

Tight Lower Bounds for the Workflow Satisfiability Problem Based on the Strong Exponential Time Hypothesis

Gregory Gutin and Magnus Wahlström
Royal Holloway, University of London, UK

Abstract

The Workflow Satisfiability Problem (WSP) asks whether there exists an assignment of authorized users to the steps in a workflow specification, subject to certain constraints on the assignment. The problem is NP-hard even when restricted to just not equals constraints. Since the number of steps k is relatively small in practice, Wang and Li (2010) introduced a parametrisation of WSP by k . Wang and Li (2010) showed that, in general, the WSP is W[1]-hard, i.e., it is unlikely that there exists a fixed-parameter tractable (FPT) algorithm for solving the WSP. Crampton et al. (2013) and Cohen et al. (2014) designed FPT algorithms of running time $O^*(2^k)$ and $O^*(2^{k \log_2 k})$ for the WSP with so-called regular and user-independent constraints, respectively. In this note, we show that there are no algorithms of running time $O^*(2^{ck})$ and $O^*(2^{ck \log_2 k})$ for the two restrictions of WSP, respectively, with any $c < 1$, unless the Strong Exponential Time Hypothesis fails.

1 Introduction

The Workflow Satisfiability Problem (WSP) is a problem studied in the security research community, with important applications to information access control. In a WSP instance, one is given a set of k steps and a set of n users, and the goal is to find an assignment from the steps to the users, subject to some instance-specific constraints and authorization lists; see formal definition below. In practice, the number of steps tends to be much smaller than the number of users. Hence it is natural to study the problem from the perspective of *parameterized complexity*, taking k as a problem parameter. In general, the resulting parameterized problem is W[1]-hard [21], hence unlikely to be FPT, but for some natural types of constraints the problem has been shown to be FPT. In particular, Crampton *et al.* [10] gave an algorithm with a running time of $O^*(2^k)$ for so-called *regular* constraints, and Cohen *et al.* [6] gave an algorithm with a running time of $O^*(2^{k \log k})$ ¹ for *user-independent* constraints; see below. User-independent constraints in particular are common in the practice of access control. It was also shown that assuming the Exponential Time Hypothesis (ETH) [16], these algorithms cannot be improved to running times of $O(2^{o(k)})$ or $O(2^{o(k \log k)})$, respectively [10, 6]. Still, because of the importance of the problem, the question of moderately improved running times, e.g., algorithms of running time $O(2^{ck})$, respectively $O(2^{ck \log k})$, for some $c < 1$, remained open and relevant. In this paper, we will show that no such algorithms are possible, unless the so-called Strong Exponential Time Hypothesis (SETH) [15] fails – that is, up to lower-order terms, the algorithms cited above are time optimal.

In the remainder of this section, we formally introduce the Workflow Satisfiability Problem (WSP) and some families of constraints of interest for the WSP. We briefly overview the WSP literature that considers the WSP as a parameterized problem, as suggested by Wang and Li [21], and state our main results. We prove the results in the next section.

WSP. In the WSP, the aim is to assign authorized users to the steps in a workflow specification, subject to some constraints arising from business rules and practices. The Workflow Satisfiability Problem has applications in information access control (e.g. see [2, 3, 5]), and it is extensively studied in the security research community (e.g. see [3, 8, 13, 21]). In the WSP, we are given a set U of users, a set S of steps, a set $\mathcal{A} = \{A(s) : s \in S\}$ of authorization lists, where $A(s) \subseteq U$ denotes the set of users who are authorized to perform step s , and a set C of constraints. In general, a constraint $c \in C$ can be described as a

¹All logarithms in this paper are of base 2.

pair $c = (T, \Theta)$, where $T \subseteq S$ is the *scope* of the constraint and Θ is a set of functions from T to U which specifies those assignments of steps in T to users in U that satisfy the constraint (authorizations disregarded). Authorizations and constraints described in the WSP literature are relatively simple such that we may assume that all authorizations and constraints can be checked in polynomial time (in $n = |U|$, $k = |S|$ and $m = |C|$). Given a *workflow* $W = (S, U, \mathcal{A}, C)$, W is *satisfiable* if there exists a function $\pi : S \rightarrow U$ called a *plan* such that

- π is *authorized*, i.e., for all $s \in S$, $\pi(s) \in A(s)$ (each step is allocated to an authorized user);
- π is *eligible*, i.e., for all $(T, \Theta) \in C$, $\pi|_T \in \Theta$ (every constraint is satisfied).

