Time-Dependent Shortest Path Problems with Penalties and Limits on Waiting

Edward He*†1, Natashia Boland ‡1, George Nemhauser §1, and Martin Savelsbergh ¶1

[1]H. Milton Stewart School of Industrial and Systems Engineering, Georgia Institute of Technology, U.S.

April 11, 2020

Abstract

Waiting at the right location at the right time can be critically important in certain variants of time-dependent shortest path problems. We investigate the computational complexity of time-dependent shortest path problems in which there is either a penalty on waiting or a limit on the total time spent waiting at a given subset of the nodes. We show that some cases are NP-hard, and others can be solved in polynomial time, depending on the choice of the subset of nodes, on whether waiting is penalized or is constrained, and on the magnitude of the penalty/waiting limit parameter.

Keywords: shortest path, time-dependent travel time, time-expanded network, computational complexity

1 Introduction

Time-dependent shortest path problems generalize the classical problem of finding a path in a network from a given origin node to a given destination node so as to minimize a function of the arcs used, by introducing the element of time. A time-dependent path is a path in time and space (the network) that departs the origin at a time no earlier than the start of a given time horizon and reaches the destination no later than the end of the horizon, where the time to traverse an arc in the network is a function of the time of departure at its tail node. Such problems are especially of interest in road networks, where traffic congestion conditions affect the time needed to traverse links in the network (Letchner et al. 2006, Franceschetti et al. 2018).

Applications in transport logistics are emerging, in part because the now ubiquitous GPS-enabled devices supply the data needed to reliably estimate arc travel time functions in road networks (Bertsimas et al. 2019), complementing the more traditional reliance on data from inductive loops (Wang and Nihan 2000, Coifman 2002).

*Corresponding author.
†edwardhe@gatech.edu
‡natashia.boland@isye.gatech.edu
§gn3@gatech.edu
¶martin.savelsbergh@isye.gatech.edu

A number of variants of the time-dependent shortest path problem have been studied, starting with the *minimum arrival time* variant in Cooke and Halsey (1966), Orda and Rom (1990), and Dean (2004b), the *minimum duration* variant in Orda and Rom (1990), Chabini (1998), Nachtigall (1995), Kanoulas et al. (2006), Ding et al. (2008), Foschini et al. (2014) and He et al. (2019), and, recently, the *minimum travel time* variant in He et al. (2019). These variants differ not only in their choice of objective function, but also in the nodes at which waiting may occur in an optimal solution. In the Minimum Travel Time Problem (MTTP), an optimal solution may wait at any node, including the origin, to take advantage of shorter travel times that occur later in the time horizon. The MTTP objective is to minimize the sum of the travel times along the arcs used; provided the path starts and ends within the given time horizon, time spent waiting is ignored. In the Minimum Arrival Time Problem (MATP), a path reaching the destination as soon as possible is sought. Under the commonly assumed *first-in-first-out* (FIFO) property of arc travel time functions, which states that waiting at the tail node of an arc can never result in an earlier arrival at its head node, waiting before departing a node is suboptimal for the MATP. In the Minimum Duration Problem (MDP) under FIFO, a path minimizing the difference between the arrival time at the destination and the departure time at the origin is sought. Thus, waiting to depart at any node other than the origin is suboptimal for the MDP. It has been shown that for piecewise linear travel time functions satisfying FIFO, all three variants can be solved in polynomial time (see Cooke and Halsey (1966), Foschini et al. (2014), and He et al. (2019)). We note that early algorithms on variants in which waiting can be beneficial, e.g., the ones in Chabini (1998) and Dean (2004a), rely on time discretization, and, thus, provide only approximate solutions, whereas recent algorithms, e.g., the ones in Foschini et al. (2014) and He et al. (2019), however, properly handle continuous time and provide optimal solutions.

That the study of different objective functions is not only of academic interest can be seen in Cooper and Cowlagi (2018). They consider the problem of planning a route in which the objective is to minimize a "weighted sum of travel duration and exposure to traffic", and argue that such an objective may be of importance to reduce the health risks for long-haul truck drivers by reducing their exposure to emissions. They observe that because real-time traffic data is becoming widely available, which allows accurate predictions of traffic, and, thus, travel times, an optimal route may involve waiting for traffic to subside. They also comment that this type of objective may be relevant for motion-planning of aerial vehicles in inclement weather.

In this paper, we consider additional variants in which there is either a penalty incurred for waiting, or there is a limit on the total time spent waiting, at a given subset of the nodes. We determine the complexity of these additional variants. For variants that are NP-hard, we provide a complexity proof and for variants that are polynomially solvable, we provide an algorithm. We will not discuss the variant in which there is a constraint on waiting at each individual node, which has been shown to be NP-hard, by Omer and Poss (2019b).

The remainder of the paper is organized as follows. In Section 2, we formally introduce the basic variants of the time-dependent shortest path problem as well as the variants with penalties and constraints on waiting, which are the focus of our research. In Section 3, we discuss a few natural extensions of the basic variants. In Section 4. we present NP-completeness proofs for the (remaining) variants with penalties and constraints on waiting that are hard. In Sections 5 and 6. we present polynomial time algorithms for the (remaining) variants with penalties and constraints on waiting that are easy. Finally, in Section 7, we summarize the complexity status of all variants with penalties and limits on waiting.

2 Problem Description and Preliminaries

2.1 Problem Setting Description and Illustration

Given is a directed network $D = (N, A)$, with nodes $N = \{1, \ldots, n\}$, arcs $A \subset N \times N$, a time horizon $[0, T]$ and, for each arc in $(i, j) \in A$, a time-dependent arc travel time function $c_{i,j} : [0, T] \to \mathbb{R}_+$. Also given is an origin node, node 1, and a destination node, node n.

We will use the following example to illustrate the problem variants we consider. The example network is given in Figure 1. It has time horizon $[0, 5]$. The piecewise linear travel time functions for each arc is given by its breakpoints in Table 1. For example, for $t \in [1, 2]$, $c_{2,3}(t) = 1.82 + (1.51 - 1.82)(t - 1) = 2.13 - 0.31t$. Arc $(3, 4)$ has no breakpoints between times 2 and 5, indicated by dashes in the table, so for $t \in [2, 5]$, $c_{3,4}(t) = 0.83 + ((1.00 - 0.83)/(5 - 2))(t - 2) = (215 + 17t)/300$. Observe that the slopes of all linear pieces are greater than -1, which implies that the FIFO property is satisfied strictly.

