Contents lists available at ScienceDirect

Information Processing Letters

www.elsevier.com/locate/ipl

NP-hardness of shortest path problems in networks with non-FIFO time-dependent travel times

Tim Zeitz

Karlsruhe Institute of Technology, Institute of Theoretical Informatics, Algorithmics, Am Fasanengarten 5, 76131, Karlsruhe, Germany

A R T I C L E   I N F O

Article history:
Received 4 March 2022
Received in revised form 23 May 2022
Accepted 23 May 2022
Available online 27 May 2022
Communicated by Leah Epstein

Keywords:
Non-FIFO time-dependent shortest paths
NP-hardness
Theory of computation

A B S T R A C T

We study the complexity of earliest arrival problems on time-dependent networks with non-FIFO travel time functions, i.e. when departing later might lead to an earlier arrival. In this paper, we present a simple proof of the weak NP-hardness of the problem for travel time functions defined on integers. This simplifies and reproduces an earlier result from Orda and Rom. Our proof generalizes to travel time functions defined on rational numbers and also implies that, in this case, the problem becomes harder, i.e. is strongly NP-hard. As arbitrary functions are impractical for applications, we also study a more realistic problem model where travel time functions are piecewise linear and represented by a sequence of breakpoints with integer coordinates. We show that this problem formulation is strongly NP-hard, too. As an intermediate step for this proof, we also show the strong NP-completeness of SubsetProduct on rational numbers.

1. Introduction

Finding shortest paths in time-dependent networks is an important problem with numerous applications. A prominent example would be routing in road networks with traffic predictions, i.e. when the travel time of a road segment depends on the time of the day at which the segment is traversed. Another example is routing packets through computer networks when link capacities and delays depend on the time. The core problem in these applications is, given a source vertex, a target vertex and a departure time, to obtain the path with the *earliest arrival* at the target. This path is sometimes also referred to as a *minimum delay* path.

A critical property which time-dependent networks may or may not exhibit is the *first-in-first-out* (FIFO) property. Informally, this property holds when it is not possible to arrive earlier by departing later. For networks where this property holds, the earliest arrival problem can be solved with a variation of Dijkstra's algorithm [1]. However, realistic networks may not always adhere to the FIFO property. For example in road networks, when an important tunnel can only be traversed during certain times, one may have to drive a longer detour when arriving too early while the tunnel has not yet opened. Of course, it may be possible to avoid such a detour by waiting. Likewise, non-FIFO networks can be transformed into FIFO networks when waiting is possible all the time at any vertex. However, allowing waiting at arbitrary locations and times is not always a realistic modeling assumption. Cars cannot park at arbitrary locations and network packets may be dropped when buffers become full. In this paper, we therefore study the complexity of the earliest arrival problem in networks with non-FIFO travel times when waiting is not allowed.

Related work Time-dependent shortest path problems were studied extensively in the past, both from a theoretic and practical perspective. As an extensive overview is beyond the scope of this work, we refer to the survey in [2]. Most practical works assume that all travel times

E-mail address: tim.zeitz@kit.edu.

https://doi.org/10.1016/j.ipl.2022.106287

adhere to the FIFO property or that arbitrary waiting is allowed [3–6]. An exception is [7] where routing for truck drivers considering temporary driving bans is studied.

The complexity of the earliest arrival problem in non-FIFO time-dependent networks has been studied before, most prominently by Orda and Rom [8,9]. In [8], they present several time-dependent shortest path algorithms and study different waiting policies. They prove that under certain assumptions allowing waiting at the source vertex is sufficient to find a path with the same arrival as the fastest path when arbitrary waiting is allowed. Also, they show that in some cases, the path with the earliest arrival may use an infinite number of hops. Many recent works [4,5,10,2] cite [8] stating that the earliest arrival problem in non-FIFO time-dependent networks is NP-hard. However, the paper just states that the hardness can be shown (second paragraph of Section 3.2) but does not provide any evidence. Some works [7,6,3] state the hardness referring to an unpublished manuscript by Orda and Rom [11]. We were not able to find this manuscript in any public source but could only obtain it through personal contact with authors. In this manuscript, Orda and Rom proved the weak NP-hardness of the time-dependent earliest arrival problem with travel time functions defined on integers and forbidden waiting by reduction from FINITE-FUNCTIONGENERATION. They also show that the problem can be solved in pseudo-polynomial time. However, since the proof is quite complex and was apparently never published, we believe that independently reproducing this result is a valuable contribution.

