# Time-dependent shortest path with discounted waits

Jérémy Omer, Michael Poss

# Time-dependent shortest paths with discounted waits

Jérémy Omer*

*Univ Rennes, INSA Rennes, CNRS, IRMAR - UMR 6625, F-35000 Rennes, France*

Michael Poss†

*LIRMM, University of Montpellier, CNRS, Montpellier, France*

March 31, 2019

**Abstract**

We study a variant of the shortest path problem in a congested environment. In this setting, the travel time of each arc is represented by a piecewise continuous affine function of departure time. Besides, the driver is allowed to wait at nodes to avoid wasting time in traffic. While waiting, the driver is able to perform useful tasks for her job or herself, so the objective is to minimize only driving time. Although optimal solutions may contain cycles and pseudo-polynomially many arcs, we provide a representation of the solutions that is polynomial in the absolute value of the inverse of the slopes as well as in the dimensions of the graph. We further prove that the problem is $\mathcal{NP}$-Hard when the slopes are integer. We introduce a restriction of the problem where waits must be integer and propose pseudo-polynomial algorithms for the latter. We also provide a pseudo-FPTAS, polynomial in the ratio between the bound on the total waiting time and the minimum travel time. Finally, we discuss harder variants of the problem and show their inapproximability.

**keywords:** Time-dependent networks; Shortest paths, $\mathcal{NP}$-completeness, Approximation, Dynamic programming, Waiting.

## 1 Introduction

We are interested in the situation of a salesman who must travel from an origin $o$ to a destination $d$ in a potentially congested transport network. The variations in traffic conditions imply that the travel time between any two points in the network can vary over the day. Instead of losing time in the traffic, the salesman can avoid peak hours by delaying his departure or stopping at specific places to perform other activities that will be useful to him or his work. If he does not have a strong constraint on his arrival time and if he can value the time he spends waiting for better traffic conditions, only travel time is wasted. Therefore, the salesman's objective is to minimize total travel time.

Alternative applications of our model are varied. One example arises in truck transportation problems where some freight must be sent from a source to a destination at minimum cost. The

*jeremy.omer@insa-rennes.fr

†michael.poss@lirmm.fr

transportation cost is essentially driven by the fuel consumption along the trip. The latter is minimized by letting the drivers pause to avoid the most congested legs. Another example is related to organized tourist trips that last several days. Some destinations along these trips cannot be missed. On the other hand, there are many more choices of interesting and secondary spots that can be visited along the way. These spots represent the nodes where waiting is allowed, bounded by the time that can be spent in each of them. Overall, the tourism agency wishes to maximize the interest of the whole trip, reducing the time spent in congested traffic by organizing well-chosen stops.

## Literature review

Route planning in congested transport networks is classically formalized as the search for a shortest path in a time-dependent network. Due to this natural application, the time-dependent shortest path problem (TDSP) has been the subject of active research since its introduction by Cooke and Halsey in 1966 [4]. In particular, the study of the algorithmic complexity of TDSP has raised subtle issues from the beginning. For instance, Dreyfus [10] presented the first polynomial algorithm for the search of a shortest path in a graph where arc costs are positive and integer-valued functions of time. However, as discussed by Kaufman and Smith [18], this result implicitly assumes that the cost functions (i.e., the travel time functions) are *consistent*. This property, also referred to as FIFO (first-in first-out), expresses that one cannot reach the head of an arc earlier by departing later from its tail. Actually, it so happens that if we alleviate the consistency assumption, it is possible to build a time-dependent network where the shortest path is infinite, as shown by Orda and Rom [20]. Finiteness of the optimal paths can be guaranteed by imposing a positive constant lower bound to arc travel time functions, but Orda and Rom [19] observe that the problem remains $\mathcal{NP}$-Hard. This statement is proved by Sherali et al. [21] by reduction from the partition problem.

In close connection to the topic of consistency is that of waiting at the tail of an arc before traversing it. The possibility of waiting has first been investigated by Halpern [15] in inconsistent networks. In his work, inconsistency arises from the existence of time intervals where departing from a node is forbidden, and waits stand for parking periods. Orda and Rom [19] continued Halpern's work by considering two different waiting models: *unrestricted waiting*, and *source waiting*. In the unrestricted version, waiting is allowed everywhere without restriction, but the arrival time is minimized, which means that waiting time is counted. Interestingly, the problem can then be solved in polynomial time if sufficiently simple travel time functions are considered (e.g., piecewise linear functions with constant numbers of breakpoints). As argued by Foschini et al. [11], the problem is then equivalent to the consistent case, since one can always wait at the tail of an arc as long as it reduces the arrival time at the head of the arc.

In the source waiting model, waiting is forbidden everywhere except for the source node. This particular case raises the question of determining arrival time at destination as a function of departure time from source. Several theoretical works have been devoted to the study of this function [1, 5, 7, 11] for piecewise-linear arc travel time functions. In these articles, the authors deal only with consistent networks. Foschini et al. [11] then established that the arrival time function is a piecewise linear function that can have a superpolynomial number of pieces, even if the total number of pieces in arc travel time functions is bounded. More specifically, if the total number of pieces in the arc functions is $K$ and $n$ is the number of nodes in the graph, then the complexity of the arrival time function is at most $Kn^{\log(\Theta(n))}$. Foschini et al. show that at most $K$ breakpoints of this function correspond to those of the arc functions, while the others are intersection of piecewise-linear functions. From this result, they establish that despite the superpolynomial complexity of the arrival time function, it is polynomial to determine the departure time that minimizes total travel time.

Cai et al. [2] considered models where each arc has a time-dependent cost, in addition to the time-dependent traversal time. What is more, they allowed waiting at every node, incurring waiting costs described by time-dependent functions. They provided dynamic programming algorithms, later improved in [3] and [6].

An even larger literature about TDSP has focused on practical solution methods (e.g., [8, 9, 17]). The effort on practical implementation issues is motivated by the need to get optimal paths in large-scale time-dependent networks in only a fraction of a second. This is necessary both for the usual needs of web applications for route planning and to solve more complex problems such as vehicle routing problem in a time-dependent network. A recent review of these works is available in [14].

## Our contributions

To answer the salesman problem presented above, we consider a TDSP in a consistent network where waiting is allowed and arc travel time functions are continuous and piecewise linear. In contrast to the unrestricted case introduced by Orda and Rom [19], the total wait is bounded and each waiting period at a node is also bounded. More importantly, we do not seek to minimize the arrival time at destination but the total travel time. Stated otherwise, the waiting time is discounted from the arrival time at destination. This can be seen as a generalization of the problem studied by Foschini et al. [11], where they also seek a minimum travel time path, but waiting is restricted to the origin. Our model is also related to one of the models studied in [2, 3, 6], were waiting at each node is bounded, with the following differences. First, in addition to local waiting constraints, we also consider a global constraint. Second, [2, 3, 6] focus only on integer values for the waiting times, while we consider both integer and continuous values for these. Last, the model of [2, 3, 6] considers paths having lengths not greater than a given time horizon $T$ (and provide algorithms polynomial in $T$), while no such restriction appears in our model.

The main focus of this article is on the complexity of the TDSP with discounted waits. Our findings reveal that this small shift from classical assumptions yields complex issues with that respect.

Our first result is in the observation that the problem may not be in $\mathcal{NP}$, since optimal paths may contain a number of arcs that is not polynomially bounded. Nevertheless, we show that the optimal paths have compact representations if the right derivatives are not too small. Using a reduction from the partition problem, we then show that the problem is $\mathcal{NP}$-Hard if the graph is acyclic and the slopes in the pieces of arc travel time functions are restricted to the values $-1$, 0 and 1.

A more detailed study leads us to consider both integer and real-valued waiting times. We show that the two problems are equivalent if the slopes in arc cost functions are all integer. But a counter-example highlights that this is not true in general, even if the intervals defining the pieces of the arc travel time functions are integer. Nevertheless, we can bound the ratio between the optimal values of the two problems if the arc functions are lower bounded by a common value $C_{min}$. This result is meaningful, because we develop a label setting algorithm that finds an optimal path in pseudo-polynomial time if waiting times are restricted to integer values. If $n$ and $m$ are the numbers of nodes and arcs in the graph, the algorithm runs in $\mathcal{O}(mW^2 + nW \log(nW))$ time. We also observe that an arbitrarily high precision can be obtained with this integer approximation by increasing the granularity of the time discretization.

As an alternative to this approach, we develop a pseudo-FPTAS having a complexity polynomial in the ratio of $W$ to $C_{min}$. The scheme is valid for both integer and real-valued waiting times.

We conclude our study with an analysis of two harder variants of the problem. In the first variant, we relax the consistency assumption, and in the second one, we consider a stronger constraint on the waiting times at each node. In both cases, the problem ends up being not approximable, and we show the strong $\mathcal{NP}$-Hardness of the latter case.

### Organization of the article

The remainder of the article is organized as follows. We formally define the problem, and introduce our notations in Section 2. We propose a compact representation of the solutions in Section 3, and show that the problem is $\mathcal{NP}$-Hard in Section 4. In Section 5 we highlight the impact of restricting waiting times to integer values. Section 6 is then devoted to the pseudo-polynomial algorithm for integer waits, and the pseudo-FPTAS is studied in Section 7. Our last results about harder variants of the problem are presented in Section 8.

