Cost-Optimal and Net-Benefit Planning —
A Parameterised Complexity View

Meysam Aghighi and Christer Bäckström
Linköping University, Linköping, Sweden
meysam.aghighi@liu.se christer.backstrom@liu.se

Abstract
Cost-optimal planning (COP) uses action costs and asks for a minimum-cost plan. It is sometimes assumed that there is no harm in using actions with zero cost or rational cost. Classical complexity analysis does not contradict this assumption: planning is \textsc{PSPACE}-complete regardless of whether action costs are positive or non-negative, integer or rational. We thus apply parameterised complexity analysis to shed more light on this issue. Our main results are the following. COP is \textsc{W[2]}-complete for positive integer costs, i.e., it is no harder than finding a minimum-length plan, but it is para-\textsc{NP}-hard if the costs are non-negative integers or positive rationals. This is a very strong indication that the latter cases are substantially harder. Net-benefit planning (NBP) additionally assigns goal utilities and asks for a plan with maximum difference between its utility and its cost. NBP is para-\textsc{NP}-hard even when action costs and utilities are positive integers, suggesting that it is harder than COP. In addition, we also analyse a large number of subclasses, using both the PUBS restrictions and restricting the number of preconditions and effects.

1 Introduction
It is very common in planning and search to assign costs to actions (or operators) and ask for a solution which minimises the sum of these. Considering the prevalence and importance of this problem in the literature, surprisingly little attention is paid to motivate and discuss the choice of numeric domain for these costs. The following are only some examples to illustrate the diversity in the literature: Katz and Domshlak [2008] and Helmert et al. [2014] use non-negative reals; Bäckström and Jonsson [2013] and Yang et al. [2008] use non-negative integers; Cooper et al. [2011] use positive integers; Coles et al. [2008] specify non-negative costs, but no type; while Thayer et al. [2012] do not specify the costs at all, not even whether negative costs are allowed. Furthermore, only two of these publications give a motivation for the choice: accounting for only a subset of the transitions in the state space [Helmert et al., 2014] and simulating several goal states by zero-cost edges to a single goal state [Yang et al., 2008]. While the choice may be dictated by models of applications in many cases, the publications listed above focus primarily on theoretical investigations or empirical studies of benchmark examples. It may seem as if the choice of domain is often either not considered very important or is so little understood that an arbitrary choice is made.

Real values can be motivated in theoretical studies, since it makes the results general and there is a powerful collection of mathematical results to use. However, specific results for reals are seldom, if ever, used. In implementations and in complexity theory we are restricted to finite representations, so the initial costs are necessarily rational numbers. Furthermore, the mathematical operations used are usually such that irrational numbers cannot arise, even in theory. Hence, we study rational numbers instead of reals.

From a perspective of classical complexity analysis, one might argue that the choice of domain is not important; cost-optimal planning is \textsc{PSPACE}-complete regardless of whether we use integers or rational numbers or whether we allow zero cost or only positive values. However, this does not correlate well with practical experience, which often indicates that cost-optimisation does not perform very well. One problem is that zero-cost actions can result in very long plans with very low cost [Richter and Westphal, 2010]. Also big differences in action costs can cause similar problems [Cushing et al., 2010]. Contrasting these findings with the observations on domain choice above indicates that we still have a considerable gap of knowledge regarding cost-optimal planning and search. In order to get a more refined picture of this issue, we apply the tool of parameterised complexity analysis.

Parameterised complexity [Downey and Fellows, 1999] has been previously applied to plan-length optimisation [Bäckström et al., 2012; Kronegger et al., 2013], but almost no such results exist for cost-optimal planning. We analyse three different cases that differ in the type of action costs: positive integers, non-negative integers and positive rationals. The first case is \textsc{W[2]}-complete, which means it is of the same complexity as finding a plan of minimum length, while both the other cases are para-\textsc{NP}-hard, a very strong evidence
that they are significantly harder than the first one. That is, the choice of numeric domain is very important from an efficiency point of view. Our results correlate well with practical experience and they can, thus, help to explain the problems encountered and to find ways around them. In addition, we analyse a number of restricted cases based on the PUBS restrictions [Bäckström and Nebel, 1995] and on restricting the number of preconditions and effects, (cf. Bylander [1994]).