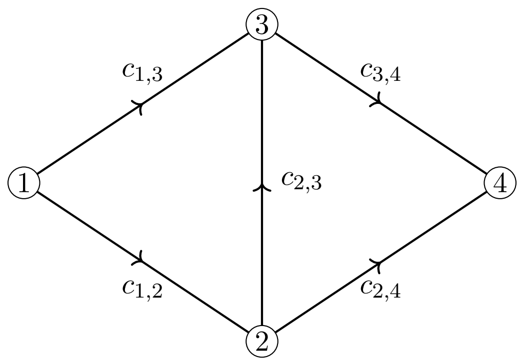


Figure 1: Network D.

Table 1: Arc travel times at each breakpoint (BP).

BP Time	Arc Travel Times				
	(1,2)	(1,3)	(2,3)	(2,4)	(3,4)
0	1.34	2.85	1.99	1.29	0.61
1	0.66	2.95	1.82	1.02	0.73
2	0.14	3.00	1.51	1.63	0.83
3	0.01	2.98	1.10	2.57	—
4	0.35	2.90	0.67	3.00	—
5	1.00	2.76	0.30	2.54	1.00

In this example, the earliest arrival time at node 2 is 1.34, achieved by departing on arc $(1, 2)$ at time $t = 0$. Now the time needed to traverse arc $(2, 3)$ if departing at time $t = 1.34$ is $c_{2,3}(1.34) = 2.13 - 0.31 \times 1.34 = 1.7146$, so travel to node 3 on the path consisting of arcs $(1, 2)$ and $(2, 3)$, without any waiting at either node 1 or node 2, arrives at node 3 at time $1.34 + 1.7146 = 3.0546$. This is later than $c_{1,3}(0) = 2.85$, so the earliest arrival at node 3 is 2.85, achieved by departing on arc $(1, 3)$ at time $t = 0$. Thus the earliest arrival time at node 4 by a path through node 3 must be at time $2.85 + c_{3,4}(2.85) > 2.85$, while the earliest arrival at node 4 via arc $(2, 4)$ must be $1.34 + c_{2,4}(1.34) = 1.34 + 1.2274 = 2.5674 < 2.85$. Thus the solution to the MATP in this example is the path consisting of arcs $(1, 2)$ and $(2, 4)$, departing the origin, node 1, at time 0 and arriving at the destination, node 4, at time 2.5674. This solution is shown in time-expanded form in Figure 2, drawn in black (times shown are rounded to two decimal places).

The same sequence of arcs also yields the MDP solution in this example, if departure at the origin is delayed by 2 units of time. At time $t = 2$, travel time on arc $(1, 2)$ is 0.14, and at time $t = 2.14$, travel time on arc $(2, 4)$ is 1.76, giving an arrival time of $2.14+1.76 = 3.90$, but a duration of only $3.9 - 2 = 0.14 + 1.76 = 1.90$. In this example, this is also the minimum travel time, so the same arc sequence and departure times on each arc solves the MTTP.

2.2 New Problem Variants

The two new variants of time-dependent shortest path problems that we consider both seek a *timed path*, which consists of a path in the network and a specified departure time for each arc in the path, that departs from node 1 at time 0 or later, and arrives at node n no later than time T. If $(i, j), (j, k) \in A$ are arcs used consecutively in a timed path, P, with associated departure times t and t', respectively, then we call $\omega(P, j) := t' - (t + c_{i,j}(t))$ the *waiting time* of P at node j. Naturally, the waiting time at any node in a timed path must be non-negative, and if it is positive, we say the path *waits at* the node. If the timed path departs the origin node, 1, at time t, we say that the waiting time at node 1 is t, while if it arrives at the destination node, n, at time t, we say the waiting time at node n is $T - t$. A timed path, P, has an associated *travel time*, denoted by $\tau(P)$, which is the sum of the traversal times of its arcs at their specified departure times. Thus, the sum of the waiting time at *all* nodes in a timed path, added to its travel time, yields T, the length of the time horizon.

In the illustrative example, the timed path consisting of arc $(1, 2)$ departing at time 2 and arc $(2, 4)$ departing at time 2.14, which solves both the MDP and MTTP in this instance, has travel time 1.90 (which, in this case, is also its duration). It has waiting time of 2 at node 1, zero at node 2 and a waiting time of $(5 - 3.9) = 1.1$ at node 4. Another example is the timed path consisting of arc $(1, 2)$ departing at time 2 and arc $(2, 4)$ departing at time 2.5. Since $c_{2,4}(2.5) = 2.1$, this timed path has travel time $0.14+2.1 = 2.24$. Its waiting time at node 1 is 2, at node 2 is $2.5-2.14 = 0.36$ and at node 4 is $5 - (2.5 + 2.1) = 0.4$.

In both problem variants we consider, we *account for* or *tally* waiting at a given set of nodes $M \subseteq N$. We call M the *tally set* and define the *tallied waiting time* of a timed path P to be

$$\omega(P) := \sum_{j \in M} \omega(P, j).$$

For a node $j \notin M$, we say waiting at j is *free*.

2.2.1 Variant with a penalty on tallied waiting time

The first variant we consider applies a penalty to the tallied waiting time. It seeks a timed path, P, so as to minimize $\tau(P) + \alpha\omega(P)$ for some penalty parameter $\alpha \geq 0$. Here $\alpha < 1$ indicates that waiting is "cheaper" than traveling and $\alpha > 1$ indicates that waiting is "more expensive" than traveling. We refer to this variant as the *Time-Dependent Shortest Path Problem with Penalties for Waiting* (TDSPP-PW). For nodes in $N \setminus M$, waiting is allowed without penalty.

To illustrate, consider the TDSPP-PW for the example with $\alpha = 0.3$ and $M = \{1\}$. The timed path consisting of arc $(1, 2)$ departing at time 2 and arc $(2, 4)$ departing at time 2.14, which solves both the MDP and MTTP in this instance, has TDSPP-PW objective value of $1.9 + \alpha \times 2 = 2.5$, since its waiting time at node 1 is 2. The optimal solution to this TDSPP-PW consists of the same arcs, but departs on $(1, 2)$ at time 1, and on $(2, 4)$ at time 1.66. It gives the optimal TDSPP-PW objective value: $(0.66 + 1.42) + \alpha \times 1 = 2.38$.