Contribution We contribute a simple proof of the weak NP-hardness of the earliest arrival problem on networks with time-dependent travel times defined on integers when waiting is forbidden. Additionally, this proof implies that the problem becomes strongly NP-hard when the functions are defined on rational numbers. Further, we show that the problem remains strongly NP-hard when we limit ourselves to a more practical travel time function model where functions are piecewise linear and given as a sequence of breakpoints with integer coordinates, but the computation is performed in the rational number domain. As an intermediate result we show that SUBSETPRODUCT on rational numbers is strongly NP-complete.

2. Preliminaries

We consider directed graphs $G = (V, E)$ with $n = |V|$ vertices and $m = |E|$ edges. We use uv as a short notation for edges. Each edge uv has a *travel time function* f_{uv}^{tt} which maps a departure time τ to a travel time $f_{uv}^{\mathrm{tt}}(\tau)$ at that departure. We consider both functions defined on integers $f : \mathbb{Z} \to \mathbb{Z}$ and on rational numbers $f : \mathbb{Q} \to \mathbb{Q}$. Realistic travel times are non-negative but here we also allow negative travel times for technical reasons. Sometimes, it is more practical to consider the *arrival time function* $f_{uv}^{\mathrm{at}}(\tau) = \tau + f_{uv}^{\mathrm{tt}}(\tau)$. For example, for two edges uv and vw, the travel time function for traversing them successively is $f_{uv}^{\mathrm{tt}}(\tau) + f_{vw}^{\mathrm{tt}}(\tau + f_{uv}^{\mathrm{tt}}(\tau))$ while in arrival time representation the result is just the composition $f_{vw}^{\mathrm{at}}(f_{uv}^{\mathrm{at}}(\tau))$.

A travel time function f^{tt} has the *first-in-first-out* (FIFO) property when the following holds:

$$\forall \tau, \epsilon > 0 : f^{\mathrm{tt}}(\tau) \leq \epsilon + f^{\mathrm{tt}}(\tau + \epsilon)$$

Assuming continuous functions, time intervals where the FIFO property is violated have slopes < -1 in travel time representation and slopes < 0 in arrival time representation.

A sequence of vertices $P = (v_1, \ldots, v_k)$ where $v_i v_{i+1} \in E$ is called a *path*. Paths may be non-simple, i.e. contain the same vertex multiple times. We denote by $P_{i,j} = (v_i, \ldots, v_j), 1 \leq i < j \leq k$ a *subpath* of P. The travel time of a path $P = (v_1, v_2, \ldots, v_k)$ can be obtained recursively: $f_P^{\mathrm{tt}}(\tau) = f_{v_1 v_2}^{\mathrm{tt}}(\tau) + f_{P_{2,k}}^{\mathrm{tt}}(\tau + f_{v_1 v_2}^{\mathrm{tt}}(\tau))$. The goal of the *time-dependent earliest arrival problem* (TDEA) is, given a source vertex s, a target vertex t and a departure time τ to find among all paths $P = (s, \ldots, t)$ the one that minimizes $f_P^{\mathrm{tt}}(\tau)$. We define the decision problem as follows:

Definition 1 *(TDEA$_T$).* Given a graph $G = (V, E)$ with non-negative travel times $f_{uv}^{\mathrm{tt}} : T \to T^{\geq 0}$ for each $uv \in E$, vertices s and t, a departure time $\tau^{\mathrm{dep}} \in T$ and a maximum arrival time $\tau^{\max} \in T$, is there a path $P = (s, \ldots, t)$ such that $\tau^{\mathrm{dep}} + f_P^{\mathrm{tt}}(\tau^{\mathrm{dep}}) \leq \tau^{\max}$?

We assume that the functions are given in some well-defined compact representation, for example piecewise constant or piecewise linear, which can be evaluated in polynomial time. For the first part of the paper, we leave the concrete function class unspecified. In the second part we use piecewise linear functions represented by a sequence of breakpoints. We define the travel times before the first and after the last breakpoints to equal the travel time of the first and last breakpoint, respectively.

3. Complexity results

In this section we present our complexity results. We start with the simplified NP-hardness proof for TDEA$_{\mathbb{Z}}$.