We further consider the related problem of net-benefit planning [van den Briel et al., 2004]. We show that this problem is para-NP-hard even if the action costs and utilities are positive integers, which suggests that it is harder than cost-optimal planning, although both problems are PSPACE-complete under classical analysis. Aghighi and Jonsson [2014] analysed a number of restricted cases based on the PUBS restrictions and it was invented with the purpose of delivering coarse-grained complexity analyses than traditional complexity theory. Aghighi and Jonsson [2014] also discussed the related problem of net-benefit planning, although both problems are

\[ I \subseteq \Sigma^\star \times Z_0 \]

constant. Aghighi and Jonsson [2014] analysed a number of restricted cases of this problem, using classical analysis. Aghighi and Jonsson [2014] also discussed the related problem of net-benefit planning, although both problems are

\[ I \subseteq \Sigma^\star \times Z_0 \]

be computed in time \( f(k) \cdot n^c \), where \( n = |I| \); and (3) there is a computable function \( g \) such that \( k' \leq g(k) \).

Not much is known about the relationship between the parameterised complexity classes and the standard ones, except that \( P \subseteq FPT \). There is otherwise no simple relationship between classes; for instance, there are NP-complete problems that are \( W[P] \)-complete but there are PSPACE-complete problems that are in \( FPT \). There are also parameterised problems outside the \( W \) hierarchy. Of particular interest to us is the class para-NP, which consists of all parameterised problems that can be solved in non-deterministic time \( f(k) \cdot n^c \), where \( f \) is an arbitrary computable function and \( c \) is a constant independent of both \( n \) and \( k \). It is known that \( W[P] \subseteq \text{para-NP} \), but not whether this inclusion is strict.

The 1th slice of a parameterised problem \( L \) is defined as \( L_i = \{ \langle \ell, k \rangle \in L \mid k = i \} \), which is not parameterised.

We will need the following generalisation of Corollary 2.16 in Flum and Grohe [2006].

Corollary 1. (to Theorem 2.14 in Flum and Grohe [2006]). Any non-trivial parameterised problem \( L \) with at least one NP-hard slice is para-NP-hard.

For a detailed account of parameterised complexity, see Downey and Fellows [1999] or Flum and Grohe [2006].

We will not formally define different problems for classical and parameterised analysis. Given an instance \( \langle I, k \rangle \), a classical analysis measures time as a function \( t(n) \) where \( n = |\langle I, k \rangle| = |I| + \log k \), while a parameterised analysis uses a multi-variable function \( t(n, k) \) where \( n = |I| \).

3 Planning

We use the SAS+ planning framework [Bäckström and Nebel, 1995]. Let \( V = \{ v_1, \ldots, v_n \} \) be a finite set of variables, with an implicit order \( v_1, \ldots, v_n \), each with a finite domain \( D(v_i) \). This defines the state space \( S(V) = D(v_1) \times \cdots \times D(v_n) \). A member \( s \in S(V) \) is called a (total) state and can be viewed as a total function that specifies a value in \( D(v_i) \) for each \( v_i \in V \). A partial state may leave the value undefined for some (or all) variables, and is thus a partial function. The value of a defined variable \( v_i \) in a (total or partial) state \( s \) is denoted \( s[v_i] \). If \( s \) is a partial state, then \( \text{vars}(s) \) is the set of variables with a defined value in \( s \).

A planning instance \( \mathcal{P} = (V, A, I, G) \) has a set of variables \( V \), a set of actions \( A \), a total initial state \( I \) and a partial goal state \( G \). Each action \( a \in A \) has a precondition \( \text{pre}(a) \) and an effect \( \text{eff}(a) \), both partial states. Let \( a \in A \) and \( s \in S(V) \). Then \( a \) is valid in \( s \) if \( \text{pre}(a) \subseteq \text{vars}(s) \) for all \( v \in \text{vars}(\text{pre}(a)) \), and the result of \( a \) in \( s \) is a state \( t \in S(V) \) such that for all \( v \in V \), \( t[v] = \text{eff}(a)[v] \) if \( v \in \text{vars}(\text{eff}(a)) \) and \( t[v] = s[v] \) otherwise. Let \( s_0, s_t \in S(V) \) and let \( \omega = a_1, \ldots, a_\ell \) be a sequence of actions. Then \( \omega \) is a plan from \( s_0 \) to \( s_t \) if either (1) \( \omega = \emptyset \) and \( \ell = 0 \) or (2) there are states \( s_1, \ldots, s_{\ell-1} \in S(V) \) such that for all \( i (1 \leq i \leq \ell) \), \( a_i \) is valid in \( s_{i-1} \) and \( s_i \) is the result of \( a_i \) in \( s_{i-1} \). Furthermore,

\[ 1 \text{Corollary 2.16 is derived from Theorem 2.14, which has membership restrictions that are not used in the relevant part of the proof. The same generalisation is tacitly used by Kronnegger et al. [2013].} \]

\[ 2 \text{L \subseteq \Sigma^\star \times Z_0 \ is non-trivial if } 0 \neq \{ 1 \mid \langle \ell, k \rangle \in L \} \neq \Sigma^\star. \]
ω is a plan (i.e., a solution) for P if it is a plan from I to some state t such that t[v] = G[v] for all v ∈ vars(G).