Solutions to two other TDSPP-PW problems using the same network and travel time functions are shown in Figure 2, presented in a time-expanded network. In both cases, the tallied waiting penalty, α, exceeds 1 (any value greater than 1 yields the given solution). The solid path solves the TDSPP-PW with waiting tallied at all nodes other than the destination. Note that this is precisely the MATP optimal solution. The dashed path solves the TDSPP-PW with waiting tallied at all nodes other than the origin. This is precisely the "reverse MATP" optimal solution: it is the path that departs the origin as late as possible. As we shall prove shortly, these are not coincidences!

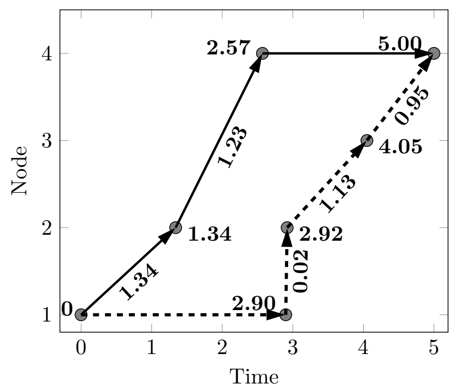


Figure 2: The optimal solution to the TDSPP-PW with $\alpha > 1$: the case of $M = N \setminus \{n\} = \{1, 2, 3\}$ is shown as solid lines and the case of $M = N \setminus \{1\} = \{2, 3, 4\}$ is shown as dashed lines.

2.2.2 Variant with a limit on tallied waiting time

The second variant we consider, rather that having a penalty associated with the tallied waiting time, imposes a limit on it. For $W \geq 0$ a given parameter, the *Time-Dependent Shortest Path Problem with Limited Waiting* (TDSPP-LW) seeks timed path P so as to minimize $\tau(P)$ subject to the constraint $\omega(P) \leq W$. For nodes in $N \setminus M$, waiting is unrestricted by the limit, W.

In the illustrative example, taking tally set $M = N \setminus \{n\} = \{1, 2, 3\}$ and waiting limit $W = 0.5$, the optimal TDSPP-LW solution waits at the origin until time 0.5, then departs on arc $(1, 2)$, arriving at time $0.5 + 1.0 = 1.5$, departs immediately on arc $(2, 4)$, (as it must, since it has already reached the waiting time limit), arriving at time $1.5 + 1.325 = 2.825$. The objective value is the path's total travel time: 2.325.

2.2.3 Recovering prior variants as special cases

These two variants provide natural generalizations of the MATP, MDP and MTTP, each of which can be recovered as a special case of TDSPP-PW and TDSPP-LW for a specific choice of M and of α or W, respectively. The MTTP is the case of TDSPP-PW with $\alpha = 0$: irrespective of the choice of M, the waiting penalty is zero, so waiting is, in effect, ignored. Under the FIFO assumption, MATP is the case of TDSPP-LW with $M = N \setminus \{n\}$ and $W = 0$, since in the MATP, waiting at any node other than the destination is suboptimal: this may be enforced as a constraint, in which case minimizing arrival time is equivalent to minimizing total travel time. The MDP, under the FIFO assumption, is the case of TDSPP-LW with $M = N \setminus \{1, n\}$ and $W = 0$, since waiting at any node other than the origin and destination is suboptimal for the MDP, so may be enforced as a constraint, in which case minimizing duration is equivalent to minimizing total travel time under this constraint.

In the remainder of this paper, we analyze the complexity of the TDSPP-PW and the TDSPP-LW, to determine the effect of the choice of M and of α or W, respectively, on the problem's complexity. We do so under the following assumptions.

2.3 Assumptions

The following assumptions on the underlying data apply throughout the remainder of this paper.

Assumption 1. *The time-dependent arc travel time functions are continuous piecewise linear.*

This is a common assumption in the literature, see, for example, Ichoua et al. (2003) and Figliozzi (2012).

Assumption 2. *The time-dependent travel time functions satisfy the first-in first-out (FIFO) property: for all $(i, j) \in A$ and $t, t' \in [0, T]$ with $t < t'$, $c_{i,j}(t') \geq c_{i,j}(t)$, guaranteeing that a later departure implies a no earlier arrival.*

For continuous piecewise linear functions, this is equivalent to the requirement that the slope of all linear pieces is at least -1. For simplicity of exposition, we require that the FIFO property holds *strictly* (slopes of all linear pieces are strictly greater than -1). Only minor modifications need to be made to the proofs in this paper to account for the non-strict FIFO property, mainly to deal carefully with the case of multiple optimal solutions.

Assumption 3. *For each node $i \in N \setminus \{1, n\}$, there exists a path from node 1 starting at time 0 visiting node i and reaching node n at or before time T.*

If there exists a node i for which Assumption 3 does not hold, it can be safely removed from the network, as it cannot be used in any feasible solution.

3 Further Cases Equivalent to MATP, MDP or MTTP

In Section 2.1, we noted that specific choices of M and of the penalty/waiting limit parameters for TDSPP-PW/TDSPP-LW immediately results in the problem becoming equivalent to an MTTP, MATP or MDP (where the latter two cases used the FIFO assumption). Hence these parameter choices lead to problems that are solvable in polynomial time. Here we provide arguments showing that other choices also give rise to one of these previously-studied variants.

We begin by observing that there is a symmetry between the choice of M with $n \in M \subseteq N \setminus \{1\}$ and $1 \in M \subseteq N \setminus \{n\}$, i.e., for a problem with one of these choices, there is an equivalent problem with the other. To establish this, it is helpful to first formally define a timed path. For later convenience, we include in the notation for a timed path not only the departure time on each arc of the path, but also redundant information: the arrival time at the head node of each arc on the path.