## 2 Problems definition and notations

We consider a directed graph $G = (V, A)$ with $n$ nodes and $m$ arcs, $m \geq n$, containing two origin and destination nodes, $o$ and $d$. The bound on total waiting time is denoted as $W \in \mathbb{Z}$ and the bound on the waiting time at each node is denoted as $W_v \in \mathbb{Z}$, $\forall v \in V$. The travel time of each arc is given by a positive continuous piecewise linear function $C_e : \mathbb{R}^+ \to \mathbb{R}^+$ defined by $r_e$ pieces plus a final constant piece. Each piece $s = 1, \ldots, r_e$ is an affine function $c_e^s + \rho_e^s t$ defined on the interval $[\tau_e^{s-1}, \tau_e^s]$, where $c_e^s \in \mathbb{Z}$, $\rho_e^s \in \mathbb{Q}$ and $\tau_e^0 = 0$. Let $r_{max} = \max_{e \in E} r_e$ be the maximum number of pieces in the arc travel time functions. We also denote by the minimum and the maximum values of all travel times functions by $C_{min} = \min_{e,t} C_e(t)$ and $C_{max} = \max_{e,t} C_e(t)$, respectively. To simplify the exposure of some results of this paper, we assume throughout that $C_e(t)$ is never smaller than 1.

**Assumption 1.** $C_{min} \geq 1$.

Let $\mathcal{P}_v$ be the set of all paths from $o$ to $v \in V$. We study herein the cost of $(p, w)$, where $p = (v_1(= o), v_2, \ldots, v_{|p|}(= d))$ belongs to $\mathcal{P}_d$, and $w = (w_1, \ldots, w_{|p|})$ is the vector of waiting times which belongs to

$$\mathcal{W}(p, W) = \left\{ w \in \mathbb{R}_+^{|p|} : \sum_{k=1}^{|p|} w_k \leq W,\ w_k \leq W_{v_k}, k = 1, \ldots, |p| \right\}.$$

When the path $p = (v_1, v_2, \ldots, v_{|p|})$ is fixed, we introduce the notation $C_k := C_{v_k v_{k+1}}$. Since we wish to minimize total travel time, the cost of $(p, w)$ can be expressed as

$$T(p, w) = \sum_{k=1}^{|p|-1} C_k(t_k), \tag{1}$$

where $t_k$ is the departure time from node $v_k$; it is computed recursively as

$$t_k = t_{k-1} + C_{k-1}(t_{k-1}) + w_k. \tag{2}$$

Notice that $T(p, w)$ for $p \in \mathcal{P}_d$ is different from $t_{|p|}$, which also accounts for the total waiting time up to node $d$. More precisely, the two quantities are related through the equation

$$T(p, w) = t_{|p|} - \sum_{k=1}^{|p|} w_k. \tag{3}$$

The aim of the article is to study the complexity of the following two optimization problems.

$$\text{opt} = \min_{p \in \mathcal{P}_d} \min_{w \in \mathcal{W}(p,W)} T(p,w), \tag{WTDSP}$$

$$\text{opt}^I = \min_{p \in \mathcal{P}_d} \min_{w \in \mathcal{W}^I(p,W)} T(p,w), \tag{WTDSP$^I$}$$

where $\mathcal{W}^I(p,\omega) = \mathcal{W}(p,\omega) \cap \mathbb{Z}^{|p|}$ contains only integer waiting times.

In most applications, we can assume that arc travel times are consistent: if we enter an arc $e$ at time $t \leq t'$, then we leave the arc at time $t + \mathcal{C}_e(t) \leq t' + \mathcal{C}_e(t')$. Hence, unless stated otherwise, we assume throughout the paper that consistency holds. For the particular travel time functions we consider, this assumption can also be stated as follows. We suppose the following holds throughout the paper, apart from Proposition 8 from Section 8.

**Assumption 2** (Consistency assumption)**.** *Let $\rho_{min} = \min_{e,s} \rho_e^s$. We have $\rho_{min} \geq -1$.*

We shall sometime refer to the right-derivative of a one-variable function $f(x)$ as $\partial_+ f(x)$.

# 3 Compact representation

Optimal solutions to WTDSP and WTDSP$^I$ may contain cyclic paths where one waits several times at the same node $v$ provided that each waiting time does not exceed the upper bound $W_v$. In fact, we can construct examples in which the optimal path traverses a node $W$ times. One such example is depicted in Figure 1. The optimal solution goes $W$ times through the loop on node 1 to wait 1 at every stop at the node. This wait is made profitable by the decreasing travel time on arc $(1,d)$. This implies that the problem may actually not be in $\mathcal{NP}$ since it may not be possible to compute the objective function in polynomial time. Of course, if the graph is acyclic, the previous situation never happens.

**Observation 1.** *If $G$ is acyclic,* WTDSP *is in $\mathcal{NP}$.*

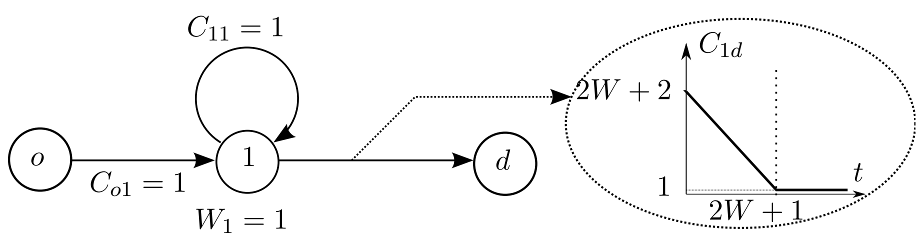


Figure 1: Instance where the optimal path traverses $W$ arcs.

Below we introduce a compact representation for candidate solutions, which allows us to verify their feasibility and compute their objective values in polynomial time, given that $\max_{e,s:|\rho_e^s|>0} 1/|\rho_e^s|$ is constant.

**Theorem 1.** *Let $\lambda = \max_{e,s:|\rho_e^s|>0} \frac{1}{|\rho_e^s|}$. The feasibility of a candidate solution $(p,w)$ and its cost can be computed in $\mathcal{O}(\lambda r_{max}^2 m^3 \log(C_{max}))$ operations.*

Before proving the theorem, we introduce two technical results that bound the number of times any piece $s$ of an arc $e$ such that $\rho_e^s \neq 0$ can be hit along a path.

**Lemma 1.** *Let $(p, w)$ be a solution of* WTDSP *and assume that there exist $1 \leq k_1 < k_2 < \cdots < k_q < |p|$ such that $(v_{k_i}, v_{k_i+1}), i = 1, \ldots, q$, are different repetitions of the same arc $e$ and $\tau_e^{s-1} \leq t_{k_i} < \tau_e^s, \forall i \in \{1, \ldots, q\}$. The following holds:*

$$t_{k_q+1} \geq t_{k_1} + (c_e^s + \rho_e^s t_{k_1}) \sum_{i=0}^{q-1} (1 + \rho_e^s)^i.$$

*Proof.* We prove the result by induction on $l$. First assume that $q = 1$. We have

$$t_{k_1+1} = t_{k_1} + C_e(t_{k_1}) + w_{k_1+1} \geq t_{k_1} + c_e^s + \rho_e^s t_{k_1}.$$

Hence the result holds.

Now, assume that the result holds for some index $q - 1 \geq 1$ and assume that there is a $q^{\text{th}}$ repetition of $e$ such that $\tau_e^{s-1} \leq t_{k_q} < \tau_e^s$. By definition of the travel time on arc $e$, we have

$$t_{k_q+1} = t_{k_q} + c_e^s + \rho_e^s \times t_{k_q} + w_{k_q+1}.$$

To use the induction hypothesis, we rewrite this equality as

$$\begin{aligned} t_{k_q+1} - t_{k_1} &= t_{k_q} - t_{k_1} + c_e^s + \rho_e^s \times t_{k_1} + \rho_e^s (t_{k_q} - t_{k_1}) + w_{k_q+1} \\ &= (1 + \rho_e^s)(t_{k_q} - t_{k_1}) + c_e^s + \rho_e^s t_{k_1} + w_{k_q+1} \end{aligned}$$

By non-negativity of travel and waiting times, we then notice that $t_{k_q} \geq t_{k_{q-1}+1}$ and $w_{k_q+1} \geq 0$, so the above yields

$$t_{k_q+1} - t_{k_1} \geq (1 + \rho_e^s)(t_{k_{q-1}+1} - t_{k_1}) + c_e^s + \rho_e^s t_{k_1}$$

We can then use the induction hypothesis to obtain

$$t_{k_q+1} - t_{k_1} \geq (c_e^s + \rho_e^s t_{k_1}) \times \left( (1 + \rho_e^s) \sum_{i=0}^{q-2} (1 + \rho_e^s)^i + 1 \right) = (c_e^s + \rho_e^s t_{k_1}) \sum_{i=0}^{q-1} (1 + \rho_e^s)^i.$$

□

For a given solution $(p, w)$ where $v_k$ is the $k^{\text{th}}$ node on $p$, we say that $e = (v_k, v_{k_1})$ is non-constant if $\tau_e^{s-1} \leq t_k < \tau_e^s$ and $\rho_e^s \neq 0$, and we call $s$ a non-constant piece of $e$. We similarly say that $e$ is constant if $\rho_e^s = 0$. Let us introduce the notation $C_e^s = C_e(\tau_e^s)$.

