming proofs. Viewing it as an optimization problem instead does not affect the complexity substantially: a decision problem in P will have a corresponding optimization problem in FP (the functional analogue of P), an NP-complete decision problem will have an optimization problem in FP^{NP} (by using binary search), and a PSPACE-complete decision problem will have an optimization problem in FPSPACE . Furthermore, we use integer values instead of rational values. Rational values can easily be replaced by integers via multiplication with suitable factors, and this reformulation leads to an equivalent instance whose size is only marginally larger. These observations also apply to the NET BENEFIT problems that we will introduce later on.

Note that $\text{PE}(\Theta)$ can be viewed as an instance of $\text{PSP}(\Theta)$ (simply by setting $K = \langle G \rangle$) and recall that $\text{PSP}(\Theta)$ is in PSPACE (van den Briel et al. 2004). We say that Θ is *closed under goal substitution* if for arbitrary $(V, A, I, G) \in \Theta$, all instances (V, A, I, G') such that $G' \in \mathcal{D}_+^{|\mathcal{V}|}$ are members of Θ . Clearly, the sets of instances satisfying the P,U,B,S and/or Bylander's restrictions are closed under goal substitution.

Lemma 1. *Let Θ be a set of SAS⁺ instances that is closed under goal substitution. If $\text{PE}(\Theta) \in \text{NP}$, then $\text{PSP}(\Theta) \in \text{NP}$.*

Proof. Let $\Pi' = (V, A, I, G, K)$ be an arbitrary instance of $\text{PSP}(\Theta)$. Nondeterministically guess a state $G' \sqsubseteq G$ such that $\langle G' \rangle \geq K$. We know that (V, A, I, G') is an instance of $\text{PE}(\Theta)$, too. Hence, we can nondeterministically guess a polynomially bounded certificate X showing that (V, A, I, G') is solvable. It follows that (X, G') is a polynomial-time verifiable certificate for Π' . \square

Thus, there is a close connection between $\text{PE}(\Theta)$ and $\text{PSP}(\Theta)$: if Θ is closed under goal substitution and $\text{PE}(\Theta)$ is NP- or PSPACE-complete, then $\text{PSP}(\Theta)$ is NP- or PSPACE-complete, respectively. We can now concentrate on instance sets Θ such that $\text{PE}(\Theta)$ is solvable in polynomial time.

Theorem 1. *PSP-B_{2+}^0 and PSP-B_{1+}^1 are NP-hard.*

Proof. We begin with $\text{PSP-}B_{2+}^0$. The proof is by a polynomial-time reduction from the NP-complete problem VERTEX COVER, (Garey and Johnson 1979, GT1):

INSTANCE: Graph $G = (\mathcal{V}, E)$ and integer $K \leq |\mathcal{V}|$.

QUESTION: Is there a vertex cover of size K or less for G , i.e., a subset $\mathcal{V}' \subseteq \mathcal{V}$ with $|\mathcal{V}'| \leq K$ such that for each edge $\{u, v\} \in E$ at least one of u and v belongs to \mathcal{V}' ?

Assume that we have an instance (\mathcal{V}, E, K) of VERTEX COVER where $\mathcal{V} = \{v_1, \dots, v_n\}$, $E = \{e_1, \dots, e_m\} \subseteq \mathcal{V}^2$, and $0 \leq K \leq |\mathcal{V}|$. Define a $\text{PSP-}B_{2+}^0$ instance $\Pi = (V, A, I, G, K')$ as follows: $V = \{v_1, \dots, v_n\} \cup \{e_j^l \mid 1 \leq j \leq m, 0 \leq l \leq 1\}$, $I = (0, \dots, 0)$, $K' = 2m + n - K$, and $G[v_i] = 0$ for every $1 \leq i \leq n$. Furthermore, for every $0 \leq l \leq 1$ and every $1 \leq j \leq m$ such that $e_j = (u, w) \in E$, let $G[e_j^l] = 1$ and let A contain the actions $a_j^l : \emptyset \rightarrow e_j^l$, u and $b_j^l : \emptyset \rightarrow e_j^l, w$.