Definition 1. *A* timed path $P = ((i_1, t_1), (i_2, t_2), \ldots, (i_K, t_K))$ *consists of a sequence of K node and time pairs such that, for all $k = 1, \ldots, K-1$, either $i_k = i_{k+1}$ and $t_{k+1} > t_k$ or $(i_k, i_{k+1}) \in A$ and $t_{k+1} = t_k + c_{i_k,i_{k+1}}(t_k)$. Furthermore, this description is minimal in the sense that at most two of the nodes i_{k-1}, i_k and i_{k+1} may be identical, for any $k = 2, \ldots, K-1$.*

We say that P starts at node i_1 and ends at node i_K, starting at time t_1 and ending at time t_K. We may also say that P is a timed path from i_1 to i_K. If $i_k \neq i_{k-1}$ for all $k = 2, \ldots, K$, then we say that P is *waiting-free*. P is *contained in* any time interval $[t^-, t^+]$ with $t^- \leq t_1$ and $t_K \leq t^+$; the interval *contains* P. We use $N(P) = \{i_1, i_2, \ldots, i_K\}$ to denote the set of nodes in P. The travel time and tallied waiting time of P are given by

$$\tau(P) = \sum_{\substack{k=1,\\ i_k \neq i_{k+1}}}^{K-1} c_{i_k,i_{k+1}}(t_k) \qquad \text{and} \qquad \omega(P) = \sum_{\substack{k=1,\\ i_k = i_{k+1} \in M}}^{K-1} (t_{k+1} - t_k).$$

Recall that if $i_k = i_{k+1}$ then $\omega(P, i_k) = t_{k+1} - t_k$. We also define $N^w(P) = \{i \in N(P) : \omega(P, i) > 0\}$ to be the set of nodes at which the path waits. So the tallied waiting time of P may be equivalently written as

$$\omega(P) = \sum_{\substack{k=1,\\ i_k = i_{k+1} \in M}}^{K-1} \omega(P, i_k) = \sum_{i \in N^w(P) \cap M} \omega(P, i).$$

For convenience, we require that timed path $P = ((i_1, t_1), \ldots, (i_K, t_K))$ is feasible for the problems we consider only if $i_1 = 1$, $t_1 = 0$, $i_K = n$ and $t_K = T$. To illustrate using the example, the timed path $P = ((1, 0), (1, 2), (2, 2.14), (2, 2.5), (4, 4.6), (4, 5))$ is feasible for the TDSPP-PW and has $\tau(P) = 0.14 + 2.1 = 2.24$. For $M = \{1, 2\}$, it has $\omega(P) = 2 + 0.36 = 2.36$.

Proposition 1. *Given a TDSPP-PW or TDSPP-LW instance with network (N, A), origin node $o \in N$, destination node $d \in N$, travel time function c_a over time horizon $[0, T]$ for each $a \in A$ and tally set $M \subseteq N$, there exists an equivalent instance of the same problem variant with the same node set, N, and the same tally set, M, but with the origin and destination nodes swapped: it has origin node d and destination node o.*

Proof. Consider a given instance (of either the TDSPP-PW or TDSPP-LW) as described in the proposition. Construct another instance by swapping the roles of 1 and n and of 0 and T, and reversing the direction of the arcs, to create arc set $\overleftarrow{A} := \{(j, i) : (i, j) \in A\}$. Construct the travel time function on each reversed arc, $(j, i) \in \overleftarrow{A}$, by $\overleftarrow{c}_{j,i}(s) := c_{i,j}(t)$ where t solves $T - t - c_{i,j}(t) = s$, for all $s \in [0, T]$. We claim that the instance having network $(N, \overleftarrow{A})$, origin node d, destination node o, travel time function $\overleftarrow{c}_a$ over time horizon $[0, T]$ for each $a \in \overleftarrow{A}$ and tally set $M \subseteq N$, is equivalent to the original instance.

To prove this claim, we consider an arbitrary timed path from o to d for the original instance. Say $P = ((i_1, t_1), (i_2, t_2), \ldots, (i_K, t_K))$ from $i_1 = o$ to $i_K = d$ is a timed path for the original instance. This corresponds to a timed path $\overleftarrow{P} = ((i_K, T - t_K), (i_{K-1}, T - t_{K-1}), \ldots, (i_1, T - t_1))$ from d to o for the new instance. For $i_{k+1} \neq i_k$, so $(i_{k+1}, i_k) \in \overleftarrow{A}$, the travel time $\overleftarrow{c}_{i_{k+1},i_k}(T - t_{k+1}) := c_{i_k,i_{k+1}}(t)$ where t solves $T - t - c_{i_k,i_{k+1}}(t) = T - t_{k+1}$, which is equivalent to $t_{k+1} = t - c_{i_k,i_{k+1}}(t)$. By the definition of P as a timed path, this is solved by $t = t_k$. Thus the two paths have identical travel time. Furthermore, for $i_{k+1} \neq i_k$, we have that $(T - t_k) - (T - t_{k+1}) = t_{k+1} - t_k$, so the two paths also have identical waiting time at every node and hence identical tallied waiting time. The result follows. □

Applying this result to the TDSPP-LW with $M = N \setminus \{1\}$, we obtain the following.

Corollary 1. *The case of TDSPP-LW with $M = N \setminus \{1\}$ and $W = 0$ is equivalent to the MATP.*

Proof. By Proposition 1, any instance of TDSPP-LW with tally set $N \setminus \{1\}$ and $W = 0$ has an equivalent instance of TDSPP-LW with tally set $N \setminus \{n\}$ and $W = 0$. This variant is, as discussed in Section 2.1, equivalent to the MATP. □

That the TDSPP-PW with $M = N$ and $0 < \alpha \leq 1$ can be solved in polynomial time follows from the following straightforward observation that it is equivalent to solving an MTTP.

Proposition 2. *The TDSPP-PW with $M = N$ and $0 < \alpha \leq 1$ is equivalent to the MTTP.*

Proof. The time in the planning horizon must be divided between travel time and waiting time, and since waiting costs no more than traveling, minimizing the combined objective is equivalent to minimizing travel time. More formally, when $M = N$, waiting time is tallied at every node, including the origin and destination, so for any timed path P from 1 to n starting and ending within the horizon $[0, T]$, it must be that $\tau(P) + \omega(P) = T$. Thus the problem of minimizing $\tau(P) + \alpha\omega(P) = \tau(P) + \alpha(T - \tau(P)) = \alpha T + (1 - \alpha)\tau(P)$ is equivalent to minimizing $\tau(P)$ when $1 - \alpha \geq 0$. □

A critical concept used to prove some of the results that follow is that a timed path can be replaced with one that follows the same sequence of arcs and waits for the same amount of time at each node, but departs earlier (or arrives later) while taking "not much more" travel time. This concept is formalized in the following lemma.