We will consider combinations of the following four restrictions on instances [Bäckström and Nebel, 1995].

**P (post-unique):** For all v ∈ V and x ∈ D(v), eff(a)[v] = x for at most one a ∈ A.

**U (unary):** For each a ∈ A, |vars(eff(a))| = 1.

**B (binary):** |D(v)| = 2 for all v ∈ V.

**S (single-valued):** For all a, b ∈ A and v ∈ V, if v ∈ vars(pre(a)) ∩ vars(pre(b)) and v ∉ vars(eff(a)) ∪ vars(eff(b)) then pre(a)[v] = pre(b)[v].

The class SAS+B corresponds to STRIPS with negative preconditions. We assume binary domains {0, 1}, where 1 is positive. We use the notation of Bylander [1994], e.g., B^2_+ denotes the restriction to actions a where pre(a) defines at most 2 variables and eff(a) defines at most one variable that must also be positive. A * denotes an unbounded effect.

Finding the minimum length of a plan is a well-studied problem in the literature, and our primary interest in it is to transfer complexity results to other problems. In order to discuss restrictions concisely, we instantiate the problem with some subclass C ⊆ SAS^+ of planning instances, e.g., C = SAS^+-UB is both unary and binary.

**LENGTH-OPTIMAL PLANNING (LOP(C))**

**Instance:** A SAS^+ instance P = (V, A, I, G) in C.

**Parameter:** A non-negative integer k.

**Question:** Does P have a plan ω of length |ω| ≤ k?

In cost-optimal planning we additionally specify a cost c(a) for each action a and ask for the minimum cost for a plan. The cost of a plan ω = a_1,...,a_n is c(ω) = ∑_{i=1}^{n} c(a_i). We additionally specify a domain Ω for the costs.

**COST-OPTIMAL PLANNING (COP(C, Ω))**

**Instance:** A tuple P = (V, A, I, G, c) such that (V, A, I, G) is in C and c : A → Ω is a cost function.

**Parameter:** A non-negative integer k.

**Question:** Does P have a plan ω of cost c(ω) ≤ k?

The following result is straightforward, but included for completeness since we are not aware of any explicit result in the literature covering all cases, i.e., Z, Z_0, Q+, and Q_0.

**Theorem 2.** Problem COP(SAS^+, Q_0) is in PSPACE and problem COP(SAS^+, Z_0) is PSPACE-hard.

**Proof sketch.** Cycles in the state space cannot improve the plan cost. Guess a cycle-free plan, one action at a time, and check the cost incrementally. This is in NPSPACE=PSPACE. Hardness by reduction from LOP(SAS^+), which is PSPACE-complete [Bylander, 1994], using unit action cost.

## 4 Complexity of Cost-optimal Planning

In this section we analyse the parameterised complexity of cost-optimal planning, first for positive integer costs and then for non-negative integer and positive rational costs. The results are summarized in Figure 1.

### 4.1 Positive Integers

We start with hardness for some restricted SAS^+ classes.

**Theorem 3.** Problem COP(SAS^+;R, Z_+) is W[1]-hard for R ∈ {UBS, B^2_0, B^2_1} and W[2]-hard for R ∈ {BS, B^4_+}.

**Proof.** LOP(C) fpt reduces trivially to COP(C, Z_+) with unit costs. The result follows since LOP(C) is W[1]-complete for R ∈ {UBS, B^2_0, B^2_1} and W[2]-complete for R ∈ {BS, B^4_+} [Bäckström et al., 2012; Bäckström et al., 2013].

Having established these hardness bounds, we turn to membership, first proving that COP(SAS^+, Z_+) is in W[2].

The following construction maps an instance P with positive integer action costs into an instance P' without action costs. Each action a in P with cost c(a) is replaced with a chain of actions a_1,...,a_n(a). The lock variable v_{lock} and the a_i variables guarantee that these actions must occur in sequence and not be interleaved with any other actions. Hence, the sequence a_1,...,a_n(a) has the same precondition and effect as a, but length c(a) instead of cost c(a). We write a : P → E to define an action a with precondition P, effect E and cost c(a), omitting c when the cost is irrelevant. We also write v = x in P (or E) for P[v = x] = E[v = x].

**Construction 4.** Let C be a class of SAS^+ instances and let (P, k) be an instance of COP(C, Z_+), where P = (V, A, I, G, c). Construct a LOP(C) instance (P', k'), where k' = k and P' = (V', A', I', G', c') is defined as:

- V' = V ∪ {v_{lock}} ∪ \{a_i | a ∈ A and 1 ≤ i ≤ c(a)}
- D(v_{lock}) = \{0, 1\}
- D(a_i) = \{0, 1\} for all a ∈ A and 1 ≤ i ≤ c(a).