Wang and Li [21] were the first to observe that the number k of steps is often quite small and so can be considered as a parameter. As a result, the WSP can be studied as a parameterized problem. Wang and Li [21] proved that the WSP is *fixed-parameter tractable* (FPT) if it includes only some special types of practical constraints (authorizations can be *arbitrary* as in all other research on WSP mentioned below). This means that the WSP restricted to the types of constraints in [21] can be solved by an FPT *algorithm*, i.e., an algorithm of running time $O(f(k)(n + k + m)^{O(1)}) = O^*(f(k))$, where $f(k)$ is a computable function of k only and O^* hides polynomial factors. However, in general, the WSP is W[1]-hard [21], which means that it is highly unlikely that, in general, the WSP is FPT.² The paper of Wang and Li has triggered an extensive study of FPT algorithms for the WSP from both theoretical and algorithm engineering points of view. We will briefly overview literature on the topic after introduction of some important families of the WSP constraints. In what follows, for a positive integer p , $[p]$ denotes the set $\{1, 2, \dots, p\}$.

WSP Constraints. We now introduce three families of WSP constraints which consecutively extend each other. Let T be a subset of S . A plan π satisfies a *steps-per-user counting constraint* (t_ℓ, t_r, T) , if a user performs either no steps in T or between t_ℓ and t_r steps. Steps-per-user counting constraints generalize the cardinality constraints which have been widely adopted by the WSP community [2, 4, 17, 20].

For $T \subseteq S$ and $u \in U$ let $\pi : T \rightarrow u$ denote the plan that assigns every step of T to u . A constraint $c = (L, \Theta)$ is *regular* if it satisfies the following condition: For any partition L_1, \dots, L_p of L such that for every $i \in [p]$ there exists an eligible³ plan $\pi : L \rightarrow U$ and user u such that $\pi^{-1}(u) = L_i$, the plan $\bigcup_{i=1}^p (L_i \rightarrow u_i)$, where all u_i 's are distinct, is eligible. Consider, as an example, a steps-per-user counting constraint (t_ℓ, t_r, L) . Let L_1, \dots, L_p be a partition of L such that for every $i \in [p]$ there exists an eligible plan $\pi : L_i \rightarrow U$ and user u such that $\pi^{-1}(u) = L_i$. Observe that for each $i \in [p]$, we have $t_\ell \leq |L_i| \leq t_r$ and so the plan $\bigcup_{i=1}^p (L_i \rightarrow u_i)$, where all u_i 's are distinct, is eligible. Thus, any steps-per-user counting constraint (t_ℓ, t_r, L) is regular.

A constraint (L, Θ) is *user-independent* if whenever $\theta \in \Theta$ and $\psi : U \rightarrow U$ is a permutation then $\psi \circ \theta \in \Theta$. In other words, user-independent constraints do not distinguish between users. Observe that all regular constraints are user-independent; however some user-independent constraints are not regular [10].

FPT Algorithms for the WSP. Crampton *et al.* [10] found a faster FPT algorithm, of running time $O^*(2^k)$, to solve the special cases of WSP studied by Wang and Li [21] and showed that the algorithm can be used for all regular constraints (all constraints studied in [21] are regular). Cohen *et al.* [6] showed that the WSP with only user-independent constraints is FPT and can be solved by an algorithm of running time $O^*(2^{k \log k})$. A simpler $O^*(2^{k \log k})$ -time algorithm was designed by Karapetyan *et al.* [18] for WSP with user-independent constraints. Also an $O^*(2^{k \log k})$ -time algorithm was obtained by Crampton *et al.* [9] for a natural optimization version of WSP, the Valued WSP, with (valued) user-independent constraints. The algorithms of these three papers were implemented in [7, 18, 9], respectively, and, in computational experiments, the implementations demonstrated a clear superiority of the FPT algorithms over well-known off-the-shelf solvers, the pseudo-boolean SAT solver SAT4J and the MIP solver CPLEX, for hard WSP and Valued WSP instances (in particular, the off-the-shelf solvers could not find solutions to many instances for which the FPT algorithm found solution within a few minutes).

Crampton *et al.* [10] and Cohen *et al.* [6], respectively, showed that under the Exponential Time Hypothesis (ETH) [16], there are no algorithms of running time $O^*(2^{o(k)})$ and $O^*(2^{o(k \log k)})$, respectively,

²For recent excellent introductions to fixed-parameter algorithms and complexity, see, e.g., [12, 14].