**Lemma 2.** *Let $(p, w)$ be a solution of* WTDSP*, and $e$ a non-constant arc of $p$. If $\rho_e^s = -1$, piece $s$ of $e$ cannot appear more than once in $p$. Otherwise, the non-constant piece cannot appear in $p$ more than $\left\lfloor \frac{\log(C_e^s) - \log(c_e^s)}{\log(1+\rho_e^s)} \right\rfloor$ times.*

*Proof.* Let $(p, w)$ be a solution of WTDSP, and $e \in p$. We first assume that $\rho_e^s = -1$ and that there exists $1 \leq k < |p|$ such that $(v_k, v_{k+1}) = e$ and $\tau_e^{s-1} \leq t_k < \tau_e^s$. Recall the assumption that $C_e(t) \geq 1$ for all $t$, which in turn implies that $c_e^s > \tau_e^s$. By definition of the travel time function of $e$, we get

$$t_{k+1} = t_k + C_e(t_k) + w_k = t_k + c_e^s - t_k + w_k > \tau_e^s,$$

so the result holds.

Let $1 \leq k_1 < \cdots < k_{q+1} < |p|$ be such that $(v_{k_i}, v_{k_i+1}), i = 1, \ldots, q+1$, are $q+1$ occurrences of piece $s$ of $e$ and suppose $\rho_e^s > -1$. In particular, this means that $t_{k_{q+1}} \leq \tau_e^s$. Since $t_{k_q+1} \leq t_{k_{q+1}}$, Lemma 1 implies that this inequality can be satisfied only if

$$
\begin{aligned}
& t_{k_1} + (c_e^s + \rho_e^s t_{k_1}) \sum_{i=0}^{q-1} (1+\rho_e^s)^i \leq \tau_e^s \\
\Leftrightarrow\ & t_{k_1} + (c_e^s + \rho_e^s t_{k_1}) \frac{1-(1+\rho_e^s)^q}{-\rho_e^s} \leq \tau_e^s \\
\Leftrightarrow\ & t_{k_1} - \frac{c_e^s}{\rho_e^s} - t_{k_1} + (c_e^s + \rho_e^s t_{k_1}) \frac{(1+\rho_e^s)^q}{\rho_e^s} \leq \tau_e^s \\
\Leftrightarrow\ & (c_e^s + \rho_e^s t_{k_1}) \frac{(1+\rho_e^s)^q}{\rho_e^s} \leq \tau_e^s + \frac{c_e^s}{\rho_e^s} \\
\Leftrightarrow\ & (c_e^s + \rho_e^s t_{k_1}) \frac{(1+\rho_e^s)^q}{\rho_e^s} \leq \frac{C_e^s}{\rho_e^s},
\end{aligned}
$$

where the last inequality follows from $c_e^s + \tau_e^s \rho_e^s = C_e^s$. Next, if $\rho_e^s < 0$, we get that $t_{k_{q+1}} \leq \tau_e^s$ only if

$$(1+\rho_e^s)^q \geq \frac{C_e^s}{c_e^s + \rho_e^s t_{k_1}} \geq \frac{C_e^s}{c_e^s}. \tag{4}$$

Similarly, if $\rho_e^s > 0$, we get that $t_{k_{q+1}} \leq \tau_e^s$ only if

$$(1+\rho_e^s)^q \leq \frac{C_e^s}{c_e^s + \rho_e^s t_{k_1}} \leq \frac{C_e^s}{c_e^s}. \tag{5}$$

Taking the logarithm of both sides of (4) and (5) yields the result. □

We are now able to prove the main result of this section.

*Proof of Theorem 1.* Let $e \in E$ and $s \in \{1, \ldots, r_e\}$. If $-1 < \rho_e^s < 0$, then by Assumption 1 and definition of $\lambda$,

$$\frac{\log(C_e^s) - \log(c_e^s)}{\log(1+\rho_e^s)} \leq -\frac{\log(c_e^s)}{\rho_e^s} \leq \lambda \log(c_e^s) \leq \lambda \log(C_{max}) \tag{6}$$

If $\rho_e^s > 0$, we similarly obtain $\frac{\log(C_e^s) - \log(c_e^s)}{\log(1+\rho_e^s)} \leq \lambda \log(C_e^s) \leq \lambda \log(C_{max})$. For a reasonable encoding of travel times, Lemma 2 thus ensures the solutions of WTDSP include a polynomial (in $\lambda$) number of non-constant arcs.

Consider some solution $(p, w)$: $p$ can then be described as the concatenation of a polynomial number of paths $(p_1, \ldots, p_q)$, where only the last arc of $p_i$ can be non-constant for all $i = 1, \ldots, q$. Let $\sigma_e^\ell \in \{1, \ldots, r_e + 1\}$ index the $\ell$-th constant piece of $C_e$ for each $\ell = 1, \ldots, L_e$. Notice that $L_e \geq 1$ since the last piece of each travel time function is constant; denote $L_{max} = \max_{e \in E} L_e$. Let $i \in \{1, \ldots, q\}$, and denote by $(v_{i_1}, \ldots, v_{i_q})$ the sequence of nodes in $p_i$. If an arc $e$ appears in position $k \leq |p_i| - 2$ of $p_i$, it is constant. Since the arrival time at the nodes of a path increases along the path, it is possible to identify $L_e + 1$ positions $k_e^1 \leq k_e^2 \leq \cdots \leq k_e^{L_e+1} \leq |p_i|$, such that

1. $e$ first appears in $p_i$ at position $k_e^1$, and it last appears at position $k_e^{L_e+1} - 1$;
2. each occurrence $k$ of $e$ in piece $\sigma_e^\ell$ is such that $k_e^{\sigma_e^\ell} \leq k < k_e^{\sigma_e^\ell + 1}$.

As a consequence, if we sort $\bigcup_{e \in p_i} \bigcup_{\ell \in \{1,\ldots,L_e+1\}} \{k_e^\ell\}$ by ascending order, we can describe $p_i$ as the concatenation of at most $mL_{max}$ subpaths $p_i^1, \ldots, p_i^r$ and one non-constant arc such that, for all $j \in \{1, \ldots, r\}$, every arc $e \in p_i^j$ has the same travel time every time it appears in $p_i^j$.

The above means that each subpath can be described with one array of size $\mathcal{O}(r_{max}m)$ containing the number of times each arc appears in the subpath, and the waiting times on the subpath can be stored in another array of size $\mathcal{O}(n)$ containing the total waiting time at each node of the subpath. The travel and waiting times over a subpath are then computed in $\mathcal{O}(m)$ operations, so that the travel and waiting times along $p_i$, $i = 1, \ldots, q$, are computed in at most $\mathcal{O}(r_{max}m^2)$ operations. By Lemma 2 and (6), $q \leq \lambda r_{max} m \log(C_{max})$, so the travel time of $p$ can be computed in at most $\mathcal{O}(\lambda r_{max}^2 m^3 \log(C_{max}))$ operations.

Reciprocally, let $(p, w)$ be a path described as the concatenation of a polynomial number of subpaths with waiting times separated by isolated arcs. Assume also that the subpaths are represented by two tables, one with the number of times each arc is followed and one with the total waiting time over each node of the subpath. The flow conservation constraints and the bounds on waiting time can be checked in $\mathcal{O}(m)$ operations on each subpath. It is also possible to check that the travel time of every arc of a subpath remains constant over the subpath by computing travel time at the start and at the end of the subpath. This is done in $\mathcal{O}(r_{max}m)$ operations. As a consequence, the validity of $(p, w)$ can be verified in a polynomial number of operations. □

# 4 Hardness

We prove below that WTDSP and WTDSP$^I$ are hard, following a polynomial reduction from the classical number partitioning problem [13] (PARTITION). It is well known that PARTITION is $\mathcal{NP}$-Complete, but it can be solved by an algorithm whose execution time is a polynomial in the sum of the weights of the items involved.

**Proposition 1.** *The decision versions of* WTDSP *and* WTDSP$^I$ *are $\mathcal{NP}$-Complete if $G$ is acyclic and $\rho_e^s \in \{-1, 0, 1\}$ for all $e$ and $s$.*

*Proof.* We consider below the case of WTDSP, the proof being similar for WTDSP$^I$. From Theorem 1, WTDSP is in $\mathcal{NP}$. We reduce PARTITION to the decision version of WTDSP in an acyclic graph to show that the latter problem is $\mathcal{NP}$-Complete.

Consider an instance of PARTITION characterized by $n$ integers $a_1, \ldots, a_n$ such that $\sum_{i=1}^n a_i = 2A$. We build the corresponding instance of WTDSP by considering a graph $G = (V, E)$ such that

$$\begin{aligned} V =& \{o, d\} \cup \{1, \ldots, n\} \cup \{1^W, \ldots, n^W\} \\ E =& \{(o, 1), (o, 1^W), (1^W, 1), (n, d)\} \cup \bigcup_{i=1}^{n-1} \{(i, i+1), (i, (i+1)^W), ((i+1)^W, (i+1))\}, \end{aligned}$$

where $o$ and $d$ are the origin and destination nodes, and $\{1^W, \ldots, n^W\}$ are waiting nodes. The travel time of the arcs $(i^W, i)$ are constant and equal to $A + a_i$ for $i = 1, \ldots, n$, and the arc $(n, d)$ has a travel time function given by $C_{nd}(t) = A + |t - 2(n+1)A|$. The upper arcs have a travel time equal to $2A$ and the travel times of $(o, 1^W)$ and of the arcs $(i-1, i^W)$ are constant and equal to $A$ for $i = 2, \ldots, n$. The maximum total waiting time is $W = A$; the node maximum waiting times are $W_{i^W} = a_i, \forall i \in \{1, \ldots, n\}$, and zero for every other node. Figure 2 illustrates the instance of WTDSP described above.