Theorem 2. TDEA$_{\mathbb{Z}}$ *is weakly NP-hard.*

Proof. We prove NP-hardness by reduction from the weakly NP-complete problem SUBSETSUM [12]. A SUBSETSUM instance consists of a multiset A of integers a_i and a target value B.[1] The goal is to decide whether there is a subset $A' \subseteq A$ such that $\sum_{a \in A'} a = B$. We construct our TDEA$_{\mathbb{Z}}$ instance as follows: The instance has vertices $V = \{v_i \mid a_i \in A\} \cup \{s, t\}$. Let $v_0 = s$ and $k = |A|$. For each SUBSETSUM element $a_i \in A$, we create two edges from v_{i-1} to v_i. The first one denoted as e_i' has constant travel time zero and the second one denoted as e_i constant travel time a_i. We insert a final edge $v_k t$ with the following non-FIFO travel time function:

[1] Note that usually SUBSETSUM is defined in terms of a set and a function mapping each element to a not necessarily unique weight. To simplify notation, we instead use a multiset containing the weights directly as elements.

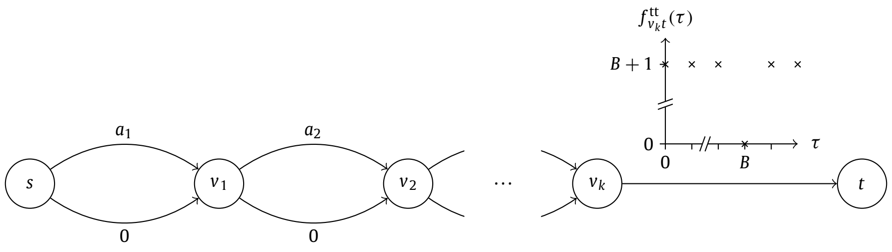

Fig. 1. Transformed TDEA instance for SubsetSum instance $(A = \{a_1, \ldots, a_k\}, B)$.

$$f^{\mathrm{tt}}_{v_k t}(\tau) = \begin{cases} 0 & \text{if } \tau = B \\ B+1 & \text{else} \end{cases}$$

Setting $\tau^{\mathrm{dep}} = 0$ and $\tau^{\max} = B$ completes the instance. See Fig. 1 for an illustration. This transformation runs in $\mathcal{O}(|A|)$. The transformed graph contains zero weights and multiedges, but neither is necessary. Zero weights can be eliminated by adding 1 to the travel time of each edge, shifting the instant where $f^{\mathrm{tt}}_{v_{|A|+1}t}$ has its minimum back by k time units and increasing the arrival time $\tau^{\max}$ by $k+1$. Multiedges can be avoided by inserting an additional node in the middle of each multiedge.

We now show the equivalency of the transformed instance. Assume that the SubsetSum instance admits a subset A' with $\sum_{a \in A'} a = B$. Consider the path P from s to v_k which uses e_i if $a_i \in A'$ and e'_i otherwise. Obviously, it has a constant total travel time of B. Thus, the edge $v_k t$ will be traversed at $\tau = B$. Since $f^{\mathrm{tt}}_{v_k t}(B) = 0$ the total travel time is B. It follows that the TDEA$_{\mathbb{Z}}$ instance admits a feasible path.

Conversely, assume that the TDEA$_{\mathbb{Z}}$ instance admits a path P of travel time B. Since the last edge has a travel time greater than B for all times except B, the travel time to v_k must have been B, too. Let $A' = \{a_i \mid e_i \in P\}$. Since the travel time to v_k was B, $\sum_{a \in A'} a = B$ has to hold and the SubsetSum instance also admits a feasible solution. This proves the weak NP-hardness. □

To prove that the problem is not strongly NP-hard, we now give a pseudo-polynomial algorithm. Algorithm 1 shows the procedure in pseudo-code. The algorithm is a time-expanded variation of the shortest path algorithm of Bellman and Ford [13]. Instead of a distance array, it maintains a 2-dimensional array indicating whether it is possible to reach a vertex at a given time. Instead of scanning edges $|V|$ times, edges are relaxed for each instant between τ^{dep} and $\tau^{\max}$. This leads to a running time in $\mathcal{O}(\tau^{\max} \cdot |E|)$ which makes the algorithm pseudo-polynomial in the input size.

For simplicity, we describe the algorithm under the assumption that all travel times are strictly positive. Travel times of zero can be handled by repeating the loop in line 4 $|V|$ times.