- For each a ∈ A, if c(a) = 1, then A' contains the actions
  - a_1 : pre(a), v_{lock} = 0 ⇒ eff(a)
  - and otherwise A' contains the actions
  - a_1 : pre(a), v_{lock} = 0 ⇒ eff(a)
- for otherwise A' contains the actions
  - a_1 : pre(a), v_{lock} = 0 ⇒ u_1 = 1, v_{lock} = 1.
  - a_i : u_i = 1 ⇒ u_{i+1} = 0, u_{i+1} = 1 (1 < i < c(a)),
  - a_i : u_i = 1 ⇒ eff(a), u_{i+1} = 0, v_{lock} = 0.

- I'[v] = I[v] for all v ∈ V and otherwise I'[v] = 0.

We now use this construction to prove that COP is in W[2] if all action costs are positive integers and polynomially bounded in the instance size.

**Theorem 5.** Let p be an arbitrary polynomial. Then COP(SAS^+, Z_+) is in W[2] if it is restricted to cost functions c such that for every instance P = (V, A, I, G, c) it holds that c(a) ≤ p(|P|) for all a ∈ A.

**Proof.** Let (P, k) be an instance of COP(SAS^+, Z_+), where P = (V, A, I, G, c), and let (P', k'), where P' = (V', A', I', G'), be the corresponding LOP(SAS^+) instance according to Construction 4. Obviously, P has a plan of cost k if and only if P' has a plan of length k' = k. This is an fpt reduction from COP(SAS^+, Z_+) to LOP(SAS^+) since k' = k and we assume that c(a) ≤ p(|P|) for all a ∈ A. It follows that COP(C, Z_+) is in W[2] since LOP(SAS^+) is in W[2] [Bäckström et al., 2012].
The restriction to polynomial costs allows for a simple proof, but is not necessary. It can be avoided by adapting the model-checking technique used in Bäckström et al. [2012]. Membership in $W[1]$ for SAS⁺-U is open, but all PUBS classes that are not $W[1]$-hard are fixed-parameter tractable.

**Theorem 6.** $\text{COP}(\text{SAS}^+ \cdot \mathbb{P}, \mathbb{Z}_+) \in \text{FPT}$.

**Proof.** Modify the fpt algorithm for $\text{LOP}(\text{SAS}^+ \cdot \mathbb{P})$ [Bäckström et al., 2012, Theorem 5] to check the plan cost, instead of the length, against $k$. The complexity result still holds since the cost can never be lower than the length in this case. \qed

## 4.2 Non-negative Integers and Positive Rationals

We now show that COP is considerably harder if we additionally allow actions to have zero cost or positive rational cost. We also need the PLAN Satisfiability (PSAT) problem, which only asks if there is a plan or not, with no parameter.

**Theorem 7.** Let $C$ be an arbitrary subclass of $\text{SAS}^+$. If $\text{PSAT}(C)$ is $\text{NP}$-hard, then $\text{COP}(C, \mathbb{Z}_0)$ is para-$\text{NP}$-hard.

**Proof.** Let $P$ be an arbitrary $\text{PSAT}(\text{SAS}^+)$ instance, where $P = \langle V, A, I, G \rangle$. Define a corresponding $\text{COP}(\text{SAS}^+, \mathbb{Z}_0)$ instance $\langle P', k' \rangle$, where $k' = 0$, $P' = \langle V, A, I, G, \alpha \rangle$ and $c(a) = 0$ for all $a \in A$. This is a polynomial-time reduction from $\text{PSAT}(C)$ to $\text{COP}(C, \mathbb{Z}_0)$ for any subclass $C \subseteq \text{SAS}^+$, since it does not alter the instance except for adding action costs. The result thus follows from Corollary 1 when $\text{COP}(C)$ is $\text{NP}$-hard. \qed

**Corollary 8.** Problem $\text{COP}(\text{SAS}^+ \cdot \mathbb{R}, \mathbb{Z}_0)$ is para-$\text{NP}$-hard for $R \in \{ \text{UB}, \text{BS}, B_{1^+}, B_{2^+} \}$.

**Proof.** Follows from Thm. 7 since $\text{PSAT}(\text{SAS}^+ \cdot \mathbb{R})$ is $\text{NP}$-hard for $R \in \{ \text{UB}, \text{BS} \}$ [Bäckström and Nebel, 1995] and for $R \in \{ B_{1^+}, B_{2^+} \}$ [Bylander, 1994]. \qed

**Theorem 9.** Let $C$ be an arbitrary subclass of $\text{SAS}^+$. If $\text{LOP}(C)$ is $\text{NP}$-hard, then $\text{COP}(C, \mathbb{Q}_+) \in \text{para-$\text{NP}$-hard}$.