For any solution $(p, w)$ of the above instance of WTDSP, we denote $I = \{i : (i^W, i) \in p\}$. We get a solution of PARTITION from a solution of WTDSP by summing the integers $a_i, i \in I$.

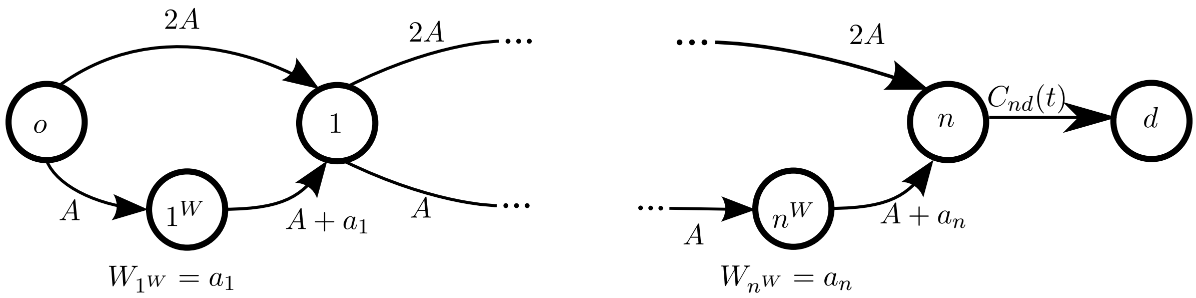


Figure 2: Reduction from an instance of PARTITION to WTDSP

Departure time from $n$ is then given by $2nA + \sum_{i \in I}(a_i + w_i)$, so the expression of $C_{nd}$ yields

$$T(p, w) = 2nA + \sum_{i \in I} a_i + A + \left| \sum_{i \in I}(a_i + w_i) - 2A \right|.$$

One readily verifies that $T(p, w) = (2n + 2)A$ if $\sum_{i \in I} a_i = A$ and $\sum_{i \in I} w_i = A$, and $T(p, w) > (2n + 2)A$ if $\sum_{i \in I} a_i > A$. If $\sum_{i \in I} a_i < A$, the maximum waiting times at the nodes involve that $\sum_{i \in I} w_i \leq \sum_{i \in I} a_i$, so

$$T(p, w) \geq (2n + 1)A + \sum_{i \in I} a_i + 2A - 2 \sum_{i \in I} a_i > (2n + 2)A.$$

As a conclusion, there exists a subset $I \subset \{1, \ldots, n\}$ such that $\sum_{i \in I} a_i = A$ if and only if the instance of WTDSP has a solution with total travel time equal to $(2n + 2)A$, which shows that the decision version of WTDSP is $\mathcal{NP}$-Complete. $\square$

## 5 Fractionality matters

In this section, we study the relation between the optimal solutions of problem WTDSP and its restriction to integer waiting times, WTDSP$^I$. While one can expect that the solutions to both problems are different in general, the following example shows that the ratio between their optimal values can be as large as $3/2$.

**Example 1.** *Consider the simple example drawn on Figure 3. One readily verifies that the optimal solution waits $M - 1$ at node $o$ and $\frac{M-1}{M}$ at node 1, yielding a total travel time of $2 + \frac{1}{M}$. Moreover, the cheapest integer solution waits $M - 1$ at node $o$ and 0 node 1, and it has a total travel time equal to 3. Hence, for any $\epsilon > 0$, the ratio between the two solutions is equal to $3/(2 + \frac{1}{M})$, which is not smaller than $3/2 - \epsilon$ if $M$ is large enough.*

The instance used in Example 1 is particularly pessimistic because its lowest travel times values are equal to 1. We show next that the ratio between the two objectives decreases with the increase of the minimum value of the travel time functions.

**Proposition 2.** *Let $C_{min} = \inf_{e \in E, t \geq 0} C_e(t) \geq 1$. Then,* $\text{opt}^I \leq (1 + \frac{1}{C_{min}})\,\text{opt}$.

To prove the result, we first show that the decreasing rates of $T(p, w)$ along each variable $w_i$ are bounded below by $-1$. Let $e^k$ denote the vector with value 1 in coordinate $k$ and 0 otherwise.

**Lemma 3.** *Let $v \in V$. For any $p \in \mathcal{P}_v$ and $\delta \geq 0$, $T(p, w + \delta e^k) \geq T(p, w) - \delta$.*

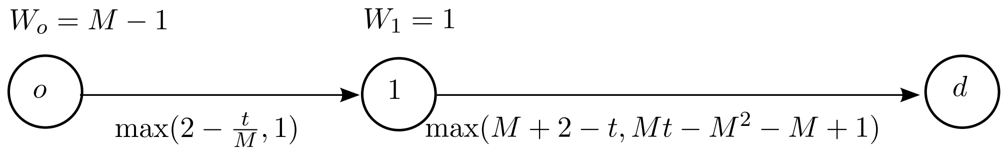


Figure 3: Instance of WTDSP with a fractional solution.

*Proof.* Let $p \in \mathcal{P}_v$ and let $t$ and $t'$ be the vectors of departure times for $w$ and $w + \delta e^k$, respectively. By definition, $t'_k = t_k + \delta$ and, since $\rho_{min} \geq -1$, we see by induction that $t_\ell \leq t'_\ell$ for all $\ell \in \{k, \dots, |p|\}$. In particular, $t_{|p|} \leq t'_{|p|}$, and the result follows from (3). □

*Proof of Proposition 2.* Let $(p^*, w^*)$ be the optimal solution to WTDSP and $(p^*, \lfloor w^* \rfloor)$ a feasible solution to WTDSP$^I$. From the definition of $C_{min}$ we have $|p^*| - 1 \leq \frac{\text{opt}}{C_{min}}$. Hence, we obtain

$$\text{opt}^I \leq T(p^*, \lfloor w^* \rfloor) \leq T(p^*, w^*) + |p^*| - 1 \leq \text{opt} + \frac{1}{C_{min}} \text{opt} = (1 + \frac{1}{C_{min}}) \text{opt}\,.$$

□

Proposition 2 shows that the error made by restricting ourselves to integer values can be quite small for practical instances. Integer solutions may also happen to be optimal in the continuous variant. There are some cases in which we can predict that this will happen, as the instance used in the proof of Proposition 1. A particularity of this instance is that all slopes of functions $C_e$ are integer. Next, we prove that when this happens, WTDSP always has an integer optimal solution.

**Proposition 3.** *If $\rho_e^s \in \mathbb{Z}$ for all $e$ and $s$, there exists an optimal solution to* WTDSP *with integer waiting times.*

*Proof.* Let $p = (v_1(= o), v_2, \dots, v_{|p|}(= d))$ be an optimal path and $w \in \mathcal{W}(p, w)$ the optimal waiting times. Notice first that Assumption 2 implies that $\rho_e^s = -1$, $\rho_e^s = 0$ or $\rho_e^s \geq 1$ for each $e \in E$ and $s = 1, \dots, r_e$. Let us recall that

$$T(p, w) = \sum_{k=1}^{|p|-1} C_k(t_k(w)), \tag{7}$$

where $t_k(w)$ is the departure time from node $v_k$ using waiting times $w$. Suppose first that $w_\ell$ is the unique fractional value of $w$ along $p$ and let $\rho = \partial_+ C_e(t_\ell(w))$ for $e = (v_\ell, v_{\ell+1})$. Three cases occur, depending on the value of $\rho$.

**$\rho = -1$.** Define $w' = \lceil w \rceil$. These waiting times are also feasible, because every maximum waiting time is integer.

Now, $t_k(w) = t_k(w')$ for each $k < \ell$, so that $C_k(t_k(w')) = C_k(t_k(w))$ for $k < \ell$. What is more, $C_\ell(t_\ell(w')) = C_\ell(t_\ell(w)) - (\lceil w_\ell \rceil - w_\ell)$. This means that $C_\ell(t_\ell(w')) \leq C_\ell(t_\ell(w))$ and $t_{\ell+1}(w') = t_{\ell+1}(w)$. From the latter, we get $C_k(t_k(w')) = C_k(t_k(w))$ for $k > \ell$, which, combined to the former, yields $T(p, w') \leq T(p, w)$.

**$\rho \geq 1$.** We define $w' = \lfloor w \rfloor$. We have

$$C_k(t_k(w')) = C_k(t_k(w)) \tag{8}$$

for all $k \in \{1, \dots, \ell - 1\}$, and

$$C_\ell(t_\ell(w')) = C_\ell(t_\ell(w)) - \rho(w_\ell - \lfloor w_\ell \rfloor). \tag{9}$$

Applying Lemma 3 to the path $(v_{\ell+1}, \dots, v_{|p|})$, we also have

$$\sum_{k=\ell+1}^{|p|-1} C_k(t_k(w')) \leq \sum_{k=\ell+1}^{|p|-1} C_k(t_k(w)) + w_\ell - \lfloor w_\ell \rfloor. \tag{10}$$

Combining (8), (9) and (10), we obtain $T(p, w') \leq T(p, w)$.