We prove correctness inductively. We show that in step τ of the loop in line 3 for all vertices v, all possible arrival times $\tau' \le \tau$ have been correctly marked in R. Clearly, the base case holds, because in the absence of zero travel times, s is the only vertex reachable at τ^{dep}. Assume the induction hypothesis for $\tau - 1$. For any path $(s, \ldots, u, v)$ of

```
Data: R[u][τ]: 2-dimensional array of booleans indicating
      whether it is possible to arrive at vertex u at time τ
Data: G = (V, A): Directed graph with travel time functions $f^{\mathrm{tt}}_e$
      for each $e \in E$
1 Function TimeExpandedBellmanFord($s, t, \tau^{\mathrm{dep}}, \tau^{\max}$):
2     R[s][$\tau^{\mathrm{dep}}$] ← true;
3     for $\tau \in [\tau^{\mathrm{dep}}, \tau^{\max}]$ do
4         for $uv \in E$ do
5             if R[u][τ] then
6                 if $u = t$ then
7                     return true;
8                 R[v][$\tau + f^{\mathrm{tt}}_{uv}(\tau)$] ← true;
9     return false;
```

Algorithm 1: Time-expanded Bellman-Ford algorithm.

travel time $\tau - \tau^{\mathrm{dep}}$ the vertex u was reachable at a time $\tau' < \tau$. Thus, R$[u][\tau']$ was correctly set, the outgoing edges of u were relaxed and R$[v][\tau]$ will also be set which completes the inductive step. Repeating the loop in line 4 $|V|$ times would additionally allow intermediate paths of travel time zero with up to $|V|$ vertices, which clearly suffices to solve this case, too.

Wojtczak showed in [14] that many common weakly NP-hard problems, including SubsetSum, become strongly NP-hard when defined on rational numbers. As our proof generalizes to rational numbers without modification, we get the following corollary:

Corollary 3. TDEA$_{\mathbb{Q}}$ *is strongly NP-hard.*

We now restrict ourselves to a function class used in many practical implementations: piecewise linear functions represented as a sequence of breakpoints. Evaluating the function at a time between two breakpoints is implemented through linear interpolation. As input data usually comes at a limited resolution, we assume that all breakpoint coordinates can be represented by integers. Nevertheless, the result of a linear interpolation may be a rational number. Thus, when performing the computation in the integer domain, one has to round. However, practical implementations often need to compute travel time functions for paths, for example to find the shortest travel time *function* between two nodes, i.e. solve *profile queries*. In the integer domain, this is very difficult because of the rounding. Therefore, many practical implementations [4,3,6] approximate computation in the continuous domain by using floating point numbers. This lead us to the question if TDEA$_{\mathbb{Q}}$ on piecewise linear functions with integer coordinates is weakly or strongly NP-hard.

With piecewise linear functions with integer breakpoints, we cannot easily construct constant travel time functions with arbitrary rational travel times. Thus, the generalizing the reduction from SubsetSum is difficult. However, we can encode rational numbers in the *slope* of arrival time function and exploit that when compositing linear functions, the result function's slope is equal to the product of the slopes of the composed functions. In the following, we show the strong NP-hardness with a two-step reduction from SubsetProduct on rational numbers as an intermediate problem. On integers, SubsetProduct is weakly NP-complete [12,15].[2] However, as we now show, when allowing rational numbers, the problem becomes strongly NP-complete. SubsetProduct is defined as follows:

Definition 4 *(*SubsetProduct*).* Given a finite multiset of positive numbers A and a positive number B, is there a subset $A' \subseteq A$ such that $\prod_{a\in A'} a = B$?

Theorem 5. SubsetProduct *on rational numbers is strongly NP-complete.*

Proof. A guessed solution for SubsetProduct can be verified in polynomial time. In the worst case, all numerators and denominators in the solution would have to be multiplied with each other. Thus, SubsetProduct is in NP even with rational numbers.

For hardness, we reduce from the ExactCoverBy3Sets problem. It is strongly NP-complete due to [12] and defined as follows:

Definition 6 *(*ExactCoverBy3Sets *(X3C)).* Given a set $X = \{x_1, \dots, x_k\}$ with k being a multiple of 3, and a collection C of 3-element subsets of X, does C contain an exact cover C' for X, where $C' \subseteq C$ and every element in X occurs in exactly one element of C'?