**Proof.** Let $\langle P, k \rangle$ be an arbitrary $\text{LOP}(\text{SAS}^+)$ instance, where $P = \langle V, A, I, G \rangle$. Define a corresponding $\text{COP}(\text{SAS}^+, \mathbb{Q}_+) \in \text{para-$\text{NP}$-hard}$ instance $\langle P', k' \rangle$, where $k' = 1$, $P' = \langle V, A, I, G, \alpha \rangle$ and $c(a) = 1/k$ for all $a \in A$. This is a polynomial-time reduction from $\text{LOP}(C)$ to $\text{COP}(C, \mathbb{Q}_+) \in \text{para-$\text{NP}$-hard}$ for any subclass $C \subseteq \text{SAS}^+$, since it does not alter the instance except for adding action costs. The result thus follows from Corollary 1 when $\text{LOP}(C)$ is $\text{NP}$-hard. \qed

**Corollary 10.** Problem $\text{COP}(\text{SAS}^+ \cdot \mathbb{R}, \mathbb{Q}_+)$ is para-$\text{NP}$-hard for $R \in \{ \text{UB}, \text{BS}, B_{0^+}, B_{1^+}, B_{1^+}\}$.

**Proof.** Follows from Thm. 9 since $\text{LOP}(\text{SAS}^+ \cdot \mathbb{R})$ is $\text{NP}$-hard for $R \in \{ \text{UB}, \text{BS}, \text{UBS} \}$ [Bäckström and Nebel, 1995] and for $R \in \{ B_{0^+}, B_{1^+}, B_{1^+}\}$ [Bylander, 1994]. \qed

The following observation explains why we cannot escape the difficulty by scaling positive rationals to positive integers.

**Observation 11.** A $\text{COP}(\text{SAS}^+, \mathbb{Q}_+)$ instance can be polynomially reduced to a $\text{COP}(\text{SAS}^+, \mathbb{Z}_0)$ instance by multiplying all costs and the parameter with a suitable value $\alpha$. This is, however, not an fpt reduction since $\alpha$ will typically not depend on the parameter (only), which contradicts condition (3) for fpt reductions.

Such reductions can be fpt reductions if we add further constraints, eg. by choosing $\alpha$ as the lowest common denominator of the costs and additionally require that also $\alpha$ is bounded by $k$. More properly, we could use one or more additional parameters to bound the costs in various ways, but that is out of the scope of this paper.

We do not prove any corresponding membership results, and it is not obvious that $\text{COP}(\text{SAS}^+, \mathbb{Z}_0)$ and $\text{COP}(\text{SAS}^+, \mathbb{Q}_+)$ are even para-$\text{NP}$, since the plan length may be exponential in the number of variables. Some cases remain open for non-negative integers, but the $\text{SAS}^+\cdot\text{PUS}$ case is easy even for non-negative rationals.

**Theorem 12.** Problem $\text{COP}(\text{SAS}^+ \cdot \text{PUS}, \mathbb{Q}_0)$ is in $\text{P}$.

**Proof sketch.** A length-optimal plan can be found in polynomial time, using a deterministic fixpoint algorithm [Bäckström, 1992]. This is the unique subset-minimal plan and thus also cost optimal. \qed

## 5 Complexity of Net-benefit Planning

The net-benefit problem is a so called oversubscription problem, where we do not expect to satisfy all of the goal. In addition to action costs, each goal variable $v$ has a utility value $U(v)$. Let $s$ be a state. The utility $U(v, s)$ in $s$ of a variable $v \in \text{vars}(G)$ is $U(v)$ if $z[v] = G[v]$, and otherwise $U(v, s) = 0$. The utility of $s$ is $U(s) = \sum_{v \in \text{vars}(G)} U(v, s)$. 

Figure 1: COP for positive integer action costs (left) and for non-negative integer and positive rational action costs (right).
If \( \omega \) is a plan from \( I \) to \( s \), then the difference \( U(\omega) - c(\omega) \) is called the net benefit of \( \omega \). The objective of the net-benefit problem is to find the maximal net benefit over all plans to any state. The functions \( c \) and \( U \) are assumed polynomial-time.

**Net-Benefit Planning (\( \text{NPBP}(C, D) \))**

**Instance:** A tuple \( \mathbb{P} = \langle V, A, I, G, c, U \rangle \) where \( \langle V, A, I, G \rangle \) is in \( C \), \( c : A \to \mathbb{D} \) is a cost function and \( U : \text{vars}(G) \to \mathbb{D} \) is a utility function.