**$\rho = 0$.** Let $\ell' \in \{\ell + 1, \dots, |p|\}$ index the first node along $p$ such that $t_{\ell'}$ lies in a non-constant piece of $C_{e'}$, where $e' = (v'_\ell, v_{\ell'+1})$. If $\rho^s_{e'} = -1$, define $w' = \lceil w \rceil$ so that $t_k(w') = \lceil t_k(w) \rceil$ for each $k \leq \ell'$. Otherwise, define $w' = \lfloor w \rfloor$ so that $t_k(w') = \lfloor t_k(w) \rfloor$ for each $k \leq \ell'$. Then, notice that because the breakpoints are integer and $\partial_+ C_k(t_k(w)) = 0$ for all $k \in \{\ell, \dots, \ell' - 1\}$, we have

$$C_k(t_k(w)) = C_k(\lceil t_k(w) \rceil) = C_k(\lfloor t_k(w) \rfloor),\ \forall k \in \{\ell, \dots, \ell' - 1\}. \tag{11}$$

The result follows by combining (11) with the analysis of the above cases applied to $v_{\ell'}$ instead of $v_\ell$ to study $\sum_{k=\ell'}^{|p|-1} C_k(t_k(w'))$.

Suppose now that there are two or more components of $w$ that are fractional, denoted $\{\ell_1, \dots, \ell_s\}$, and let $\rho = \partial_+ C_e(t_{\ell_1}(w))$. In what follows, we build a solution that has a smaller number of fractional components and whose travel time is not greater than that of the original solution. Repeating the procedure yields an optimal integer solution.

**$\rho = -1$.** Define $w'$ by transferring the fractional part from component $\ell_2$ to component $\ell_1$ until $w'_{\ell_1} = \lceil w_{\ell_1} \rceil$ or $w'_{\ell_2} = \lfloor w'_{\ell_2} \rfloor$. We see similarly as above that $T(p, w') \leq T(p, w)$.

**$\rho \geq 1$.** Define $w'_{\ell_1} = \lfloor w_{\ell_1} \rfloor$ and $w'_k = w_k$ for $k \neq \ell_1$. Using the same argument as above, we obtain $T(p, w') \leq T(p, w)$.

**$\rho = 0$.** Let us define $\rho_k = \partial_+ C_e(t_k(w))$ for each $\ell_1 < k < \ell_2$. We consider three sub-cases. First, if $\rho_k = 0$ for each $\ell_1 < k < \ell_2$, then we define $w'$ by transferring the fractional part from component $\ell_1$ to component $\ell_2$ until $w'_{\ell_2} = \lceil w_{\ell_2} \rceil$ or $w'_{\ell_1} = \lfloor w'_{\ell_1} \rfloor$. Only the departure times of nodes in $\{v_{\ell_1}, \dots, v_{\ell_2 - 1}\}$ are affected and these do not cross any of the (integer) breakpoints of their respective cost functions.

For the other two sub-cases, we define $k' = \min\{k : \ell_1 < k < \ell_2, \rho_k \neq 0\}$. If $\rho_{k'} > 0$, we define $w'$ as in the first sub-case. Otherwise, $\rho_{k'} < 0$, and we define $w'$ by transferring the fractional part from component $\ell_2$ to component $\ell_1$ until $w'_{\ell_1} = \lceil w_{\ell_1} \rceil$ or $w'_{\ell_2} = \lfloor w'_{\ell_2} \rfloor$.

□

## 6 Pseudo-polynomial-time algorithms

We now show how WTDSP$^I$ can be solved in pseudo-polynomial time, yielding also a pseudo-FPTAS for WTDSP. For any $v \in V$, let $T^I_v(\omega)$ be the minimum travel time among all paths from $o$ to $v$ whose waiting times are integer and total waiting times (not including the wait at node $v$) are equal to $\omega$:

$$T^I_v(\omega) = \min_{p \in \mathcal{P}_v} \min_{w \in \mathcal{W}^I(p, \omega)} T(p, w). \tag{12}$$

In particular, $T_o^I(0) = 0$ and $T_o^I(\omega) = +\infty$ for $\omega > 0$. The main idea of the algorithm described in this section relies on defining $T_v^I(\omega)$ recursively along candidate paths. Considering a particular path $p = (v_1, \ldots, v_{|p|}) \in \mathcal{P}_d$ and $\omega \in \mathbb{Z}_+$, we have

$$T_{v_k}^I(\omega) = \min_{\nu \in [(\omega - W_{v_{k-1}})^+, \omega] \cap \mathbb{Z}} \left\{ T_{v_{k-1}}^I(\nu) + C_{k-1}(T_{v_{k-1}}^I(\nu) + \omega) \right\}, \tag{13}$$

where $\nu \in [(\omega - W_{v_{k-1}})^+, \omega] \cap \mathbb{Z}$ guarantees that the wait at node $v_{k-1}$ is at most $W_{v_{k-1}}$. Combining the above recursion with a dynamic search for the optimal path, we obtain a dynamic programming (DP) recursion:

$$T_v^I(\omega) = \begin{cases} \min\limits_{(u,v) \in E} \left\{ \min\limits_{\nu \in \{(\omega - W_u)^+, \ldots, \omega\}} \left\{ T_u^I(\nu) + C_{uv}(T_u^I(\nu) + \omega) \right\} \right\} & v \neq o \\ 0 & v = o,\ \omega = 0 \\ +\infty & v = o,\ \omega \neq 0 \end{cases} \tag{14}$$

To introduce a pseudo-polynomial-time algorithm, we first describe a simpler version of the problem that assumes that $G$ is acyclic. Observe that the reduction of PARTITION we made in the proof of Proposition 1 involves an acyclic graph, so this assumption maintains $\mathcal{NP}$-Completeness.

**Proposition 4.** *If $G$ is acyclic,* $\mathrm{opt}^I$ *can be computed in* $\mathcal{O}(mW^2)$ *operations.*

*Proof.* Since $G$ is a directed acyclic graph, we can order its nodes such that $(i, j) \in E \Rightarrow i \leq j$. Hence, recursion (14) can be solved in $\mathcal{O}(W^2 m)$ operations by computing the value function $T_v^I(\omega)$ according to the order of the nodes. □

The DP recursion (14) cannot be used to solve the problem in general graphs. Specifically, the presence of cycles in $G$ together with the fact that $\omega$ belongs to the inner minimization domain yields cycles in the graph that represents the states of the DP. Stated otherwise, one cannot define an order to fill the states of the value-function $T_v^I(\omega)$. We can overcome this issue by considering a label-setting algorithm, which allows for cycles in the solution.

**Proposition 5.** *In general graphs,* $\mathrm{opt}^I$ *can be computed in* $\mathcal{O}(mW^2 + nW\log(nW))$ *operations.*

*Proof.* To prove the result, we solve WTDSP$^I$ with Algorithm 1 using a Fibonacci heap [12]. The validity of the algorithm can be proved similarly to Dijkstra's algorithm, proving by induction on the cardinality of $S$ that $\theta_v(\omega) = T_v(\omega)$ for each $(v, \omega) \in S$.

□

We finish this section by mentioning that the above algorithms can be extended to provide a pseudo-FPTAS for opt. Let us introduce the notation $\mathbb{Z}/q = \{\frac{z}{q} : z \in \mathbb{Z}\}$. We can refine (14) for the granularity $\mathbb{Z}/q$ as follows.

$$T_v^q(\omega) = \begin{cases} \min\limits_{(u,v) \in E} \left\{ \min\limits_{w \in [(\omega - W_u)^+, \omega] \cap \mathbb{Z}/q} \left\{ T_u^q(w) + C_{uv}(T_u^q(w) + \omega) \right\} \right\} & v \neq o \\ 0 & v = o,\ \omega = 0 \\ +\infty & v = o,\ \omega \neq 0 \end{cases} \tag{15}$$

We can similarly adapt Algorithm 1 to the granularity $\mathbb{Z}/q$. Next, we denote the optimal solution of the problem solved by (15) as $\mathrm{opt}^q = \min_{\omega \in [0,W] \cap \mathbb{Z}/q} T_d^q(\omega)$. One readily verifies that computing $\mathrm{opt}^q$ through (15) can be done in $\mathcal{O}(mq^2W^2)$ operations in acyclic graphs. Similarly, adapting Algorithm 1 to the granularity $\mathbb{Z}/q$ leads to a time complexity of $\mathcal{O}\left(mq^2W^2 + nqW\log(nqW)\right)$ operations. What is more, the next result shows that the approximation of opt provided by $\mathrm{opt}^q$ is tighter than that provided by $\mathrm{opt}^I$.