³We consider only constraints whose scope is a subset of L .

for the WSP with regular and user-independent constraints, respectively. However, these results leave possibility of the existence of algorithms of running time $O^*(2^{ck})$ and $O^*(2^{ck \log k})$, respectively, with $c < 1$. Such algorithms could be of practical interest as well as theoretical, if the improvement were significant enough. The aim of this note is to show that, unfortunately, such algorithms do not exist unless the Strong Exponential-Time Hypothesis (SETH) fails. Recall that SETH [15] states that

$$\lim_{t \rightarrow \infty} \inf \{c \geq 0 : t\text{-SAT has an algorithm in time } O(2^{cn})\} = 1.$$

SETH is a stronger hypothesis than ETH, and has been used repeatedly to argue that various algorithms are “probably optimal” [11, 19, 1]. In this sense, we show that the above-mentioned algorithms for regular respectively user-independent WSP are probably optimal, i.e., that they cannot be improved by current state of the art techniques and that improving them is as hard as improving the running time of SAT algorithms.

2 Lower Bounds

It is easy to prove that the WSP with regular constraints cannot be solved in time $O^*(2^{ck})$ for any $c < 1$ unless SETH fails via a simple reduction from SET SPLITTING. In SET SPLITTING, we are given a set S and a family $\{S_1, \dots, S_p\}$ of its subsets, and our aim is to decide whether there is a function $f : S \rightarrow \{1, 2\}$ such that both $f^{-1}(1) \cap S_i$ and $f^{-1}(2) \cap S_i$ are nonempty for every $i \in [p]$. Cygan *et al.* [11] proved that SET SPLITTING cannot be solved in time $O^*(2^{c|S|})$ for any $c < 1$, unless SETH fails. To reduce SET SPLITTING to the WSP with regular constraints, let S be the set of WSP steps, $U = \{1, 2\}$, $A(s) = U$ for each $s \in S$, and $C = \{(1, |S_i| - 1, S_i) : i \in [p]\}$. It remains to recall that $(1, |S_i| - 1, S_i)$ is a steps-per-user counting constraint, which is regular.

In the rest of this section, we prove that the WSP with user-independent constraints cannot be solved in time $O^*(k^{ck})$ for any $c < 1$ unless SETH fails. We will show it by an appropriate reduction from r -SAT to the WSP with user-independent constraints via (d, r) -CSP, the Constraint Satisfaction Problem with domain size d and every constraint of arity at most r . In (d, r) -CSP, we will consider only *clause-like constraints*, which are constraints with only one forbidden assignment for the scope variables.⁴

Let us fix a constant arity r , and let \mathcal{F} be an r -SAT formula with n variables. Let us fix a function $f(n) \in O(n^{o(1)}) \cap \omega(\log n)$ such that $n/f(n)$ is a power of 2, e.g., $\frac{1}{2} \log n \log \log n \leq f(n) \leq \log n \log \log n$. Let $d = n/f(n)$. We will first convert \mathcal{F} to an instance of (d, r) -CSP with $\lceil n/\log d \rceil$ variables, then reduce this instance to a WSP instance with appropriate size parameters. The following is our first step (which is simply done by grouping variables).

Lemma 1. *There is a reduction from r -SAT with n variables to (d, r) -CSP with only clause-like constraints and with $k = \lceil n/\log d \rceil$ variables, where $d = n/f(n)$. The reduction runs in polynomial time.*

Proof. Let the variables of \mathcal{F} be $X = \{x_1, \dots, x_n\}$ and $\ell = \log d = O(\log n)$. For simplicity, add extra variables to \mathcal{F} so that n is a multiple of ℓ . Note that this requires adding at most $\ell = o(n)$ new variables. We group X into $k = n/\ell$ variable groups $\mathcal{V} = \{V_1, \dots, V_k\}$ of ℓ variables per group. We also define a new domain $D = \{0, 1\}^\ell$. For a variable group V_i and a tuple $b = (b_1, \dots, b_\ell) \in D$, the statement $V_i = b$ is interpreted as the assignment where the j 'th member of V_i gets value b_j . Hence assignments $\mathcal{V} \rightarrow D$ are in 1-1 relationship with assignments $X \rightarrow \{0, 1\}$.