For the transformation, we first generate the first $2k+1$ primes. We refer to these primes in two groups of each k primes $\{p_1, \dots, p_k\}$ and $\{p'_1, \dots, p'_k\}$ and one final prime number p^*. Generating these primes is possible in polynomial running time [14]. We construct the set $A = A_X \cup A_C$ from one number for each element in X and one number for each 3-element set in C:

$$A_X = \left\{ \frac{p_i p'_{i+1}}{p'_i} \,\middle|\, x_i \in X \setminus x_k \right\} \cup \left\{ p^* \cdot \frac{p_k p'_1}{p'_k} \right\}$$

$$A_C = \left\{ \frac{1}{p_i p_j p_k} \,\middle|\, \{c_i, c_j, c_k\} \in C \right\}$$

Setting $B = p^*$ completes the instance.

Assume that the X3C instance admits a set $C' \subseteq C$ which covers X. Consider the set $A' = A_X \cup A_{C'}$ containing all elements from A_X and all elements from A_C corresponding to elements form C'. By construction, the product over all numbers in A' equals p^*:

$$\prod_{a\in A'} a = \prod_{a\in A_X} a \cdot \prod_{a\in A_{C'}} a = p^* \prod_{x_i\in X} \frac{p_i p'_i}{p'_i} \cdot \prod_{x_i\in X} \frac{1}{p_i} = p^*$$

It follows that the SubsetProduct admits a feasible subset, too.

Conversely, assume S' is a fulfilling subset for the SubsetProduct instance. First, since a_k is the only element which contains p^* as a factor, it must be contained in S'. Secondly, note that if S' contains any element from A_X, it has to contain *all* elements in A_X to cancel out the p'_i prime factors. The product of all elements in A_X is equal to $p^* \cdot \prod_{x_i\in X} p_i$. For the total product of all elements in S' to be equal to p^*, the product of the remaining elements in S' has to equal exactly $\prod_{x_i\in X} p_i^{-1}$. This can only be the case when the elements of C corresponding to these remaining elements in S' form an exact cover of X. Thus, the X3C instance admits a feasible cover. □

With the strong hardness of SubsetProduct on rational numbers, we can now prove our main result.

Theorem 7. TDEA$_\mathbb{Q}$ *with piecewise linear functions represented by a sequence of breakpoints is strongly NP-hard even when all input numbers are integers.*

Proof. Let $A = \{\frac{p_1}{q_1}, \dots, \frac{p_k}{q_k}\}$, $B = \frac{p^*}{q^*}$ be our SubsetProduct instance. First, we sort A in ascending order, i.e. $\frac{p_i}{q_i} \le \frac{p_{i+1}}{q_{i+1}}$. We then construct the TDEA$_\mathbb{Q}$ instance as follows: Similarly to the hardness proof for TDEA$_\mathbb{Z}$, we build a graph with $k+2$ vertices and two parallel edges e_i and e'_i between v_{i-1} and v_i for $1 \le i \le k$ and a final non-FIFO edge between v_k and t. All edges e'_i get constant travel time zero. The edges e_i get a time-dependent travel time function which has slope $\frac{p_i}{q_i}$ in the *arrival time representation*. The final edge has a travel time function with breakpoints $(0, p^*+1)$ and $(p^*, 0)$. The topology is the same as in the previous proof as illustrated in Fig. 1. We set $\tau^{\text{dep}} = q^*$ and $\tau^{\max} = p^*$. We pick the breakpoints for the e_i functions in such a way that the arrival time function has the correct slope at least for all times in $[0, \max(p^*, q^*)]$. Thus, with $s_i = \lceil \frac{\max(p^*, q^*)}{q_i} \rceil$ the edge e_i will have the arrival time function breakpoints $(0, 0)$ and $(s_i q_i, s_i p_i)$. This transformation has polynomial running time and also all numbers are polynomial bounded in the size of the SubsetProduct instance.