$\theta_o(0) = 0$
$\theta_v(\omega) = +\infty, v \in V, \omega = 0, \ldots, W, (v,\omega) \neq (o,0)$
$S = \emptyset$ `// set of marked labels`
**repeat**
$\quad (u,\omega) = \underset{(v,w) \in V \times \{0,\ldots,W\} \setminus S}{\arg\min} \{\theta_v(w)\}$ `// mark the label with minimum travel time`
$\quad S \leftarrow S \cup \{(u,\omega)\}$
$\quad$ **for** $(u,v) \in E$ **do**
$\quad\quad$ **for** $\nu = 0, \ldots, \min(W_u, W - \omega)$ **do**
$\quad\quad\quad \theta_v(\omega+\nu) = \min(\theta_v(\omega+\nu), \theta_u(\omega) + C_{uv}(\theta_u(\omega) + \omega + \nu))$ `// update minimum travel times`
**until** $u = d$
**output:** $\theta_v(\omega), v \in V, \omega \in \{0, \ldots, W\}$

**Algorithm 1:** Label-setting algorithm for WTDSP

**Proposition 6.** *Let* $C_{min} = \inf_{e \in E, t \geq 0} C_e(t) \geq 1$*. Then,* $\text{opt}^q \leq \left(1 + \frac{1}{q \times C_{min}}\right) \text{opt}$*.*

*Proof.* Let $(p^*, w^*)$ be the optimal solution to WTDSP and $(p^*, \lfloor q \times w^* \rfloor / q)$ a feasible solution to its $\mathbb{Z}/q$ counterpart. We obtain

$$\text{opt}^q \leq T(p^*, \lfloor q \times w^* \rfloor / q) \leq T(p^*, w^*) + \frac{|p^*|}{q} \leq \text{opt} + \frac{1}{q \times C_{min}} \text{opt} = \left(1 + \frac{1}{q \times C_{min}}\right) \text{opt}\,.$$

□

Considering the worst-case of Proposition 6 ($C_{min} = 1$), we obtain a pseudo-fully polynomial-time approximation scheme for WTDSP.

# 7 A pseudo-FPTAS for bounded waiting times

Next we study approximation algorithms for WTDSP. Our first observation is that the problem may be hard to approximate in general because optimal solutions could contain pseudo-polynomially many arcs. The example below shows that solutions with polynomially many arcs cannot yield constant-factor approximations for the problem.

**Example 2.** *Let* $\gamma(n)$ *be a given polynomial function. Consider a set of instances built on the graph from Figure 1, with* $W \geq \gamma(n)$*. Then, for any integers* $K$ *and* $L$ *large enough, we define a specific instance as follows:* $W_o = 0$*,* $W_1 = K$*,* $W = L \times K$*,* $C_{o1}(t) = C_{11}(t) = 1$*, and* $C_{1d}(t) = \max(L \times (K+1) + 2 - t, 1)$*. The optimal solution to this instance contains* $L$ *loops around node 1 and its total travel time is equal to* $L + 2$*. The best path,* $p$*, from* $o$ *to* $d$ *with length at most* $\gamma(n)$ *is that having* $\gamma(n)$ *loops around node 1. What is more, the total travel time of* $p$ *is* $L + 2 + K(L - \gamma(n))$*. Hence, for any* $K$*, we can choose* $L$ *large enough so that the approximation ratio provided by* $p$ *is greater than* $1 + K$*.*

The difficulty of getting good approximate solutions in Example 2 is due to the large ratio between $W$ and the minimum travel time $C_{min}$, which we denote by $\alpha$ in what follows. Notice that assuming that $\alpha$ is constant still leads to hard optimization problems as the instances used in Proposition 1 have $\alpha = 1$. The following example shows that optimal solutions can be approximated by the factor $1 + \epsilon$ by paths having polynomially many arcs in $n, m$, $\alpha$, and $1/\epsilon$.

$L = \frac{\epsilon W}{\alpha}$;
$\widetilde{W} \leftarrow \lfloor \frac{W}{L} \rfloor$;
$\widetilde{W}_v \leftarrow \lfloor \frac{W_v}{L} \rfloor$ for all $v \in V$;
$\widetilde{C}_e(t) \leftarrow L^{-1} \times C_e(Lt)$ for all $e \in E$;
Run Algorithm 1 using $\widetilde{W}, \{\widetilde{W}_v\}_{v\in V}$ and $\{\widetilde{C}_e\}_{e\in E}$ instead of $W, \{W_v\}_{v\in V}$ and $\{C_e\}_{e\in E}$;
Let $(\widetilde{p}, \widetilde{w})$ be the optimal solution and $\widetilde{\text{opt}}^I$ the optimal value;
**return** $(\widetilde{p}, L\widetilde{w})$

**Algorithm 2:** Pseudo-FPTAS for WTDSP

**Example 3.** *Consider another variant of the example from Figure 1 described by the integer $K$ with $W_o = 0$, $W_1 = 1$, $W = K$, $C_{o1}(t) = C_{11}(t) = K$ and $C_{1d}(t) = \max(K(K+3) - t, K)$. Hence, $\alpha = 1$. The optimal solution contains $K$ loops around node 1 and its total travel time is equal to $K(K+2)$. Yet, for any $\epsilon > 0$, this modified example can be approximated by the factor $1+\epsilon$ by the path from $o$ to $d$ that contains $\min(K, \lfloor 1/\epsilon \rfloor)$ loops around 1.*

We now provide a pseudo-FPTAS, described in Algorithm 2. The latter follows the classical technique introduced by Ibarra and Kim [16], where the parameters of the instance are rounded and the rounded problem is solved exactly (using Algorithm 1 in our case). In particular, Algorithm 2 involves rounded travel time functions $\widetilde{C}_e(t) = L^{-1} \times C_e(Lt), \forall e \in E$, and waiting bounds $\widetilde{W} = \lfloor \frac{W}{L} \rfloor$ and $\widetilde{W}_v = \lfloor \frac{W_v}{L} \rfloor, \forall v \in V$. For a given path $p = (v_1(= o), v_2, \ldots, v_{|p|}(= d)) \in \mathcal{P}_d$, we get the following set of feasible waiting times.

$$\widetilde{\mathcal{W}}(p, \widetilde{W}) = \left\{ \widetilde{w} \in \mathbb{R}_+^{|p|} : \sum_{k=1}^{|p|} \widetilde{w}_k \le \widetilde{W},\ \widetilde{w}_k \le \widetilde{W}_{v_k}, k = 1, \ldots, |p| \right\}.$$

Similarly to (1), we then define

$$\widetilde{T}(p, \widetilde{w}) = \sum_{k=1}^{|p|-1} \widetilde{C}_k(\widetilde{t}_k),$$

where $\widetilde{w} \in \widetilde{\mathcal{W}}(p, \widetilde{W})$ and $\widetilde{t}_k$ is the departure time from node $v_k$ computed recursively as

$$\widetilde{t}_k = \widetilde{t}_{k-1} + \widetilde{C}_{k-1}(\widetilde{t}_{k-1}) + \widetilde{w}_k. \tag{16}$$

The corresponding optimal travel time is denoted as $\widetilde{\text{opt}}$. We see that $\widetilde{T}(p, \widetilde{w})$ can be easily related to $T(p, L\widetilde{w})$.

**Lemma 4.** *For any path $p \in \mathcal{P}_d$ and $\widetilde{w} \in \widetilde{\mathcal{W}}(p, \widetilde{W})$, we have $\widetilde{T}(p, \widetilde{w}) = L^{-1} \times T(p, L\widetilde{w})$.*

*Proof.* Let $t$ be the solution to recursion (2) for $w = L \times \widetilde{w} \in \mathcal{W}(p, W)$, and $\widetilde{t}$ be the solution to recursion (16) for $\widetilde{w}$. We see by recurrence that $\widetilde{t}_k = L^{-1} t_k$. Specifically, $\widetilde{t}_1 = \widetilde{w}_1 = L^{-1} w_1 = L^{-1} t_1$, and

$$\widetilde{t}_k = \widetilde{t}_{k-1} + \widetilde{C}_{k-1}(\widetilde{t}_{k-1}) + \widetilde{w}^k = L^{-1} t_{k-1} + L^{-1} C_{k-1}(L \times L^{-1} \times t_{k-1}) + L^{-1} w_k = L^{-1} t_k.$$

Hence,

$$\widetilde{T}(p, \widetilde{w}) = \sum_{k=1}^{|p|-1} L^{-1} \times C_k(L \times L^{-1} \times t_k) = L^{-1} T(p, w).$$

□

To compute the approximation ratio of Algorithm 2, we introduce the following notations. We define $\mathbf{W} = (W, W_v : v \in V)$ as the $(n+1)$-tuple of all bounds on waiting times, and define similarly $\widetilde{\mathbf{W}} = (\widetilde{W}, \widetilde{W}_v : v \in V)$. Further, we let $\mathrm{opt}(\mathbf{W})$ be the optimal solution of WTDSP for the specific waiting times $\mathbf{W}$; we define similarly $\mathrm{opt}^I(\mathbf{W})$. The following result details how far $\mathrm{opt}^I(\lfloor \mathbf{W} \rfloor)$ is from $\mathrm{opt}(\mathbf{W})$. We skip the proof, because it is a straightforward adaptation of that of Proposition 2.