Assume Π has a solution ω and, additionally, assume that ω achieves the highest possible number of goals. Let t and s be the number of v_i and e_j^l variables that are equal to 1, respectively, in the state s_G resulting from applying ω to I . We first show that $s_G[e_j^l] = 1$ for all i, j and, consequently, that $s = 2m$. Assume to the contrary that $s_G[e_j^l] = 0$ for some l, j . If $s_G[e_j^{1-l}] = 1$, then we can set e_j^l to 1, too, by using either the action a_j^l or b_j^l . Note that this can be done without assigning the value 1 to any additional v_i variables. This new plan achieves a strictly higher number of goals which contradicts the choice of ω .

If $s_G[e_j^0] = s_G[e_j^1] = 0$, then we can set both of them to 1 by using actions a_j^0 and a_j^1 . This gives us two new satisfied goals and, possibly, one less v_i variable satisfying its goal. All in all, this new plan achieves a strictly higher number of goals which once again contradicts the choice of ω . Hence, all e_j^l variables can be assumed to have value 1 and $s = 2m$. Finally, note that $n - t$ is the number of v_i variables that are given the value 0 and, thus, contribute to the number of satisfied goals. It follows that $s + n - t \geq K' \Rightarrow 2m + n - t \geq 2m + n - K \Leftrightarrow t \leq K$. Consequently, (\mathcal{V}, E, K) has a solution.

If (\mathcal{V}, E, K) has a solution $\mathcal{V}' \subseteq \mathcal{V}$, then let $A' = \{a_1, \dots, a_p\}$ contain the actions $a \in A$ satisfying $\text{post}(a)[v'] \neq \mathbf{u}$ for some $v' \in \mathcal{V}'$. Let $\omega = (a_1, \dots, a_p)$ and note that ω is applicable in I . Furthermore, it will achieve at least $2m + n - K$ goals.

The proof for $\text{PSP-}B_{1+}^1$ is virtually identical: the only major difference is that we replace actions of the type $a_j^l : \emptyset \rightarrow e_j^l, u$ with the two actions $\emptyset \rightarrow u$ and $u \rightarrow e_j^l$, and we do an analogous replacement for b_j^l actions. \square

Theorem 2. PSP-PUBS_+ is NP-hard.

Proof. Proof by reduction from the NP-complete problem INDEPENDENT SET, (Garey and Johnson 1979, GT20):

INSTANCE: Graph $G = (V, E)$ and integer $K \leq |V|$.

QUESTION: Does G contain an independent set of size K or more? i.e., a subset $V' \subseteq V$ such that $|V'| \geq K$ and such that no two vertices in V' are joined by an edge in E .

Assume that (\mathcal{V}, E, K) is an instance of INDEPENDENT SET where $\mathcal{V} = \{v_1, \dots, v_n\}$. We define an instance of PSP-PUBS_+ $\Pi = (V, A, I, G, K)$ as follows: $V = \{v_1, \dots, v_n\}$, $A = \{a_1, \dots, a_n\}$, $I = (0, \dots, 0)$, $G = (1, \dots, 1)$, and for every $v_i \in V$ we define $a_i : \bar{v}_{i_1}, \dots, \bar{v}_{i_t} \rightarrow v_i$ where v_{i_1}, \dots, v_{i_t} are the neighbours of v_i . It is straightforward to verify that this instance is an instance of PSP-PUBS_+ .

Let $\{v_{i_1}, \dots, v_{i_p}\} \subseteq \mathcal{V}$ be a solution to (\mathcal{V}, E, K) . We immediately see that $(a_{i_1}, \dots, a_{i_p})$ is a solution to Π ; merely note that once a vertex v_i has been selected (by inserting a_i into the plan), all of its neighbours are 'blocked' from further consideration by the choice of preconditions. Similarly, if Π has a solution $(a_{i_1}, \dots, a_{i_p})$, then $\{v_{i_1}, \dots, v_{i_p}\}$ is a solution to (\mathcal{V}, E, K) . \square

4 The Net Benefit Problem

Given a goal state $G \in \mathcal{D}_+^n$ and a utility vector $U \in \mathbb{N}_0^n$ (where \mathbb{N}_0 denotes the set of non-negative integers), we let $m_{G,U}(S) = \sum \{U[i] \mid S[i] = G[i] \neq \mathbf{u}\}$, i.e. $m_{G,U}(S)$ denotes the total utility of a state S given a goal state G and the utilities of the components of G . Let Θ denote a set of SAS^+ instances. The *net benefit problem* $\text{NBP}(\Theta)$ has the following definition:

INSTANCE: A tuple (V, A, I, G, c, U, K) where $(V, A, I, G) \in \Theta$, c is a function from A to \mathbb{N}_0 , $U \in \mathbb{N}_0^n$, and K is a positive integer.