**Parameter:** A non-negative integer \( k \).

**Question:** Is there a state \( s \in S(V) \) and a plan \( \omega \) from \( I \) to \( s \) such that \( U(s) - c(\omega) \geq k \)?

van den Briel et al. [2004] show that net-benefit planning is \( \text{PSPACE} \)-complete for non-negative costs and utilities. They do not specify if they consider integer or rational values, but this should not matter for \( \text{PSPACE} \)-completeness. Keyder and Geffner [2009] as well as Aghtgh and Jonsson [2014] present transformations from \( \text{NPBP} \) to planning. However, neither of these is an fpt reduction so the upper bounds for planning in the previous section do not automatically carry over to \( \text{NPBP} \).

We prove in this section that the net benefit problem is para-NP-hard even if the action costs and utilities are restricted to positive integers. That is, in contrast to cost-optimal planning, it is not obviously simpler to plan with positive integers than with non-negative integers or positive rationals. This holds also for very restricted classes.

**Construction 13.** Let \( \langle \mathbb{P}, k \rangle \) be a LOP(SAS\(^+\)) instance, where \( \mathbb{P} = \langle V, A, I, G \rangle \). Without losing generality, assume that \( \text{vars}(G) = \{v_1, \ldots, v_m\} \). Define a corresponding \( \text{NPBP}(\text{SAS}^+, \mathbb{Z}_+)_1 \) instance \( \langle \mathbb{P}', k' \rangle \), where \( k' = 1 \) and \( \mathbb{P}' = \langle V', A', I', G', c, U \rangle \) is defined as:

- \( V' = V \cup \{w, u_1, \ldots, u_m\} \), where \( D(w) = D(u_1) = \cdots = D(u_m) = \{0, 1\} \).
- \( A' = \{a' \mid a \in A\} \cup \{b_{w}, b_1, \ldots, b_m\} \), where
  - \( a' : \text{pre}(a), w = 0 \Rightarrow \text{eff}(a), \) for all \( a \in A \),
  - \( b_w : w = 0 \Rightarrow 1 \),
  - \( b_1 : v_1 = G[v_1], w = 1 \Rightarrow u_1 = 1 \),
  - \( b_i : v_i = G[v_i], u_{i-1} = 1 \Leftrightarrow u_i = 1, \) \( (2 \leq i \leq m) \).
- \( I'[v] = I[v] \) for all \( v \in V \) and otherwise \( I'[v] = 0 \).
- \( G'[u_m] = 1 \) and \( G' \) is otherwise undefined.
- \( U(u_m) = 2k + 2 \).

**Theorem 14.** Problems \( \text{NPBP}(\text{SAS}^+\text{-PBS}, \mathbb{Z}_+) \) and \( \text{NPBP}(\text{SAS}^+\text{-UBS}^+, \mathbb{Z}_+) \) are para-NP-hard.

**Proof.** Construction 13 maps an instance \( \langle \mathbb{P}, k \rangle \) of LOP(SAS\(^+\)) to an instance \( \langle \mathbb{P}', k' \rangle \) of \( \text{NPBP}(\text{SAS}^+, \mathbb{Z}_+) \). Suppose \( \mathbb{P} \) has a plan \( \omega = a_1, \ldots, a_k \) of length \( k \). Then \( \omega' = a'_1, \ldots, a'_k, b_{w}, b_1, \ldots, b_m \) is a plan for \( \mathbb{P}' \), where \( c(\omega') = km + 1 + mk = 2km + 1 \) and \( U(\omega') = 2km + 2 \), i.e. the net benefit is \( 1 = k' \). To the contrary, suppose \( \mathbb{P}' \) has a plan \( \omega' \) with net benefit of \( k' \) or more, to some state \( s \). This is only possible if \( s'[u_m] = 1 \), which requires the actions \( \beta = b_{w}, b_1, \ldots, b_m \) in that order. Hence, \( \omega' \) must have a prefix \( \alpha = a'_1, \ldots, a'_i \) that is a plan from \( I \) to some state \( s' \) such that \( \beta \) is a plan from \( s' \) to \( s \) and \( s'[v] = G[v] \) for all \( v \in \text{vars}(G) \). This implies that the action sequence \( a_1, \ldots, a_t \) that corresponds to \( \alpha \) is a plan for \( \mathbb{P} \). Furthermore, \( c(\omega') = km + 1 + mk + (\ell + k)m + 1 \) and \( U(\omega') = 2km + 2 \), so \( \omega' \) can have a net benefit of 1 or more only if \( \ell \leq k \). We conclude that \( \mathbb{P} \) has a plan of length \( k \), or less, if and only if \( \mathbb{P}' \) has a plan with net benefit \( k' \geq 1 \).

This construction is a polynomial-time reduction from LOP(SAS\(^+\)) to \( \text{NPBP}(\text{SAS}^+, \mathbb{Z}_+) \). Since \( \text{SAS}^+\text{-PBS} \) is closed under this reduction and it adds only one precondition, the result follows from Corollary 1 since both LOP(SAS\(^+\text{-PBS} \)) is NP-hard [Bäckström and Nebel, 1995] and LOP(SAS\(^+\text{-UBS}^+ \)) is NP-hard [Bylander, 1994].