**Lemma 5.** *For any $\mathbf{W}' \in \mathbb{R}^{n+1}$, we have $\mathrm{opt}^I(\lfloor \mathbf{W}' \rfloor) \leq (1 + \frac{1}{C_{min}}) \,\mathrm{opt}(\mathbf{W}')$.*

**Proposition 7.** *Let $\alpha = \frac{W}{C_{min}}$. Algorithm 2 is a pseudo-FPTAS for* opt *running in $\mathcal{O}\left(\frac{m\alpha^2}{\epsilon^2} + \frac{n\alpha}{\epsilon} \log\left(\frac{n\alpha}{\epsilon}\right)\right)$ operations.*

*Proof.* Let $(p^*, w^*)$ be the optimal solution of WTDSP and $(\widetilde{p}, L\widetilde{w})$ be the solution returned by Algorithm 2. Notice that the definitions of $\widetilde{C}_e$ and $\alpha$, imply that

$$\inf_{e \in E, t \geq 0} \widetilde{C}_e = \frac{\inf\limits_{e \in E, t \geq 0} C_e(t)}{L} = \frac{C_{min}}{W} \frac{\alpha}{\epsilon} \geq \frac{1}{\epsilon}.$$

Therefore,

$$\begin{aligned}
T(\widetilde{p}, L\widetilde{w}) &= L \times \widetilde{T}(\widetilde{p}, \widetilde{w}) && (\textit{Lemma 4}) \\
&= L \times \widetilde{\mathrm{opt}}^I(\widetilde{\mathbf{W}}) && (\textit{Definition of } \widetilde{\mathrm{opt}}^I) \\
&\leq L(1+\epsilon)\widetilde{\mathrm{opt}}\left(\frac{\mathbf{W}}{L}\right) && (\textit{Lemma 5}) \\
&\leq L(1+\epsilon)\widetilde{T}(p^*, w^*/L) && (\textit{Definition of } \widetilde{\mathrm{opt}}) \\
&= (1+\epsilon)T(p^*, w^*) && (\textit{Lemma 4}) \\
&= (1+\epsilon)\,\mathrm{opt} && (\textit{Definition of } \mathrm{opt})
\end{aligned}$$

What is more, observing that $\widetilde{W} = \lfloor \alpha/\epsilon \rfloor$, we get that Algorithm 2 runs in

$$\mathcal{O}\left(\frac{m\alpha^2}{\epsilon^2} + \frac{n\alpha}{\epsilon} \log\left(\frac{n\alpha}{\epsilon}\right)\right)$$

operations. □

We conclude the section by discussing how to extend the pseudo-FPTAS to WTDSP$^I$. If $L \leq 1$, then $W < \alpha/\epsilon$ and the exact pseudo-polynomial-time algorithms run in polynomial time. Otherwise, we consider $L' = \lfloor L \rfloor$ in Algorithm 2, in which case the algorithm returns solution $(\widetilde{p}, L'\widetilde{w})$. The solution is feasible for WTDSP$^I$ and satisfies $T(\widetilde{p}, L'\widetilde{w}) \leq (1+\epsilon)\,\mathrm{opt} \leq (1+\epsilon)\,\mathrm{opt}^I$.

# 8 Harder variants of WTDSP

Out of completeness, we consider the case the time-dependent network is not consistent (i.e., where Assumption 2 is not satisfied). This can be meaningful in applications where different modes of transport can be taken depending on departure time. It has been considered for instance by Orda and Rom [19] in the context of telecommunications, where a message can be sent through different channels whose availabilities depend on time.

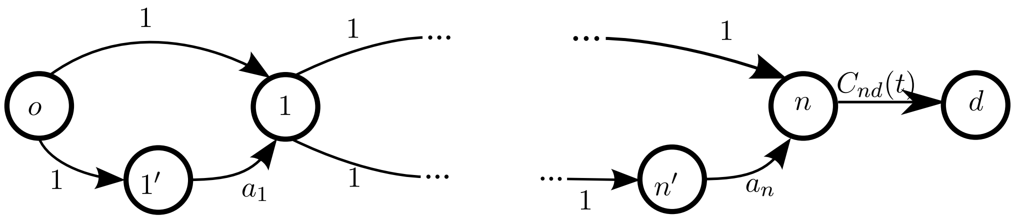


Figure 4: Reduction from partition to show inapproximability.

Orda and Rom [19] state that the problem is $\mathcal{NP}$-Hard when no waiting time is allowed ($W = 0$). As in the rest of the literature, the authors aimed at minimizing arrival time, but without wait, this is equivalent to minimizing arrival time. A proof of $\mathcal{NP}$-Hardness is given by Sherali et al. in [21], where the reduction is made from PARTITION. The reduction is similar to that depicted in Figure 2, but they use a piecewise constant travel time function on the last arc. Below, we extend their result to the class of functions considered in this article, and we study the approximability of the problem.

**Proposition 8.** *If we relax Assumption 2 in the definition of* WTDSP*, then, for any polynomial-time computable function $\gamma(n)$,* WTDSP *cannot be approximated within a factor of $\gamma(n)$ if $\mathcal{P} \neq \mathcal{NP}$, even when $W = 0$.*

*Proof.* We reduce PARTITION to WTDSP to show the inapproximability result. For this, consider an instance of PARTITION characterized by $n$ integers $a_1, \ldots, a_n$ such that $\sum_{i=1}^{n} a_i = 2A$, and consider the corresponding instance depicted on Figure 4, where the travel time $C_e$ of each arc different from $(n, d)$ is a constant function indicated on the figure. In addition, we define $C_{nd}(t) = (n + A + 1)\gamma(n)|t - n - A| + 1$. If the answer to the partition problem is *yes*, we obtain a path of length $n + A + 1$; if it is *no*, there exists no path shorter than $(n + A + 1)\gamma(n)$. $\square$

Another variant of WTDSP could be considered with more practical significance than relaxing Assumption 2. For all $v \in V$, the upper bound $W_v$ means that every time a path goes through $v$, it is allowed to wait at most $W_v$. This means that in a cyclic path going $k$ times through $v$, it is allowed to wait $k \times W_v$ at $v$. This will not be relevant in some contexts, which is why we wish to consider the variant where it is allowed to wait only $W_v$ in total at each node $v$. This comes down to replacing the constraint $w_k \leq W_k$ from the definition of $\mathcal{W}(p, \omega)$ with

$$\sum_{k \in \{1,\ldots,|p|\}: v_k = v} w_k \leq W_v, \; \forall v \in V. \tag{17}$$

The optimization problem we obtain by adding the above constraint to WTDSP is denoted as TWTDSP.

**Proposition 9.** *The optimization problem* TWTDSP *is $\mathcal{NP}$-Hard, and, for any polynomial-time computable function $\gamma(n)$, it cannot be approximated within a factor of $\gamma(n)$ if $\mathcal{P} \neq \mathcal{NP}$.*

*Proof.* We prove the result by reduction from the classical HAMILTONIAN CYCLE problem [13].

We consider an instance of HAMILTONIAN CYCLE described by a directed graph $G = (V, E)$ and let $\gamma(n)$ be any polynomial-time computable function. We also define $\beta(n) = (2n + 3)\gamma(n)$.

As illustrated by Figure 5, we create an instance of TWTDSP by duplicating each node $i$ of $G$ into one *in* and one *out* node, $i^{\text{in}}$ and $i^{\text{out}}$, so that one arc $(i^{\text{out}}, j^{\text{in}})$ with constant travel time is

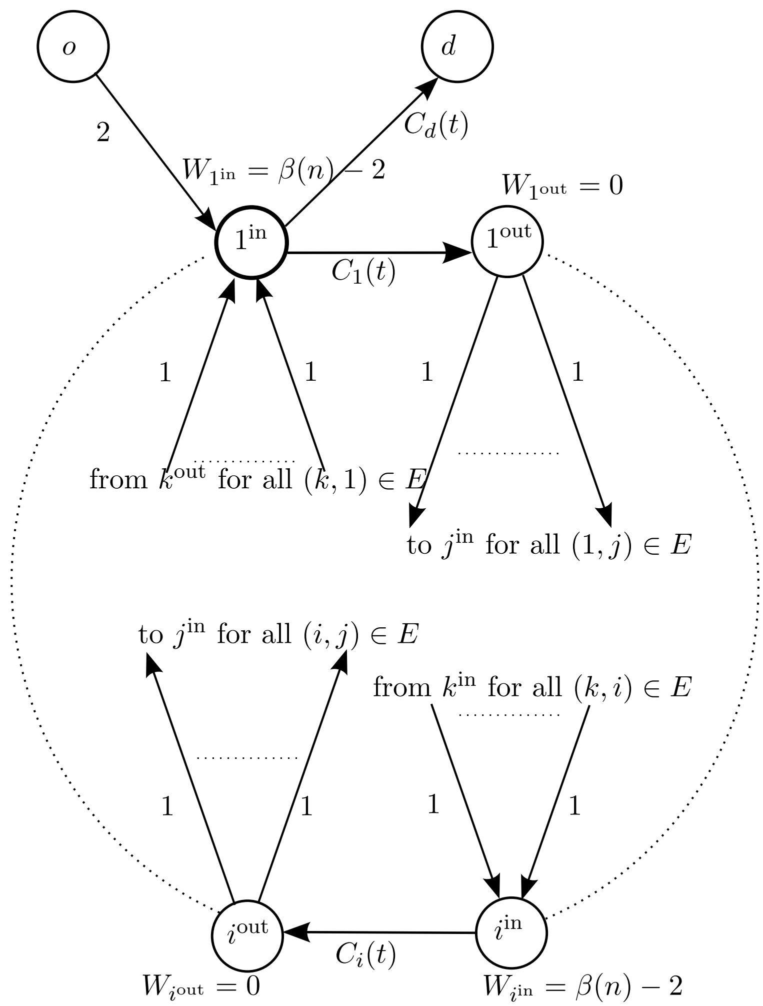


Figure 5: Reduction from HAMILTONIAN CYCLE to an instance of TWTDSP.

created for each arc $(i, j)$ of $E$. We also create one arc $(i^{\text{in}}, i^{\text{out}})$ for each $i \in V$ and define its travel time function as:

$$C_i(t) = \begin{cases} (\beta(n) - 1)(t - k\beta(n)) + 1 & \text{if } k\beta(n) \leq t \leq k\beta(n) + 1, k = 0, \dots, n-1 \\ -t + (k+1)\beta(n) + 1 & \text{if } k\beta(n) + 1 \leq t \leq (k+1)\beta(n), k = 0, \dots, n-1 \\ (\beta(n) - 1)(t - n\beta(n)) + 1 & \text{if } t \geq n\beta(n) \end{cases}$$

The origin node $o$ is connected to $1^{\text{in}}$ with constant travel time 2 and $1^{\text{in}}$ is connected to $d$ with travel time function

$$C_d(t) = \max\{-t + n\beta(n) + 3, \gamma(n)(t - n\beta(n) - 2) + 1\}.$$

Finally, the maximum wait is equal to $\beta(n) - 2$ at nodes $i^{\text{in}}$ and 0 on the other nodes.