QUESTION: Is there a plan $p = (a_1, \dots, a_t)$ starting from I and leading to a state S such that $m_{G,U}(S) - \sum_{i=1}^t c(a_i) \geq K$?

The value $m_{G,U}(S) - \sum_{i=1}^t c(a_i)$ is called the *net benefit* of the plan p . Note that NBP always has a solution with net benefit ≥ 0 : the empty plan. $\text{PSP}(\Theta)$ is trivially polynomial-time reducible to $\text{NBP}(\Theta)$ by letting the action cost function c always return 0 and choosing $U = \{1\}^n$. We next prove that there is a connection between BCPE and NBP that is analogous to the connection between PE and PSP established in Lemma 1.

Lemma 2. Let Θ be a set of SAS^+ instances that is closed under goal substitution. If $\text{BCPE}(\Theta) \in \text{NP}$, then $\text{NBP}(\Theta) \in \text{NP}$.

Proof. Let $\Pi = (V, A, I, G, c, U, K)$ be an arbitrary instance of $\text{NBP}(\Theta)$. Nondeterministically guess

- (1) two numbers $v, \gamma \in \mathbb{N}_0$ such that $v - \gamma \geq K$,
- (2) a state S such that $m_{G,U}(S) \geq v$, and
- (3) a certificate X showing that there exists a plan $\omega = (a_1, \dots, a_n)$ for (V, A, I, S) with $\gamma \geq \sum_{i=1}^n c(a_i)$.

X exists for some guess of S and can be verified in polynomial time since $\text{BCPE}(\Theta)$ is in NP. Hence, (v, γ, S, X) exists if and only if Π has a solution, and the certificate (v, γ, S, X) can be verified in polynomial time. \square

We are now ready to prove the necessary complexity results for NBP. We begin with a tractability result.

Theorem 3. NBP_1^0 is in P.

Proof. Let $\Pi = (V, A, I, G, c, U, K)$ be an arbitrary instance of NBP_1^0 . Let A' contain those actions that achieves

some component of the goal G , and if there are multiple actions for the same component, then pick one action with the lowest cost according to function c . Finally, let $\{a_1, \dots, a_n\}$ denote the actions $a \in A'$ satisfying $u - c(a) > 0$ where u is the utility of the goal achieved by a . Let S be the state resulting from applying (a_1, \dots, a_n) to I . It is clear that Π has a solution if and only if $m_{G,U}(S) - \sum_{i=1}^n c(a_i) \geq K$. \square

Next, we prove that certain NBP problems are members of NP by showing that the corresponding BCPE problems are in NP and using Lemma 2. The complexity of finding shortest solutions for these problems is well-studied (Bäckström and Nebel 1995; Bylander 1994) but we need results for arbitrarily weighted actions. Given a solvable instance $\Pi = (V, A, I, G, K)$ of BCPE, let $S(\Pi)$ contain the solutions to Π with lowest possible cost and let $S'(\Pi)$ contain the members of $S(\Pi)$ having minimum length.

Theorem 4. NBP^0 is in NP.

Proof. We show that BCPE^0 is in NP and use Lemma 2. Let $\Pi = (V, A, I, G, c, K)$ be an arbitrary solvable instance of BCPE^0 and arbitrarily choose $\omega \in S'(\Pi)$. We claim that every action in ω occurs at most once and, consequently, that $|\omega| \leq |A|$ and BCPE^0 is NP since we can simply list the actions in the plan. Assume to the contrary that there is an action a that occurs in ω more than once. Construct a new plan ω' by deleting all occurrences of a except the last one. Clearly, ω' is still a plan from I to G : the actions in A have no preconditions so the removal of a will not affect the applicability of other actions. If $c(a) > 0$, then the total cost of ω' is strictly lower than the total cost of ω which leads to a contradiction. If $c(a) = 0$, then ω' and ω have the same cost but $|\omega'| < |\omega|$ which leads to a contradiction, too. \square

Theorem 5. $\text{NBP-}B_+$ is in NP.

Proof. We show that $\text{BCPE-}B_+$ is in NP and use Lemma 2. Let $\Pi = (V, A, I, G, c, K)$ be an arbitrary solvable instance of $\text{BCPE-}B_+$ and arbitrarily choose $\omega \in S'(\Pi)$. Each action in ω changes at least one variable from 0 to 1 and no action changes it back. Hence, $|\omega| \leq |V|$ and we are done. \square

Theorem 6. NBP-US and $\text{NBP-}B_1^+$ are in NP.