Lemma 1. *Let P be a timed path starting at node i at time t_i and ending at node j at time t_j, that does not wait at either i or j. Let $[t^-, t^+]$ be a time interval that contains P. Then for any $\beta \in (0, t_i - t^-]$, the path Q that*

1. *has the same arc (and node) sequence as P,*
2. *starts at node i at time $t_i - \beta \geq t^-$, and*
3. *spends the same amount of time waiting at nodes as P does and so has $\omega(Q, h) = \omega(P, h)$ for all $h \in N(P)$,*

satisfies $\tau(Q) < \tau(P) + \beta$. Similarly, for all $\beta \in (0, t^+ - t_j]$, the path Q that

1. *has the same arc (and node) sequence as P,*
2. *ends at node j at time $t_j + \beta \leq t^+$, and*
3. *has $\omega(Q, h) = \omega(P, h)$ for all $h \in N(P)$,*

satisfies $\tau(Q) < \tau(P) + \beta$.

Proof. Consider some $\beta \in (0, t_i - t^-]$ and define Q to be the timed path that starts at node i at time $t_i - \beta$, follows the same arc sequence as P does, waiting at each node for the same amount of time as P. By inductive application of the (strict) FIFO property, Q ends at node j before P does. Say Q ends at time $t' < t_j$. Thus, since

$$\tau(Q) + \sum_{h \in N(Q)} \omega(Q,h) = t' - (t_i - \beta) \quad \text{and} \quad \tau(P) + \sum_{h \in N(P)} \omega(P,h) = t_j - t_i,$$

it must be that

$$\begin{aligned}\tau(Q) = t' - t_i + \beta - \sum_{h \in N(Q)} \omega(Q,h) = t' - t_i + \beta - \sum_{h \in N(P)} \omega(P,h) &< t_j - t_i - \sum_{h \in N(P)} \omega(P,h) + \beta \\ &= \tau(P) + \beta,\end{aligned}$$

as required.

The case of $\beta \in (0, t^+ - t_j]$ with Q ending at node j at time $t_j + \beta \leq t^+$ is similar; FIFO is applied inductively in the reverse direction, backward along the path from j. □

The proof of Lemma 1 employs the observation that (strict) FIFO extends to timed paths that wait the same amount of time at each node, by inductive application of the (strict) FIFO property. Since this observation is useful in later proofs, we state it formally.

Observation 1. *If two timed paths P_1 and P_2 have the same node sequence and wait at the same nodes for the same amount of time, but P_1 departs earlier than P_2, then the timed copy of each node in the sequence for P_1 is earlier than the timed copy for that same node in P_2.*

We now give results for the TDSPP-PW with $\alpha > 1$.

Lemma 2. *For the TDSPP-PW, when $\alpha > 1$ it is suboptimal to wait at nodes in the tally set if either the origin node or the destination node is not in the tally set.*

Proof. Suppose, for contradiction, that P is an optimal timed path that waits at a node in the tally set, M. Let $i \in M$ be such a node, and say (i,t) and (i,t') are two consecutive elements in P with $t' > t$. Split P into two timed paths: P_1 is the part of P starting from node 1 and ending with (i,t) and P_2 is the part of P starting with (i,t') and ending at node n. Consider the case that $1 \notin M$. Then we may, without loss of generality, assume that P, and hence P_1, has no waiting at node 1. Note P_1 is contained in the interval $[0, t']$. Applying the second part of Lemma 1 to P_1 with $\beta = t' - t = \omega(P,i)$, there exists a timed path Q from node 1 to node i that arrives at time t' with $\omega(Q) = \omega(P_1)$ and $\tau(Q) < \tau(P_1) + \beta$. By concatenating Q and P_2, noting that the resulting timed path has no waiting at node $i \in M$, we obtain a timed path with tallied waiting time equal to that of P minus $\omega(P,i)$. Thus, by writing $Q \cup P_2$ to denote the concatenation of Q and P_2, we have that

$$\begin{aligned}\tau(Q \cup P_2) + \alpha\omega(Q \cup P_2) &= \tau(Q) + \tau(P_2) + \alpha(\omega(P) - \omega(P,i)) \\ &< \tau(P_1) + \beta + \tau(P_2) + \alpha(\omega(P) - \beta) = \tau(P) + \alpha\omega(P) - (\alpha - 1)\beta,\end{aligned}$$

which contradicts the optimality of P, since $\alpha - 1 > 0$. The alternative case, of $n \notin M$, can be shown similarly to result in a contradiction. In this case, the second part of Lemma 1 is applied to P_2, which is contained in the interval $[t, T]$, again applied with $\beta = t' - t > 0$. □

Proposition 3. *The case of TDSPP-PW where $M = N \setminus \{1, n\}$ and $\alpha > 1$ is equivalent to MDP.*

Proof. By Lemma 2, any optimal solution to the TDSPP-PW in this case cannot wait at any node in M. Therefore, this TDSPP-PW is equivalent to minimizing travel time subject to no waiting at any node except the origin or destination, which is exactly the MDP (under FIFO). □

Proposition 4. *The case of TDSPP-PW where $M = N \setminus \{n\}$ or $M = N \setminus \{1\}$, and $\alpha > 1$, is equivalent to MATP.*

Proof. Suppose $\alpha > 1$. By Lemma 2, any optimal solution to the TDSPP-PW cannot wait at any node in M. Therefore, the TDSPP-PW with $M = N \setminus \{n\}$ is equivalent to minimizing travel time subject to no waiting at any node except the destination, which is exactly the MATP (under FIFO). The case of $M = N \setminus \{1\}$ follows by equivalence to the $M = N \setminus \{n\}$ case, from Corollary 1. □

We have so far established that for specific choices of M and specific values of α and W, we recover MATP, MDP and MTTP. Hence, for these choices, the TDSPP-PW and TDSPP-LW are solvable in polynomial time. Table 2 shows the cases for which the complexity status has been established so far.

Table 2: Complexity results for variants of TDSPP-PW and TDSPP-LW obtained so far. Recall that the case of the TDSPP-PW with $\alpha = 0$ is precisely the MTTP, irrespective of the choice of M.