Next, for every clause C in \mathcal{F} , we proceed as follows. Let $V(C)$ be the scope of C , and observe that C is falsified by exactly one assignment to $V(C)$. Similarly, the problem (d, r) -CSP allows us to arbitrarily specify forbidden assignments to sets of up to r variables. Clearly, the variables of $V(C)$ occur in at most r variable groups in \mathcal{V} . We can simply enumerate all assignments to these variable groups, these being at most $|D|^r = d^r \leq n^r$, and for every such assignment that is an extension of the assignment forbidden by C , we add a constraint to (d, r) -CSP forbidding this assignment. Since this is a polynomial number of constraints for every clause of \mathcal{F} , this can be done in polynomial time in total (recall that r is a constant). Some of the resulting constraints may have arity less than r , e.g., some constraints may even be unary. This is allowed in our problem model. \square

⁴Note that clauses of CNF SAT are clause-like constraints, which “justifies” the term clause-like. Clearly, an arbitrary CSP constraint can be decomposed into clause-like constraints.

Next, we show how we reduce from (d, r) -CSP to WSP with user-independent constraints.

Lemma 2. *Let $d = n/f(n)$. There is a polynomial-time reduction from (d, r) -CSP with only clause-like constraints and with $k = \lceil n/\log d \rceil$ variables to the user-independent WSP with d users and $k + d$ steps.*

Proof. Consider a (d, r) -CSP instance with only clause-like constraints; we will use notation as above, i.e., variable set $\mathcal{V} = \{V_1, \dots, V_k\}$ and domain D . However, number and rename elements of D such that $D = \{1, \dots, d\}$. We create a WSP instance with two sets of steps, and users $U = D$. The *fixed steps* are $S_D = \{s_1, \dots, s_d\}$, where for $i \in [d]$ the authorization list of s_i is $A(s_i) = \{i\}$. The *free steps* are $S_X = \{s'_1, \dots, s'_k\}$, each of which has a full authorization list $A(s) = U$.

Recall that every constraint in the (d, r) -CSP instance has a single forbidden assignment. Next, for every constraint in the (d, r) -CSP instance, over a scope $C = \{V_{q(1)}, \dots, V_{q(p)}\}$ ($p \leq r$) with a single forbidden assignment $\phi : C \rightarrow D$, we add the following constraint to the WSP instance:

$$\neg\left(\bigwedge_{i=1}^p (s'_{q(i)} = s_{\phi(V_{q(i)})})\right),$$

where $s'_t = s_j$ means that s'_t and s_j must be assigned to the same user. Note that the above WSP constraints are user-independent as they do not distinguish between users. It is clear that the reduction can be performed in polynomial time in the size of the input.

For correctness, we make two observations. First, by construction, for every user $i \in U$ and every authorized plan $\phi : S_D \cup S_X \rightarrow U$, there is exactly one step $s \in S_D$ such that $\phi(s) = i$. Hence, for every step $s' \in S_X$, we have $\phi(s') = i$ if and only if $\phi(s') = \phi(s_i)$. Second, let ϕ be an authorized plan as above, and define $\phi' : \mathcal{V} \rightarrow D$ by $\phi'(V_i) = \phi(s'_i)$. Then (by the previous observation) for every constraint C of the (d, r) -CSP instance, ϕ' satisfies C if and only if ϕ is eligible for the corresponding constraint of the WSP instance. This shows the equivalence of the instances. \square

Note that our WSP instance has constraint of bounded arity (at most $2r$). This may not be critical, but it alleviates some potential concerns (e.g., the specific encoding of the WSP constraints is not important). We can now wrap up the proof.

Theorem 1. *The WSP with user-independent constraints cannot be solved in time $O^*(2^{ck \log k})$ for any $c < 1$ unless SETH fails.*

Proof. In this proof, for functions $g'(n)$ and $g''(n)$, we write $g'(n) \sim g''(n)$ if $g'(n) = g''(n)(1 + o(1))$. Observe that $g'(n) \sim g''(n)$ and $g''(n) \sim g'''(n)$ imply $g'(n) \sim g'''(n)$.

Chaining the two reductions above, we have a polynomial-time reduction from an r -SAT instance \mathcal{F} on n variables to the WSP instance on $k + d$ variables, where $d = n/f(n)$ and $k = \lceil n/\log d \rceil$. In particular, we can write

$$k \sim \frac{n}{\log n - \log f(n)} = \frac{n}{\log n} \cdot \frac{\log n}{\log n - \log f(n)} = \frac{n}{\log n} \left(1 + \frac{\log f(n)}{\log n - \log f(n)}\right) \sim \frac{n}{\log n}.$$

Similarly, we have $d = \lceil n/f(n) \rceil = o(n/\log n)$, hence

$$k' = k + d \sim \frac{n}{\log n}.$$

Now note that

$$k' \log k' = \frac{n}{\log n} (1 + o(1)) (\log n - \log \log n + o(1)) \sim n.$$

Hence for any $c < 1$, solving the WSP instance in the stated time would imply solving every r -SAT instance for every r in time $O(2^{c'n})$ for some $c' < 1$ independent of r . This would contradict SETH. \square

Acknowledgments

Gutin's research was partially supported by EPSRC grant EP/K005162/1 and by Royal Society Wolfson Research Merit Award.