Proof. We show that NBP-US is in NP; this immediately implies that $\text{NBP-}B_1^+$ is in NP, too. By Lemma 2, it is sufficient to prove that BCPE-US is in NP. Let $\Pi = (V, A, I, G, c, K)$ be an arbitrary solvable instance of BCPE-US and arbitrarily choose $\omega = (a_1, \dots, a_t) \in S'(\Pi)$. We claim that every action appears at most twice in ω and that $|\omega| \leq 2|A|$. For every $v \in V$, let $\omega_v = (a_1, \dots, a_k)$ denote the subsequence of ω containing those actions changing variable v , and let $X_v = (x_0, x_1, x_2, \dots, x_k)$ be defined such that $x_0 = I[v]$ and $\text{pre}(a_i)[v] = x_{i-1}$, $\text{post}(a_i)[v] = x_i$ for $1 \leq i \leq k$. Since Π satisfies restriction S, we know that for every $v \in V$ and every $a, b \in A$, if $\text{pre}(a)[v] \neq \mathbf{u}$, $\text{pre}(b)[v] \neq \mathbf{u}$ and $\text{post}(a)[v] = \text{post}(b)[v] = \mathbf{u}$, then $\text{pre}(a)[v] = \text{pre}(b)[v]$. Let y_v denote this unique ‘prevail’ value for v if it exists and let $y_v = \mathbf{u}$ otherwise. Arbitrarily choose $v \in V$ and consider the following three cases:

None of the values in X_v equals y_v . We claim that no action appears more than one time in ω_v . Assume to the contrary that for some $i < j$, the action a_i is the same as action a_j . Then, the subsequence $(a_i, a_{i+1}, \dots, a_{j-1})$ produces v values that are not needed in the precondition of any other actions or the goal. Recall that these actions *only* affect variable v (due to restriction U) so we can simply remove them from ω_v . This results in a plan with either lower cost or shorter length than ω which leads to a contradiction.

At least two of the values in X_v equal y_v . If $x_i = x_j = y_v$ for some $i < j$, then we can delete actions $(a_{i+1}, \dots, a_{j-1}, a_j)$ from ω_v — the reason behind this is similar to the previous case. Hence, this cannot occur.

Exactly one value in X_v equals y_v . Assume that $x_i = y_v$ and divide ω_v into $\omega_v^1 = (a_1, \dots, a_{i-1})$ and $\omega_v^2 = (a_{i+1}, \dots, a_k)$. In each part, every action can appear at most once since, otherwise, the actions between the two occurrences can be deleted (once again in a fashion similar to the first and second cases) and this results in a cheaper and/or shorter plan. This leads to a contradiction so every action appears at most two times in ω_v .

By restriction U, every action appears in at most one of the ω_v sequences which implies, by the three cases above, that each action in A appears at most two times in ω . \square

5 Summary of Complexity Results

We will now summarize the complexity results for PSP and NBP. To do so, we need a few more hardness results.

Theorem 7. (Bäckström and Nebel 1995; Bylander 1994) $\text{PE-}B^{1+}$, $\text{PE-}B_1$, $\text{PE-}B_2^{2+}$, PE-UB , and PE-BS are PSPACE-complete problems.

Bylander (1994) does not explicitly state that $\text{PE-}B^{1+}$ is PSPACE-complete but it is a direct consequence of his PSPACE-completeness proof for $\text{PE-}B^1$. The complexity results for PSP and NBP under Bylander’s restrictions can be found in Tables 1–4. The “*” symbol means that there is no restriction on the number of pre- or postconditions while ‘ ≥ 2 ’ implies that the result holds for any fixed number ≥ 2 . The tables also contain information about the PE problem from (Bäckström and Nebel 1995; Bylander 1994): the symbol ‘†’ indicates when we know for sure (under the assumption $\text{P} \neq \text{NP}$) that the complexity of PE differs from the complexity of PSP and NBP. The difference is always the same, namely, the PE problem is in P while the oversubscription problem is NP-complete. One sees that the complexity of PE and oversubscription planning only differs in a small number of ‘borderline’ cases.