	TDSPP-PW		TDSPP-LW	
Tally Set	$0 < \alpha \leq 1$	$\alpha > 1$	$W = 0$	$W > 0$
$M = N$	Polytime (Prop. 2)			
$N \setminus M = \{1, n\}$		Polytime (Prop. 3)	Polytime (MDP)	
$N \setminus M = \{n\}$ or $\{1\}$		Polytime (Prop. 4)	Polytime (MATP)	

In what follows, we will provide results that allow us to complete the table, and to exhaustively resolve the complexity for all cases of the tally set M.

4 Cases that are NP-Hard

Some variants of TDSPP-PW and TDSPP-LW are generalizations of the exact path length (EPL) problem: given a directed graph $D = (N, A)$ with integer costs on each arc, an integer B and two nodes $i, j \in N$, determine whether there is a path in D from node i to node j with cost exactly B. Without loss of generality, we assume $i = 1$ and $j = n = |N|$.

Theorem 1. *(Nykänen and Ukkonen 2002) The EPL problem is weakly NP-hard even if the edge weights of the graph G are non-negative integers.*

Theorem 2. *The variant of TDSPP-PW in which waiting is penalized at every node ($M = N$) and the waiting cost is greater than the travel cost ($\alpha > 1$) is weakly NP-hard.*

Proof. We provide a reduction from EPL. Given an instance of the EPL as described above, construct an instance of the TDSPP-PW on the same graph, with constant (time-independent) travel times on arcs equal to the cost of the arc in EPL and with time horizon $[0, B]$. We claim that the optimal value of the TDSPP-PW is B if and only if the answer to EPL is 'Yes'. To see this, first consider any feasible timed path, P, for the TDSPP-PW. The sum of the EPL costs on the arcs used in P is precisely its travel time, $\tau(P)$. Since waiting is tallied at every node, $\tau(P)+\omega(P)=B$, and the objective value of P is $\tau(P)+\alpha\omega(P)$. Thus for any choice of $\alpha>1$, the optimal value of the TDSPP-PW is at least B, and can only equal B if there exists a timed path with no waiting. Thus the optimal value of the TDSPP-PW is B if and only if there is a feasible timed path, P, with $\omega(P)=0$, which is equivalent to $\tau(P)=B$, and the claim is proved. It is clear that this reduction is polynomial. EPL is NP-hard by Theorem 1, therefore this variant of TDSPP-PW is NP-hard. □

Theorem 3. *The variant of TDSPP-LW in which waiting is tallied at every node ($M=N$) is weakly NP-hard for $W\geq 0$.*

Proof. We provide a reduction from EPL to the problem of deciding whether or not the TDSPP-LW problem is feasible. Given an instance of the EPL as described above, construct an instance of the TDSPP-LW on the same graph, with constant (time-independent) travel times on arcs equal to the cost of the arc in EPL and with time horizon $[0,B]$. Now for any $W\in[0,1)$, there exists a feasible solution to the resulting TDSPP-LW instance if and only if the answer to EPL is 'Yes'. This is because any feasible solution, P, to the TDSPP-LW has $\tau(P)+\omega(P)=B$ and $0\leq\omega(P)\leq W<1$. But B and $\tau(B)$ are integers, so $\omega(P)$ is too: it must be that $\omega(P)=0$ and hence $\tau(P)=B$. Thus the TDSPP-LW instance has a feasible solution if and only if it has a timed path with travel time exactly B. However, the sum of the EPL costs on the arcs used in a timed path for the TDSPP-LW is precisely its travel time and the claim follows. It is clear that this reduction is polynomial. EPL is NP-hard by Theorem 1, therefore this variant of TDSPP-LW is NP-hard. □

Next, we consider the case of a general tally set M. Let $\{a_1,a_2,\ldots,a_n\}$ be a set of integers, $I=\{1,\ldots,n\}$, $\sum_{j\in I}a_j=2B$, and $b_i:=\sum_{j=1}^{i-1}3a_j$ for $i=1,\ldots,n$ (taking $b_1:=0$). Construct an instance of TDSPP-LW as follows:

- nodes $N=\{1,2,3,\ldots,2n-1,2n,2n+1\}$, with origin 1 and destination $2n+1$;
- arcs $A=\displaystyle\bigcup_{i=1,2,\ldots,n}\{(2i-1,2i),(2i-1,2i+1),(2i,2i+1)\}$;
- tally set $M=\{2i\mid i=1,2,\ldots,n\}$;
- arc travel time functions given by
  - □ $c_{2i-1,2i}(t)=|b_i-t|$, for all $i=1,2,\ldots,n$,
  - □ $c_{2i,2i+1}(t)=|b_{i+1}-t|$, for all $i=1,2,\ldots,n$ and
  - □ $c_{2i-1,2i+1}(t)=|b_i-t|+a_i$, for all $i=1,2,\ldots,n$, for all $t\in\mathbb{R}$;
- time horizon $[0,T]$ with $T=4B$; and
- tallied waiting time limit $W=3B$.

The network is shown in Figure 3 and the travel time functions are shown in Figure 4.

For this instance, all travel time function gradients are in $\{-1,1\}$, so the FIFO property is satisfied, but not strictly, Furthermore, the travel time functions are nonnegative at all times, but do not satisfy the positive travel time assumption that we usually require. The instance can