References

- [1] A. Abboud and V. V. Williams. Popular conjectures imply strong lower bounds for dynamic problems. In *FOCS '14*, pages 434–443, 2014.
- [2] American National Standards Institute. ANSI INCITS 359-2004 for role based access control, 2004.
- [3] D. Basin, S. J. Burri, and G. Karjoth. Obstruction-free Authorization Enforcement: Aligning Security and Business Objectives. *Journal of Computer Security*, 22:661–698, 2014.
- [4] E. Bertino, P. A. Bonatti, and E. Ferrari. TRBAC: A temporal role-based access control model. *ACM Trans. Inf. Syst. Secur.*, 4(3):191–233, 2001.
- [5] E. Bertino, E. Ferrari, and V. Atluri. The specification and enforcement of authorization constraints in workflow management systems. *ACM Trans. Inf. Syst. Secur.*, 2(1):65–104, 1999.
- [6] D. Cohen, J. Crampton, A. Gagarin, G. Gutin, and M. Jones. Iterative plan construction for the workflow satisfiability problem. *Journal of Artificial Intelligence Research*, 51:555–577, 2014.
- [7] D. Cohen, J. Crampton, A. Gagarin, G. Gutin, and M. Jones. Algorithms for the workflow satisfiability problem engineered for counting constraints. *Journal of Combinatorial Optimization*, available online doi:10.1007/s10878-015-9877-7:1–22, 2015.
- [8] J. Crampton. A reference monitor for workflow systems with constrained task execution. In *SACMAT*, pages 38–47. ACM, 2005.
- [9] J. Crampton, G. Gutin, and D. Karapetyan. Valued workflow satisfiability problem. In *Proceedings of the 20th ACM Symposium on Access Control Models and Technologies*, SACMAT '15, pages 3–13. ACM, 2015.
- [10] J. Crampton, G. Gutin, and A. Yeo. On the parameterized complexity and kernelization of the workflow satisfiability problem. *ACM Trans. Inf. Syst. Secur.*, 16(1):4, 2013.
- [11] M. Cygan, H. Dell, D. Lokshtanov, D. Marx, J. Nederlof, Y. Okamoto, R. Paturi, S. Saurabh, and M. Wahlström. On problems as hard as CNF-SAT. In *IEEE Conference on Computational Complexity*, pages 74–84. IEEE Computer Society, 2012.
- [12] M. Cygan, F. V. Fomin, L. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh. *Parameterized Algorithms*. Springer, 2015.
- [13] D. R. dos Santos, S. Ranise, L. Compagna, and S. E. Ponta. Assisting the deployment of security-sensitive workflows by finding execution scenarios. In P. Samarati, editor, *Data and Applications Security and Privacy XXIX*, volume 9149 of *Lecture Notes in Computer Science*, pages 85–100. Springer, 2015.
- [14] R. G. Downey and M. R. Fellows. *Fundamentals of Parameterized Complexity*. Springer, 2013.
- [15] R. Impagliazzo and R. Paturi. On the complexity of k -SAT. *J. Comput. Syst. Sci.*, 62(2):367–375, 2001.
- [16] R. Impagliazzo, R. Paturi, and F. Zane. Which problems have strongly exponential complexity? *J. Comput. Syst. Sci.*, 63(4):512–530, 2001.
- [17] J. Joshi, E. Bertino, U. Latif, and A. Ghafoor. A generalized temporal role-based access control model. *IEEE Trans. Knowl. Data Eng.*, 17(1):4–23, 2005.
- [18] D. Karapetyan, A. Gagarin, and G. Gutin. Pattern backtracking algorithm for the workflow satisfiability problem with user-independent constraints. In *9th International Frontiers of Algorithmics Workshop, 3–5 July 2015*, volume 9130 of *Lecture Notes in Computer Science*, pages 138–149. Springer, 2015.
- [19] D. Lokshtanov, D. Marx, and S. Saurabh. Known algorithms on graphs on bounded treewidth are probably optimal. In *SODA '11*, pages 777–789, 2011.

- [20] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman. Role-based access control models. *IEEE Computer*, 29(2):38–47, 1996.
- [21] Q. Wang and N. Li. Satisfiability and resiliency in workflow authorization systems. *ACM Trans. Inf. Syst. Secur.*, 13(4):40, 2010.