The results can be inferred as follows: both PSP and NBP problem are members of PSPACE (van den Briel et al. 2004). PSPACE-hardness follows from Theorem 7 combined with the fact that $\text{PE}(\Theta)$ polynomial-time reduces to $\text{PSP}(\Theta)$ and $\text{NBP}(\Theta)$ for all Θ . NP-hardness follows from the results in Section 3 — recall that if $\text{PSP}(\Gamma)$ is NP-hard, then $\text{NBP}(\Gamma)$ is NP-hard, too. The NP membership results is shown in Section 4 — note that if $\text{NBP}(\Theta)$ is in NP, then $\text{PSP}(\Theta)$ is in NP, too. Finally, the tractability results follow from Theorem 3. Results for the P,U,B,S restrictions are collected in Figure 1.

Table 1: Results for PSP and NBP

		post		
		1	≥ 2	*
pre	0	P	NP-c. †	NP-c. †
	1	NP-h.	NP-h.	PSPACE-c.
	≥ 2	NP-h.	PSPACE-c.	PSPACE-c.
	*	PSPACE-c.	PSPACE-c.	PSPACE-c.

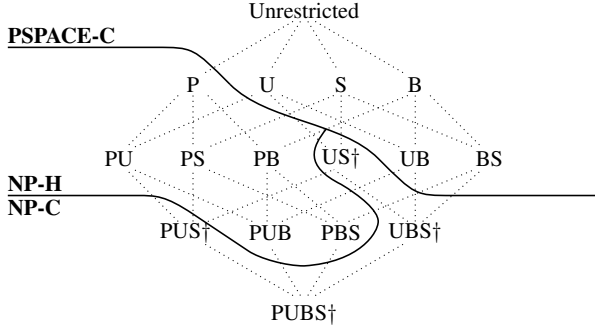


Figure 1: Results for P,U,B,S restrictions

These results and the results in Table 1 hold for arbitrary domains \mathcal{D} satisfying $|\mathcal{D}| \geq 2$. Tables 2–4 contain results for two-valued domains where we restrict ourselves to positive pre- and/or postconditions. Once again, the complexity of PSP and NBP coincide in all cases.

It may be slightly surprising that the PSP and NBP problems have the same complexity but this is inherent in the restrictions we consider. Of course, this is not true for all instance sets: Jonsson (1999, Sec. 4) shows that there is a set Θ of planning instances that (1) always have a solution but (2) finding the shortest solution is PSPACE-hard. (1) implies that $\text{PSP}(\Theta)$ is trivially solvable in polynomial time while (2) implies that $\text{NBP}(\Theta)$ is PSPACE-complete.

6 Compiling NBP into PE

We continue by presenting a method to compile NBP into PE. The basic motivation behind this reduction is to provide a method for using ordinary planners for solving the NBP problem. Our main concern is to make the reduction as lean as possible, i.e. we want the reduction to increase the size of the resulting instance as little as possible. The previously presented complexity results cannot be inferred from

Table 2: Results for + prec. / free postc.

		post		
		1	≥ 2	*
+ pre	0	P	NP-c. †	NP-c. †
	1	NP-c. †	NP-h.	PSPACE-c.
	≥ 2	NP-c. †	PSPACE-c.	PSPACE-c.
	*	NP-c. †	PSPACE-c.	PSPACE-c.

Table 3: Results for free prec. / + postc.

		+ post		
		1	≥ 2	*
pre	0	P	NP-c. †	NP-c. †
	1	NP-c.	NP-c.	NP-c.
	≥ 2	NP-c.	NP-c.	NP-c.
	*	NP-c.	NP-c.	NP-c.

Table 4: Results for + prec. / + postc.

		+ post		
		1	≥ 2	*
+ pre	0	P	NP-c. †	NP-c. †
	1	NP-c. †	NP-c.	NP-c.
	≥ 2	NP-c. †	NP-c.	NP-c.
	*	NP-c. †	NP-c.	NP-c.

this reduction since it is not guaranteed to preserve any of the properties except B.

The method is based on a certain kind of counter which we introduce next. The counter is very important in the reduction since it is used for keeping track of the net benefit during the course of the plan. Given an integer $0 \leq m$, we let $(m_{n-1} \dots m_1 m_0)_2$ denote m written in binary where m_{n-1} is the most significant bit and m_0 the least significant. Given a sequence of binary variables $X = (x_{n-1}, \dots, x_0)$, we can easily use such a sequence to represent the number m : let $x_i = 1$ if $m_i = 1$ and $x_i = 0$, otherwise. Define $(X \simeq m) = \{x_i \mid m_i = 1\} \cup \{\bar{x}_i \mid m_i = 0\}$. For instance, if $X = (x_2, x_1, x_0)$, the action $(X \simeq 3) \rightarrow (X \simeq 5)$ equals $\bar{x}_2, x_1, x_0 \rightarrow x_2, \bar{x}_1, x_0$. Given a state S over variable sequence X , we let $\llbracket S \rrbracket$ denote the number that X represents.