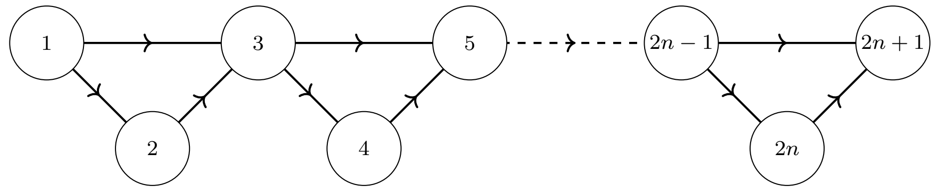
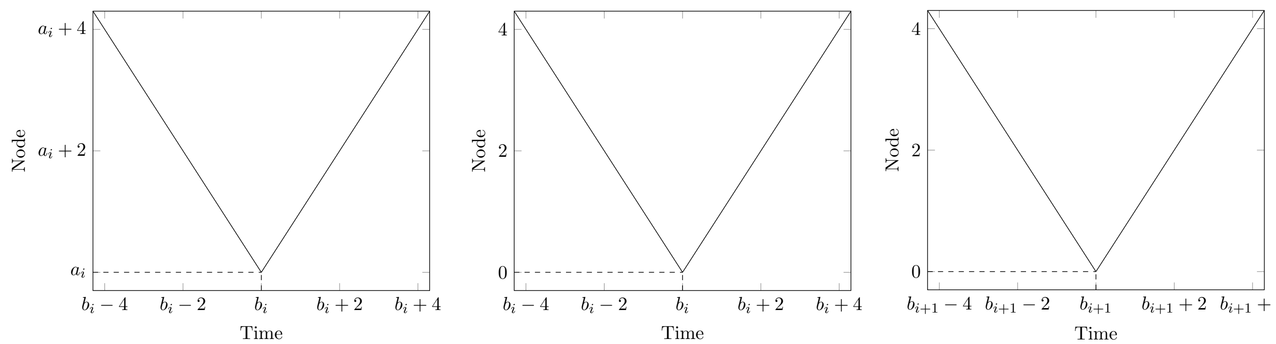


Figure 3: Network $D = (N, A)$.



(a) Travel time function $c_{2i-1,2i+1}$. **(b)** Travel time function $c_{2i-1,2i}$. **(c)** Travel time function $c_{2i,2i+1}$.

Figure 4: Travel time functions.

be modified slightly to satisfy these properties, and the arguments below can be adapted to the modified instance. For simplicity, we use the instance without these modifications.

The network $D = (N, A)$ comprises (overlapping) subnetworks $D_i = (N_i, A_i)$, where $N_i = \{2i-1, 2i, 2i+1\}$ and $A_i = \{(2i-1, 2i), (2i-1, 2i+1), (2i, 2i+1)\}$, for $i = 1, 2, \ldots, n$. To traverse D_i from node $2i-1$ to node $2i+1$, there are two options: either travel along path $p_i = (2i-1, 2i+1)$, using only odd numbered nodes, or along path $q_i = (2i-1, 2i, 2i+1)$, using one even numbered node.

For a timed path P from node 1 to node $2n+1$, we define $f(P) = \tau(P) + \frac{1}{3}\omega(P)$. Let P_i denote the part of P restricted to D_i. Thus P_i is a timed path using either the arc sequence of p_i or the arc sequence of q_i. Let $f_i(P)$ denote the contribution to $f(P)$ of the subpath in D_i, which must be either $\tau(P_i)$ if P_i uses p_i or $\tau(P_i) + \frac{1}{3}\omega(P_i) = \tau(P_i) + \frac{1}{3}\omega(P_i, 2i)$ if P_i uses q_i, since the only node in N_i that is in the tally set is node $2i$. Nodes with an odd index are not in the tally set M, thus waiting at odd indexed nodes does not need to be accounted for and hence $f(P) = \sum_{i=1}^{n} f_i(P)$.

Lemma 3. *For any $i \in I$ and any timed path S in D_i from node $2i-1$ to node $2i+1$, $f(S) \geq a_i$. Moreover, $f(S) = a_i$ only when (i) S departs node $2i-1$ at time b_i and (ii) if S uses q_i then S departs node $2i$ at time $b_i + 3a_i$.*

Proof. Let $i \in I$ and S be a timed path in D_i from node $2i-1$ to node $2i+1$. If S uses p_i, then since travel time function $c_{2i-1,2i+1}(t) \geq a_i$ for all t, we have $f(S) \geq a_i$. Moreover, $c_{2i-1,2i+1}(t) = a_i$ if and only if $t = b_i$, and the result follows.

Otherwise, it must be that S uses q_i. Consider the timed path $S^* = ((2i-1, b_i), (2i, b_i), (2i, b_i + 3a_i), (2_i + 1, b_i + 3a_i))$, which departs at node $2i-1$ at time b_i and departs node $2i$ at time $b_{i+1} = b_i + 3a_i$, has travel time $\tau(S^*) = 0$ and waiting time $\omega(S^*) = 3a_i$, and, thus, $f(S^*) = a_i$. We now show that this is a local minimum of f for timed paths using q_i.

Observe that departing earlier at node $2i-1$ or departing later at node $2i$ results in an increase in travel time and an increase in waiting time at node $2i$, and so cannot lead to a reduction in the value of f. Also, departing node $2i-1$ at time $b_i+\delta_1$ for small $\delta_1>0$ results in an increase in travel time of δ_1 and a decrease in waiting time at node $2i$ of $2\delta_1$, giving a change in $f(S^*)$ of $\delta_1-\frac{2}{3}\delta_1=\frac{1}{3}\delta_1>0$. Departing node $2i$ at time $b_i+3a_i-\delta_2$ for small $\delta_2>0$ results in an increase in travel time of δ_2 and a decrease in waiting time at node $2i$ of δ_2, giving a change in $f(S^*)$ of $\delta_2-\frac{1}{3}\delta_2=\frac{2}{3}\delta_2>0$. Due to the positive waiting time at node $2i$ in S^*, small changes in these departure times can be carried out independently and the impact on f is additive. Thus S^* is a strict local minimizer of f for timed paths using q_i.

It is clear that f for any timed path using q_i is a convex function of $(t_1,t_2)\in\mathbb{R}^2$ where t_1 is the departure time at node $2i-1$ and t_2 is the departure time at node $2i$. Hence S^* must be a unique global minimum. □

The result below follows directly from Lemma 3, the definition of the travel time functions and the definition of b_{i+1}, which gives $b_{i+1}=b_i+3a_i$.