Some of these arrival time functions have slope < 1 and thus for $\tau \in [0, \tau^{\max}]$ we get $f^{\text{at}}(\tau) < \tau$. This means the corresponding travel time functions have negative travel times. We can obtain an equivalent instance without negative weights as follows: Let e_i be an edge with a breakpoint (x, y) in the travel time function where $y < 0$. We

[2] Garey and Johnson [12] state that SubsetProduct on positive integers is strongly NP-complete by reduction from ExactCoverBy3Sets with reference to private communication with A.C. Yao in 1978. This is somewhat surprising since there is no obvious reason why the pseudo-polynomial dynamic programming algorithm for SubsetSum should not be applicable to SubsetProduct. This strongly suggests that the problem is only weakly NP-complete and that the original reduction proved only weak hardness. Otherwise, this would prove P = NP. While we were unable to obtain the original proof, other realizations of the same reduction can be found online [16]. This reduction uses a number exponential in the input size for the target product B. Clearly, this admits a pseudo-polynomial algorithm. As it turns out, on integers, the problem is only weakly NP-hard. Johnson's NP-completeness column later contains a correction [15].

increase the travel times of both e_i and e'_i by the constant $c = -y$. Additionally, the departure times coordinates of breakpoints in every following travel time function must be increased by c, i.e. for (x, y) a breakpoint of a travel time function of an edge with head v_j where $j > i$, this breakpoint will be set to $(x + c, y)$. Also, $\tau^{\max}$ must be increased by c. Even though negative travel times are not necessary, we still use them during the rest of the proof because it simplifies the calculations. This also applies to travel times of zero which can be avoided similarly.

Suppose the SubsetProduct instance admits a subset A' such that the product of all elements in A' is B. We consider the st-path which uses e_i when $\frac{p_i}{q_i} \in A'$ and e'_i otherwise. When visiting vertex v_{i-1} at instant τ_{i-1} and traversing the edge e_i with arrival time breakpoints $(0, 0)$ and $(s_i q_i, s_i p_i)$ with $\tau_{i-1} \leq s_i q_i$, the arrival time at v_i can be computed by linear interpolation: $\tau_i = \tau_{i-1} \cdot \frac{p_i}{q_i}$. Assuming $\tau_{i-1} < s_i q_i$ for $1 \leq i \leq k$, the arrival time τ_k at the final vertex v_k before t can be computed as $\tau^{\text{dep}} \cdot \prod_{a_i \in A'} \frac{p_i}{q_i} = q^* \cdot \frac{p^*}{q^*} = p^*$. The final edge has a travel time of 0 at p^* which leads to an arrival of p^* at t which shows, that the TDEA$_{\mathbb{Q}}$ instance admits a feasible path given that our assumption on the τ_i holds. This assumption is valid because the elements of A and thus the slopes of the edges are ordered ascendingly. For all $\frac{p_i}{q_i} \leq 1$, $\tau_i \leq \tau_{i-1}$, i.e. the travel time is negative, and we arrive earlier than we started, so $\tau_i < q^*$ holds. Once edges with $\frac{p_i}{q_i} > 1$ are reached, the τ_i grow monotonically. But since the final result p^* is within the desired range, all intermediate τ_i have to be, too. Thus, the assumption holds for all τ_i.

Conversely, assume the TDEA$_{\mathbb{Q}}$ instance admits a path P with the desired arrival time. This is only possible when $\tau_k = p^*$ because before p^*, the arrival time function has a strictly negative slope and after p^* a strictly positive slope. With the same argument as in the previous paragraph, we get that all $\tau_i \leq \max(p^*, q^*)$. Hence, the following equation holds: $\tau^{\text{dep}} \cdot \prod_{a_i \in P} \frac{s_i p_i}{s_i q_i} = \tau^{\max} = p^*$ which is equivalent to $\prod_{a_i \in P} \frac{p_i}{q_i} = \frac{p^*}{q^*}$. Thus, the set $A' = \{\frac{p_i}{q_i} \mid a_i \in P\}$ is a fulfilling subset for the SubsetProduct instance. □

So far, we only discussed NP-hardness but not inclusion in NP. On first glance, it appears likely that the TDEA problem lies in NP because verifying path lengths takes running time linear time in the path length. However, this is not sufficient because solutions might have superpolynomial length in the input size. Consider a graph with two vertices s and t, one loop edge at s with constant travel time 1 and one st edge with travel time 0 at instant 2^k and travel time 2^{k+1} during the rest of the time as depicted in Fig. 2. For an instance with $\tau^{\text{dep}} = 0$ and $\tau^{\max} = 2^k$, the solution is a path which uses the loop at s 2^k times. A naive encoding of this solution would already be exponentially bigger than the (binary encoded) input. Even if it would be possible to show that a compact encoding of solutions is always possible, one would still need to find a way to evaluate the travel time of this path in polynomial running time in the input size. Whether this is possible likely strongly depends on the involved function classes. Therefore, the question remains open. Nevertheless, since

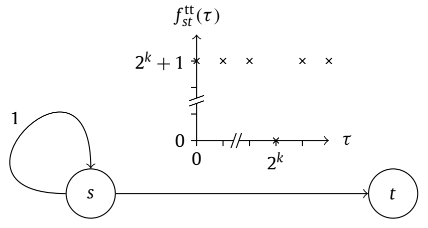

Fig. 2. Example graph with a shortest path with an exponential number of hops.

we only used acyclic graphs in our reductions, we can still conclude that when only allowing simple paths, the TDEA problem is NP-complete.