Let $X = (x_{k-1}, \dots, x_0)$ be a sequence of variables. If $0 \leq n < k$ and $\llbracket X \rrbracket < 2^k - 2^n$, then X is transformed into a new state X' such that $\llbracket X' \rrbracket = \llbracket X \rrbracket + 2^n$ by exactly one of the following $k - n$ actions:

$$\begin{aligned}
 a_1^n &: \bar{x}_n \rightarrow x_n \\
 a_2^n &: \bar{x}_{n+1}, x_n \rightarrow x_{n+1}, \bar{x}_n \\
 &\vdots \\
 a_{k-n}^n &: \bar{x}_{k-1}, x_{k-2}, \dots, x_n \rightarrow x_{k-1}, \bar{x}_{k-2}, \dots, \bar{x}_n
 \end{aligned}$$

We use these actions to build a counter with arbitrary step size and the ability to count both upwards and downwards. Arbitrarily choose an integer $0 \leq x < 2^k$ and a step length $0 \leq s < 2^k$. Introduce ‘upward trigger’ variables $C = \{c^i \mid 0 \leq i < k\}$, ‘downward trigger’ variables $D = \{d^i \mid 0 \leq i < k\}$, and for every $0 \leq i < k$ and $1 \leq l \leq k - i$, introduce actions

$$\begin{aligned}
 inc_l^i &: c^i, \mathbf{pre}(a_l^i) \rightarrow \bar{c}^i, \mathbf{post}(a_l^i) \\
 dec_l^i &: d^i, \mathbf{post}(a_l^i) \rightarrow \bar{d}^i, \mathbf{pre}(a_l^i)
 \end{aligned}$$

Consider a planning instance (V, A, I, G) where $V = X \cup C \cup D$, A contains the actions above, I specifies the values of the X variables such that $\llbracket I[X] \rrbracket = x$, the C variables equal $C \simeq s$ and the D variables equal 0. Finally, let $G[v] = \mathbf{u}$ for $v \in X \cup D$ and $G[v] = 0$ for $v \in C$. It is not hard to see that

this instance has a solution and it reaches a state S where $\llbracket S[X] \rrbracket = x + s$. Similarly, if we want to lower X with s steps, we use the downward trigger variables in D instead of C . Note that if the X and D variables are representing numbers x and d such that $x < d$, then the counter cannot set all D variables to 0: this way, underflows can be detected and prevented.

We now use the counter for compiling NBP into PE. Given an instance $I = (V, A, I, G, c, U, K)$ of NBP, we construct an instance of PE where we have a global counter (to keep track of the net benefit) over variables X . Initially, X is loaded with the value $M = \sum_{i=1}^{|V|} U[i]$, i.e. the highest possible net benefit. This is done in order to avoid negative numbers. Each action in the plan decreases the counter with its corresponding weight. If the value goes below 0, then the NBP instance has no solution. After having found a plan with total cost γ , we increase the counter with the total utility v achieved by this plan. At this point, the counter has value $M - \gamma + v$ and we want to check whether this value is larger or equal to $M + K$ (in order to verify that $v - \gamma \geq K$). This is done by allowing ‘free’ decreasing of the counter until reaching the goal value $M + K$.

Construction 1. Let $\Pi = (V, A, I, G, c, U, K)$ be an instance of NBP such that $A = \{a_1, \dots, a_p\}$. Construct a SAS⁺ instance $\Pi' = (V', A', I', G')$ as follows: let $M = \sum_{i=1}^{|V|} U[i]$ and $m = \lceil \log M \rceil + 1$. We will use a counter over variables X, C, D on m bits in the construction. Define $V' = V \cup X \cup C \cup D \cup B \cup E$ where $B = \{b_i \mid 1 \leq i \leq \lceil \log |A| \rceil + 1\}$ are variables used to prevent action interference and $E = \{end_g \mid G[g] \neq \mathbf{u}\}$ are variables used to guarantee that after starting to count the utility of the goals, no other action is applicable. Also, $I'[v] = I[v]$ for $v \in V$, $I'[x_i] = p_i$ for $1 \leq i \leq m$ where $M = (p_m \dots p_1)_2$, and $I'[v] = 0$ otherwise. $G'[x_i] = q_i$ for $1 \leq i \leq m$ where $M + K = (q_m \dots q_1)_2$, and $G'[v] = \mathbf{u}$ otherwise. We define A' as follows: for every $a_i \in A$, extend A' with