**Theorem 15.** \( \text{NPBP}(\text{SAS}^+\text{-PBS}, \mathbb{Z}_+) \) is para-NP-hard.

**Proof sketch.** Analogous to the proof of Theorem 14 but modified as follows. Replace each variable \( v_i \) with two variables \( v_i^L \) and \( v_i^R \). These are always set simultaneously to the same value, which is possible since the actions need not be unary. We can now test the \( v_i^L \) variables in the preconditions of the \( a' \) actions and the \( v_i^R \) variables in the preconditions of the \( b_j \) actions. We also replace variable \( w \) with two variables, \( w^L \) and \( w^R \), which always have opposite values, i.e. \( w = 0 \) if \( w^L = 1 \) and \( w = 1 \) if \( w^R = 1 \). This new instance satisfies restriction \( S \) but not restriction \( U \) and is otherwise equivalent to the instance in Construction 13.

For \( \text{SAS}^+\text{-UBS} \), we reduce from the MSC problem, which is NP-complete. [Garey and Johnson, 1979, Problem SP5].

**Minimum Set Cover (MSC)**

**Instance:** A set \( S \) and a set \( C \) of subsets of \( S \).

**Parameter:** A positive integer \( k \).

**Question:** Does \( S \) have a cover of size \( k \), i.e. a subset \( C' \subseteq C \) such that \( \cup_{c \in C'} = S \) and \( |C'| = k \)?

**Theorem 16.** \( \text{NPBP}(\text{SAS}^+\text{-UBS}, \mathbb{Z}_+) \) is para-NP-hard.

**Proof.** We first define a reduction from Minimum Set Cover (MSC) to \( \text{NPBP}(\text{SAS}^+\text{-UBS}, \mathbb{Z}_+) \). Let \( \langle I, k \rangle \) be an MSC instance where \( I = \langle S, C \rangle \), \( S = \{x_1, \ldots, x_n\} \) and \( C = \{c_1, \ldots, c_m\} \). Define the corresponding instance \( \langle \mathbb{P}, k' \rangle \) of \( \text{NPBP}(\text{SAS}^+\text{-UBS}, \mathbb{Z}_+) \), where \( k' = k \) and \( \mathbb{P} = \langle V, A, I, G, c, U \rangle \) is defined as follows:

- \( V = \{x_i, v_i \mid x_i \in S \} \cup C, \) all with domain \( \{0, 1\} \).\n
- \( A \) contains the following actions:
  - \( \text{en}_i : \emptyset \Rightarrow c_i = 1, \) for all \( c_i \in C \),
  - \( \text{set}_i : c_i = 1 \Rightarrow x_j = 1, \) for all \( c_i \in C \) and \( x_j \in c_i \),
  - \( \text{vfy}_i : x_1 = 1 \Rightarrow v_i = 1 \), for all \( i \in S \).

- \( I[v] = 0 \) for all \( v \in V \).
- \( G[v_m] = 1 \) and \( G \) is otherwise undefined.
- \( U(u_m) = 3kn + 1 \).

Suppose \( I \) has a cover of size \( k \). Then \( \mathbb{P} \) has a plan \( \omega \) with \( k \) \( en \) actions, \( n \) \( set \) actions and all \( n \) \( vfy \) actions, i.e. \( c(\omega) = 3kn \), so the net benefit is \( 1 = k' \). Instead suppose \( \mathbb{P} \) has a plan with net benefit \( k' \geq 1 \). It must contain \( n \) set
actions and the n vfy actions, with total cost 2kn. The plan cost is at most 3kn so there are at most k en actions. Hence, 1 has a cover of size k. This is a polynomial-time reduction from MSC to NBP(SAS+ -UBS, Z+) since P ∈ SAS+- UBS. Hence, NBP(SAS+ -UBS, Z+) is NP-hard since MSC is NP-complete. It follows from Corollary 1 that NBP(SAS+ -UBS, Z+) is para-NP-hard.

For the final cases we reduce from INDEPENDENT SET, which is W[1]-complete [Downey and Fellows, 1999].

INDEPENDENT SET (IS)
Instance: A graph G = (V, E).
Parameter: A positive integer k.
Question: Is there a subset V′ ⊆ V such that \{u, v\} ∉ E for all u, v ∈ V′ and |V′| = k?

Theorem 17. NBP(SAS+ -PUBS, Z+) is W[1]-hard.