4. Conclusion

In this work, we presented a simple transformation from SubsetSum to the TDEA$_{\mathbb{Z}}$ problem which confirms its weak NP-hardness. Relating our proof to a recent result on the strong NP-hardness of SubsetSum on rational numbers showed that TDEA$_{\mathbb{Q}}$ on arbitrary rational functions is strongly NP-hard, too. Finally, we showed that the problem remains strongly NP-hard for a typical practical model when the travel times are piecewise linear functions defined on rational numbers but all coordinates are integers.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

I want to thank my colleague Jonas Sauer for many helpful discussions on proof ideas. I also want to thank my former colleague Lukas Barth for a great idea on a proof simplification. Finally, I want to thank my colleague Paul Jungeblut for proofreading a draft of this paper and his many helpful suggestions for improvements.

Funding

This research was funded by Karlsruhe Institute of Technology and the DFG project under grant number WA 654/23-2 in DFG research group 2083.

References

[1] S.E. Dreyfus, An appraisal of some shortest-path algorithms, Oper. Res. 17 (3) (1969) 395–412.
[2] M. Gendreau, G. Ghiani, E. Guerriero, Time-dependent routing problems: a review, Comput. Oper. Res. 64 (2015) 189–197.
[3] G.V. Batz, Time-Dependent Route Planning with Contraction Hierarchies, Ph.D. thesis, 2014.
[4] D. Delling, D. Wagner, Time-dependent route planning, in: R.K. Ahuja, R.H. Möhring, C. Zaroliagis (Eds.), Robust and Online Large-Scale Optimization, in: Lecture Notes in Computer Science, vol. 5868, Springer, 2009, pp. 207–230.

[5] G. Nannicini, D. Delling, L. Liberti, D. Schultes, Bidirectional A* search on time-dependent road networks, Networks 59 (2012) 240–251, Best Paper Award.

[6] B. Strasser, D. Wagner, T. Zeitz, Space-efficient, fast and exact routing in time-dependent road networks, Algorithms 14 (3) (January 2021), https://www.mdpi.com/1999-4893/14/3/90.

[7] A. Kleff, F. Schulz, J. Wagenblatt, T. Zeitz, Efficient route planning with temporary driving bans, road closures, and rated parking areas, in: S. Faro, D. Cantone (Eds.), Proceedings of the 18th International Symposium on Experimental Algorithms (SEA'20), in: Leibniz International Proceedings in Informatics, vol. 160, 2020.

[8] A. Orda, R. Rom, Shortest-path and minimum delay algorithms in networks with time-dependent edge-length, J. ACM 37 (3) (1990) 607–625.

[9] A. Orda, R. Rom, Minimum weight paths in time-dependent networks, Networks 21 (1991) 295–319.

[10] L. Foschini, J. Hershberger, S. Suri, On the complexity of time-dependent shortest paths, Algorithmica 68 (4) (2014) 1075–1097.

[11] A. Orda, R. Rom, Traveling without waiting in time-dependent networks is NP-hard, Tech. rep., Dept. Electrical Engineering, Technion-Israel Institute of Technology, 1989.

[12] M.R. Garey, D.S. Johnson, Computers and Intractability. A Guide to the Theory of NP-Completeness, W.H. Freeman and Company, 1979.

[13] R. Bellman, On a routing problem, Q. Appl. Math. 16 (1) (1958) 87–90.

[14] D. Wojtczak, On strong NP-completeness of rational problems, in: International Computer Science Symposium in Russia, Springer, 2018, pp. 308–320.

[15] D.S. Johnson, The NP-completeness column: an ongoing guide, J. Algorithms 2 (4) (1981) 393–405.

[16] Is the subset product problem NP-complete?, https://cs.stackexchange.com/a/27973. (Accessed 24 February 2022).