$$\begin{aligned} a'_i &: \mathbf{pre}(a_i), \overline{B}, \overline{E} \rightarrow (B \simeq i), (D \simeq c(a_i)) \text{ and} \\ a''_i &: (B \simeq i), \overline{D} \rightarrow \overline{B}, \mathbf{post}(a_i). \end{aligned}$$

For every $v_i \in V$ such that $G[v_i] \neq \mathbf{u}$, extend A' with

$$g'_i : \overline{B}, \overline{end}_{v_i}, \overline{C}, (v_i = G[v_i]) \rightarrow end_{v_i}, (C \simeq U[v_i])$$

Finally, add the following actions to A' :

$$\mathit{freesubtract}_l : \mathbf{post}(a_l^0) \rightarrow \mathbf{pre}(a_l^0), 1 \leq l \leq m.$$

We note that the instance built in Construction 1 can easily be constructed in polynomial-time and that the size of the instance increases slowly compared to the size of the original NBP instance. Since $|X| = |C| = |D| = m$, $|B| = \log |A|$ and $|E| = |G|$, the total number of variables in Π' is $|V'| = |V| + 3m + \log |A| + |G|$. Here, one should recall that $|A| \leq (|\mathcal{D}| + 1)^{|V|} \cdot (|\mathcal{D}| + 1)^{|V|}$, implying that $\log |A| \leq 2|V| \log (|\mathcal{D}| + 1)$, and that m is the *logarithm* of the sum of the utility vector U .

Theorem 8. Construction 1 is a reduction from NBP to PE.

Proof. Let Π be an instance of NBP and let Π' be the PE instance obtained via Construction 1. Assume Π has a solution

$\omega = (a_1, \dots, a_r)$ and that it satisfies the goals for variables v_1, \dots, v_s (for simplicity). It can be seen that

$$\omega' = (a'_1, \Delta_1, a''_1, a'_2, \Delta_2, a''_2, \dots, a'_r, \Delta_r, a''_r, g'_1, \nabla_1, g'_2, \nabla_2, \dots, g'_s, \nabla_s, \Delta)$$

is a solution for Π' where Δ_i is a sequence of actions decreasing the counter initiated by a'_i , ∇_j is a sequence of actions increasing the counter initiated by g'_j , and Δ is a sequence of *freesubtract* actions that decrease the counter by net benefit of ω minus K . Since ω is a solution for Π with net benefit at least K , ω' is applicable and the counter value right before applying Δ actions is at least $M + K$. Note that the minimum and maximum value of the counter are always kept in the interval $[0, 2M]$ that the counter covers.

The other direction is similar and thus omitted. \square

7 Discussion

In the first part of this paper, we presented complexity results for the PSP and NBP problems. Clearly, there are many complexity issues in connection with oversubscription planning that are worth studying. One example is to consider instances with restricted *causal graphs*. Classical planning problems for such instances have been intensively studied (Brafman and Domshlak 2003; Chen and Giménez 2010; Katz and Domshlak 2008). In particular, it is interesting to see that the structure of the causal graph can be exploited to identify tractable BCPE problems (Katz and Domshlak 2008). This suggests that tractability results for NBP may be obtained this way, too. Another relevant topic is to study the parameterized complexity of oversubscription planning. Lately, many parameterized complexity results for planning have appeared (Bäckström et al. 2012; de Haan, Roubícková, and Szeider 2013; Kronegger, Pfandler, and Pichler 2013)

In the second part of the paper, we presented a way of compiling NBP into PE. Both compiling different planning problems into each other (Keyder and Geffner 2009; Palacios and Geffner 2009; Taig and Brafman 2013) and compiling planning into other problems (Cashmore, Fox, and Giunchiglia 2012; van den Briel and Kambhampati 2005; Kautz and Selman 1992) have been very active research areas for quite some time. It is both exciting and slightly surprising to see that competitive planning algorithms result from compilation: for instance, Keyder & Geffner (2009) report encouraging results for benchmark oversubscription problems from the 2008 IPC competition. Experimental evaluation of our compilation method is an obvious direction for future research. In particular, using it to solve cost-optimal planning with the aid of ordinary, non-optimizing planners is an interesting possibility. One should also note that the compilation method can be extended in different directions: for instance, modifying it to handle negative utilities is straightforward.