Proof. Proof by reduction from INDEPENDENT SET (IS). Let (G, k) be an instance of IS, where G = (V, E). Construct an instance (P, k′) of NBP(SAS+, Z+), where k′ = k and P = (V′, A, I, G, c, U) is defined as follows:

- V′ = V, where D(v) = \{0, 1\} for all v ∈ V′.
- For each v ∈ V, A contains the action a_v : \{u = 0 \mid \{u, v\} \in E\} \has \text{ \mathcal{D}} v = 1.
- For all v ∈ V, I[v] = 0, G[v] = 1 and U(v) = 2.

Obviously, G has an independent set of size k if and only if P has a plan with net benefit k′ = k. P is in SAS+ -PUBS, so this is an fpt reduction from IS to NBP(SAS+ -PUBS, Z+). The result follows since IS is W[1]-complete.

The NBP results coincide with COP(C, Q0) in Figure 1, except that SAS+ -PUBS and SAS+ -PUS are W[1]-hard.

6 Discussion
A somewhat similar, yet different, theoretical analysis was done by Helmut [2002], who studied the classical complexity of planning for different cases of using numerical variables and arithmetic actions, but without action costs. Practical approaches to combining numerical variables and action costs exist, though, (cf. Ivanovic et al. [2014]).

Analyses of different cost domains are usually based on practical experience, e.g. Richter and Westphal [2010] note that their planner LAMA does not perform well with cost-optimising heuristics. Indeed, in the International planning competition (IPC) 2008, none of the cost-optimising planners performed as well as the baseline planner, which ignored costs. It is not an isolated problem that cost-optimising planning seems not to deliver as expected [Cushing et al., 2010].

One reason is zero-cost actions. If the cheapest plans are very long, with many zero-cost actions, then a cost-optimising planner might time out, while a length-optimising planner may find a good enough plan. This correlates very well with our theoretical findings. A similar case arises for big spans in action costs, resulting in long and cheap plans with many cheap actions, called the ε-cost trap by Cushing et al. [2010].

Their argument scales the costs to the interval [0, 1]. Our results raise the question whether this seemingly harmless scaling might, in fact, contribute to the problem.

Cooper et al. [2011] may seem to avoid these problems, since they only allow positive integer costs. However, they transform the instance into a CSP instance and add virtual zero-cost actions. This raises the question of whether properties of positive integers are lost or not, and how to generally do such transformations? For instance, is it safe to simulate multiple goal states (cf. [Yang et al., 2008])?

Our results also suggest an obvious way to improve efficiency. Non-negative integers can be mapped to positive integers by a linear transformation c′(a) = λ · c(a) + b, for some positive integer constants λ and b. We get

\[ c′(\omega) = \sum_{a \in \omega} (\lambda \cdot c(a) + b) = \lambda \cdot c(\omega) + b \cdot |\omega|. \]

This moves the problem from being para-NP-hard to being W[2]-complete, at the expense of overestimating the optimal solutions. This mapping enforces a balance between optimising the cost and the length, a balance which can be tuned by the constants λ and b. However, this is already used in practice to improve efficiency, e.g. LAMA uses a heuristic that puts equal weight to the length and the cost of the plan [Richter and Westphal, 2010]. This is yet another correlation between our theoretical results and practical experience.

Other cost-based heuristics also depend on our results. For instance, Corollary 10 suggests that the h+ heuristic [Hoffmann, 2005] is very difficult to compute, unless restricted to positive integers; it is usually approximated in practice since it is NP-complete even for unit cost. An interesting example is Betz and Helmut [2009], who allow non-negative integer costs and prove that the h+ heuristic cannot be approximated within a constant. The proof, however, uses a technique similar to our Construction 4 to avoid using zero-cost actions, even though it would have been much simpler with such actions. In retrospect, this was a good decision since our results indicate that their result is stronger without zero costs.

In this paper, we analysed cases based on the PUBS restrictions and the number of preconditions and effects. Another way is to restrict the structure of the causal graphs and/or the domain-transition graphs (DTGs). There are numerous classical complexity results for cost-based planning under such restrictions (cf. Katz and Domshlak [2008]). Parameterised results are scarce, though. One exception is Bäckström [2014] who shows that COP(C, Q0) is in FPT for acyclic DTGs, using a combination of parameters on the causal graph and the DTGs.

One reason why COP(SAS+, Q0) is easier than NBP(SAS+, Z+) is most likely that the parameter implicitly bounds the plan length in the first case, but not in the second case, where the parameter only bounds the difference between the utility and the cost of a plan. One may consider adding further parameters, for instance, the plan length. This is similar to a common variant of NBP which assigns a cost budget B and asks for a plan with maximum utility under the constraint that the plan cost does not exceed B [Mirkis and Domshlak, 2013]. This problem can be viewed as a parameterised problem with two parameters, the budget B and the desired utility k, which could be easier.
References