First, observe that if the departure time from a node $i^{\text{in}}$ is such that $k\beta(n)+1 \leq t_{i^{\text{in}}} \leq (k+1)\beta(n)$ for some $k \in \{0, \dots, n-1\}$, the expression of $C_i$ implies that $t_{i^{\text{in}}} + C_i(t_{i^{\text{in}}}) = (k+1)\beta(n) + 1$. Since the arcs originating from the *out* nodes all have constant costs equal to 1, this involves that for $k \leq n-1$, the departure time from the $k^{\text{th}}$ *out* node on a path is at least $k\beta(n)+1$. In particular, if $v_{n+1}^{\text{in}}$ is the $(n+1)^{\text{th}}$ *in* node on a path, $t_{v_{n+1}^{\text{in}}} \geq n\beta(n) + 2$. As a consequence, if the path traverses

$(v_{n+1}^{\text{in}}, v_{n+1}^{\text{out}})$, the last piece of the arc travel time function involves that

$$C_{v_{n+1}} \geq \beta(n). \tag{18}$$

Second, let $(p, w)$ be a feasible solution, and denote as $t(d-1)$ the departure time from $1^{\text{in}}$ before traversing $(1^{\text{in}}, d)$. The expression of $C_d$ gives

$$\begin{aligned} T(p,w) &= t(d-1) + C_d(t(d-1)) - \sum_{i=1}^{|p|-1} w_i \\ &= -\sum_{i=1}^{|p|-1} w_i + \begin{cases} n\beta(n) + 3 & \text{if } t(d-1) \leq n\beta(n) + 2 \\ \gamma(n)(2t(d-1) - n\beta(n) - 2) + 1 & \text{if } t(d-1) > n\beta(n) + 2 \end{cases} \end{aligned}$$

From the maximum wait values and constraint (17), we also know that $\sum_{i=1}^{|p|-1} w_i \leq k(\beta(n) - 2)$, where $k$ is the number of distinct *in* nodes in the path. In particular, this means that

$$T(p,w) \geq (2k+3) + (n-k)\beta(n). \tag{19}$$

Now, if the answer to HAMILTONIAN CYCLE is *yes*, there is a path $(v_1, \ldots, v_n, v_{n+1})$ in $G$ where $v_1 = v_{n+1} = 1$ and $v_1, \ldots, v_n$ are distinct nodes of $G$. Thus, we can build a solution $(p, w)$ of TWTDSP where $p = (o, v_1^{\text{in}}, v_1^{\text{out}}, \ldots, v_n^{\text{in}}, v_n^{\text{out}}, 1^{\text{in}}, d)$ and $\sum_{i=1}^{|p|-1} w_i = n(\beta(n) - 2)$ (by waiting $\beta(n) - 2$ at nodes $v_i^{\text{in}}$ for $i = 1, \ldots, n$). With this solution, we get $T(p, w) = (2n + 3)$.

Finally, assume that the answer is *no*. We first observe that there is a straightforward bijection between the cycles starting from node 1 in $G$ and the set of paths that appear in the solutions of TWTDSP. This means that for any solution $(p, w)$ of TWTDSP, either $p$ goes through less than $n$ distinct *in* nodes or it goes through at least $n + 1$ *out* nodes. In the former case, inequality (19) shows that $T(p, w) \geq (2k + 3) + \beta(n) > \gamma(n)(2n + 3)$. In the latter case, inequality (18) applies on the $(n + 1)^{\text{th}}$ arc connecting an *in* node to an *out* node in $p$, which also proves that $T(p, w) > \gamma(n)(2n + 3)$. □

Observe that although constraint (17) guarantees a maximum total waiting time at each node, it does not exclude the possibility of a cyclic optimal solution. To see this, consider the simple example drawn on Figure 6. The costs of the arcs other than $(2, d)$ are all constant and equal to 1, $C_{2d}(t) = \max\{7 - t, 1\}$, and the bounds are $W_v = 0, \forall v \neq 3$, and $W_3 = 1$. It is straightforward to verify that the optimal value is 6 by considering the solution defined as $p = (o, 1, 2, 3, 1, 2, d)$ and $w_3 = 1$. The only acyclic path from $o$ to $d$ is $(o, 1, 2, d)$ whose total travel time is equal to 7. It is also interesting to notice that adding constraint (17) does not change the optimal value of this instance.

# References


[1] G. Borradaile and M. Shirley, *Time-dependent shortest paths in bounded treewidth graphs*, (2017), arxiv:1706.01508v1.

[2] X. Cai, T. Kloks, and C.K. Wong, *Time-varying shortest path problems with constraints*, Networks **29** (1997), 141–150.

[3] I. Chabini and B. Dean, *Shortest path problems in discrete-time dynamic networks: Complexity, algorithms and implementations*, Unpublished manuscript (1999).

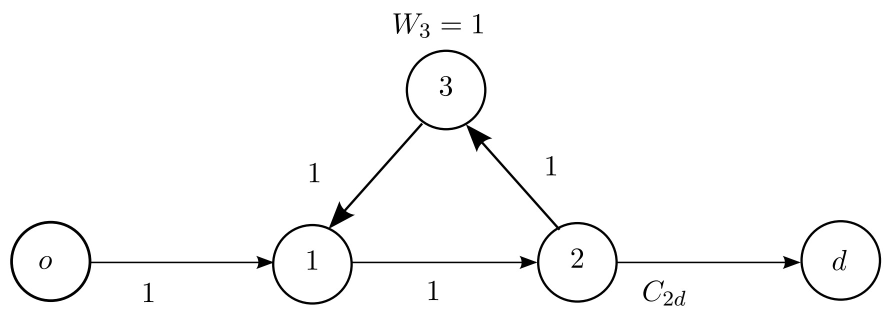


Figure 6: Instance of WTDSP with a cyclic solution.

[4] K.L. Cooke and E. Halsey, *The shortest route through a network with time-dependent internodal transit times*, J. Math. Anal. Appl. **14** (1966), 493 – 498.

[5] B.C. Dean., Shortest paths in FIFO time-dependent networks: Theory and algorithms. Computer Science Technical Report, MIT, 2004.

[6] B.C. Dean, *Algorithms for minimum-cost paths in time-dependent networks with waiting policies*, Networks **44** (2004), 41–46.

[7] F. Dehne, M.T. Omran, and J.R. Sack, *Shortest paths in time-dependent FIFO networks*, Algorithmica **62** (2012), 416–435.

[8] D. Delling and D. Wagner, "*Time-dependent route planning*," *Robust and Online Large-Scale Optimization: Models and Techniques for Transportation Systems*, R.K. Ahuja, R.H. Möhring, and C.D. Zaroliagis (eds.), Springer Berlin Heidelberg, 2009, pp. 207–230.

[9] B. Ding, J.X. Yu, and L. Qin, *Finding time-dependent shortest paths over large graphs*, Proceedings of the 11th International Conference on Extending Database Technology: Advances in Database Technology, 2008, EDBT '08, pp. 205–216.

[10] S.E. Dreyfus, *An appraisal of some shortest-path algorithms*, Oper. Res. **17** (1969), 395–412.

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

[12] M.L. Fredman and R.E. Tarjan, *Fibonacci heaps and their uses in improved network optimization algorithms*, J ACM **34** (1987), 596–615.

[13] M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman & Co., New York, NY, USA, 1979.

[14] M. Gendreau, G. Ghiani, and E. Guerriero, *Time-dependent routing problems: A review*, Comput. Oper. Res. **64** (2015), 189–197.

[15] J. Halpern, *Shortest route with time dependent length of edges and limited delay possibilities in nodes*, Zeitschrift für operations research **21** (1977), 117–124.

[16] O.H. Ibarra and C.E. Kim, *Fast approximation algorithms for the knapsack and sum of subset problems*, J ACM **22** (1975), 463–468.

[17] E. Kanoulas, Y. Du, T. Xia, and D. Zhang, *Finding fastest paths on a road network with speed patterns*, 22nd International Conference on Data Engineering (ICDE'06), 2006, pp. 1–10.

[18] D.E. Kaufman and R.L. Smith, *Fastest paths in time-dependent networks for intelligent vehicle-highway systems application*, I V H S J. **1** (1993), 1–11.

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

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

[21] H.D. Sherali, K. Ozbay, and S. Subramanian, *The time-dependent shortest pair of disjoint paths problem: Complexity, models, and algorithms*, Networks **31** (1998), 259–272.