Acknowledgements

Meysam Aghighi is partially supported by the *National Graduate School in Computer Science (CUGS)*, Sweden.

References

- Bäckström, C., and Klein, I. 1991. Planning in polynomial time: the SAS-PUBS class. *Computational Intelligence* 7(3):181–197.
- Bäckström, C., and Nebel, B. 1995. Complexity results for SAS+ planning. *Computational Intelligence* 11(4):625–655.
- Bäckström, C.; Chen, Y.; Jonsson, P.; Ordyniak, S.; and Szeider, S. 2012. The complexity of planning revisited—a parameterized analysis. In *Proc. of the 26th AAAI Conference on Artificial Intelligence (AAAI-2012)*.
- Benton, J.; Do, M.; and Kambhampati, S. 2009. Anytime heuristic search for partial satisfaction planning. *Artif. Intell.* 173(5-6):562–592.
- Brafman, R. I., and Domshlak, C. 2003. Structure and complexity in planning with unary operators. *J. Artif. Intell. Res.* 18:315–349.
- Bylander, T. 1994. The computational complexity of propositional strips planning. *Artif. Intell.* 69(1):165–204.
- Cashmore, M.; Fox, M.; and Giunchiglia, E. 2012. Planning as quantified boolean formula. In *Proc. 20th European Conference on Artificial Intelligence (ECAI-2012)*, 217–222.
- Chen, H., and Giménez, O. 2010. Causal graphs and structurally restricted planning. *J. Comput. Syst. Sci.* 76(7):579–592.
- de Haan, R.; Roubícková, A.; and Szeider, S. 2013. Parameterized complexity results for plan reuse. In *Proc. of the 27th AAAI Conference on Artificial Intelligence (AAAI-2013)*.
- Garey, M. R., and Johnson, D. S. 1979. *Computers and intractability*, volume 174. Freeman New York.
- Giménez, O., and Jonsson, A. 2012. The influence of k -dependence on the complexity of planning. *Artif. Intell.* 177-179:25–45.
- Jonsson, P. 1999. Strong bounds on the approximability of two PSPACE-hard problems in propositional planning. *Ann. Math. Artif. Intell.* 26(1-4):133–147.
- Katz, M., and Domshlak, C. 2008. New islands of tractability of cost-optimal planning. *J. Artif. Intell. Res.* 32:203–288.
- Kautz, H. A., and Selman, B. 1992. Planning as satisfiability. In *Proc. 10th European Conference on Artificial Intelligence (ECAI-2012)*, 359–363.
- Keyder, E., and Geffner, H. 2009. Soft goals can be compiled away. *J. Artif. Intell. Res.* 36:547–556.
- Kronegger, M.; Pfandler, A.; and Pichler, R. 2013. Parameterized complexity of optimal planning: A detailed map. In *Proc. of the 23rd International Joint Conference on Artificial Intelligence (IJCAI-2013)*.
- Mirkis, V., and Domshlak, C. 2013. Abstractions for over-subscription planning. In *Proc. 23rd International Conference on Automated Planning and Scheduling (ICAPS-2013)*.
- Palacios, H., and Geffner, H. 2009. Compiling uncertainty away in conformant planning problems with bounded width. *J. Artif. Intell. Res.* 35:623–675.
- Smith, D. 2004. Choosing objectives in over-subscription planning. In *Proc. 14th International Conference on Automated Planning and Scheduling (ICAPS-2004)*, 393–401.
- Taig, R., and Brafman, R. I. 2013. Compiling conformant probabilistic planning problems into classical planning. In *Proc. 23rd International Conference on Automated Planning and Scheduling (ICAPS-2013)*.
- van den Briel, M., and Kambhampati, S. 2005. Optiplan: Unifying IP-based and graph-based planning. *J. Artif. Intell. Res.* 24:919–931.
- van den Briel, M.; Nigenda, R. S.; Do, M. B.; and Kambhampati, S. 2004. Effective approaches for partial satisfaction (over-subscription) planning. In *Proc. 19th National Conference on Artificial Intelligence (AAAI-2004)*, 562–569.