Corollary 2. *For any $i\in\{1,\dots,n\}$ and any timed path S in D_i from node $2i-1$ to node $2i+1$, if $f(S)=a_i$ then (i) S arrives at node $2i+1$ no later than b_{i+1}, (ii) if S uses p_i then $\tau(S)=a_i$ and $\omega(S)=0$, and (iii) if S uses q_i then $\tau(S)=0$ and $\omega(S)=3a_i$.*

Lemma 4. *Any feasible timed path P with $\tau(P)\le B$ satisfies $f(P)\le 2B$.*

Proof. A feasible timed path P has $\omega(P)\le 3B$. Hence, $f(P)=\tau(P)+\frac{1}{3}\omega(P)\le 2B$. □

Lemma 5. *Any timed path P has $f(P)\ge 2B$. Moreover, $f(P)=2B$ only when for all $i\in I$, P_i departs node $2i-1$ at time b_i and if P_i uses q_i then P departs node $2i$ at time b_i+3a_i.*

Proof. By Lemma 3, $f(P_i)\ge a_i$ for each $i\in I$. As already noted $f(P)=\sum_{i\in I}f(P_i)\ge\sum_{i\in I}a_i=2B$. Obviously, if $f(P)=2B$ then $f(P_i)=a_i$ for all $i\in I$, and Lemma 3 gives the conditions for this to hold. □

Theorem 4. *The variant of* TDSPP-LW *in which waiting is constrained at a subset of nodes $M\subseteq N$ and there is a positive limit on total waiting ($W>0$) is NP-hard.*

Proof. We prove that its decision version, in which we want decide if there is path with travel time less than a given constant, is NP-Complete by a transformation from Partition.

Partition: Given a set of integers $\{a_1,a_2,\dots,a_n\}$, does there exist a subset $S\subseteq I=\{1,\dots,n\}$ such that $\sum_{j\in S}a_j=\frac{1}{2}\sum_{j\in I}a_j=B$?

The transformation takes an instance of Partition and constructs an instance of TDSPP-LW as outlined at the start of this section and takes the constant to be B, i.e., we want to decide if there is a feasible timed path with travel time less than or equal to B.

First, we show that a YES instance of Partition induces a YES instance of TDSPP-LW. Let $S\subset I$ with $\sum_{i\in S}a_i=B$. Take the timed path P formed by concatenating, for $i=1,\dots,n$, the timed path that uses p_i whenever $i\in S$ and q_i whenever $i\notin S$, with the departure times specified in Lemma 3. Such concatenation is feasible by Corollary 2. By construction, $\tau(P)=B$ and $\omega(P)=3B$.

Next, we show that a YES instance of TDSPP-LW with constant B induces a YES instance of PARTITION. Lemmas 4 and 5 imply that a feasible timed path P with $\tau(P) \leq B$ has $f(P) = 2B$. Now since the tallied waiting time limit $W = 3B$, it must be that $\omega(P) \leq 3B$, so

$$f(P) = \tau(P) + \tfrac{1}{3}\omega(P) = 2B \quad \Rightarrow \quad \tau(P) + \tfrac{1}{3}(3B) \geq 2B \quad \Rightarrow \quad \tau(P) \geq B.$$

Hence $\tau(P) = B$ and $\omega(P) = 3B$. Furthermore, since equality holds, by Lemma 5 and Corollary 2, $\tau(P) = \sum_{i \in I: P_i \text{ uses } p_i} \tau(P_i)$. Thus selecting i to be in S whenever P_i uses p_i is chosen (and not selecting i to be in S whenever q_i is used), a feasible solution to the instance of PARTITION is obtained, since

$$B = \tau(P) = \sum_{i \in I: P_i \text{ uses } p_i} \tau(P_i) = \sum_{i \in S} a_i,$$

by construction. □

Because neither 1 nor n are in the tally set M in the transformation used to prove Theorem 4, we have the following theorem as an immediate consequence.

Theorem 5. *The variant of* TDSPP-LW *in which waiting is constrained at a subset of nodes $M \subseteq N$ with $1, n \notin M$ and a positive limit on total waiting ($W > 0$) is NP-hard.*

5 Cases of TDSPP-PW Solvable in Polynomial Time via a Time-Expanded Network

We now discuss cases that can be reduced to solving a (standard) shortest path problem in a time-expanded network that has size polynomial in the size of the given instance. We first explain how the time-expanded network is constructed.

5.1 The Time-Expanded Network

The time-expanded networks central to our polynomial time complexity results are constructed as a finite set of *timed nodes*, each a node-time pair of the form (i, t) with $i \in N$ and $t \in [0, T]$, and *timed arcs*, of the form $((i, t), (j, t'))$ where (i, t) and (j, t') are timed nodes and $t' \geq t$. We also refer to a timed node (i, t) as a *timed copy of* i. The timed node set is constructed by solving one MATP and one reverse MATP (introduced in Section 2.2.1) for each breakpoint of an arc travel time function, as follows. For arc a outgoing from node i and time t a breakpoint of its travel time function:

- (i, t) is included in the set of timed nodes,
- for each $j \in N \setminus \{i\}$ reachable from (i, t) within the time horizon, (j, t') is included in the set of timed nodes, where t' is the earliest arrival time at j if departing i at time t (the value of the MATP with origin node i, destination node j and time horizon $[t, T]$), and
- for each $j \in N \setminus \{i\}$ with i reachable from $(j, 0)$ no later than t, (j, t') is included in the set of timed nodes, where t' is the latest departure time from j to arrive at node i at time t (the value of the reverse MATP with origin node j, destination node i and time horizon $[0, t]$).

Algorithms for the MATP, described in Cooke and Halsey (1966), Orda and Rom (1990), Dean (2004b), for example, can easily be adapted to find the minimum arrival time path to *all* nodes in $N \setminus \{i\}$ for the same computational effort as finding the minimum arrival time path to just one destination. Such algorithms ensure that if the minimum arrival time path departing from node i at time t arrives at node k at time t'' along arc (j,k), having departed j at time t', then t' is the minimum arrival time at j starting from (i,t). Thus such algorithms generate a timed version of a forward shortest path tree, with at most one timed node per node in the network. As per Proposition 1, reverse MATP is equivalent to MATP; solving reverse MATP generates a timed version of a backward shortest path tree. Hence, the number of timed nodes in the time-expanded network is at most $2nK$, where K is the total number of travel time function breakpoints.

The set of timed arcs is constructed by including $((i,t),(j,t'))$ whenever arc $(i,j) \in A$ is traversed starting at time t in any such forward or backward shortest path tree. In this case it must be that $t' = t + c_{i,j}(t)$. We call such timed arcs *travel arcs*. We say that the *travel time* of timed arc $((i,t),(j,t'))$ is $c_{i,j}(t) = t' - t$. In addition, the set of timed arcs also includes *waiting arcs*: if $(i,t_1),(i,t_2),\ldots,(i,t_r)$ is the set of all timed copies of i, listed in increasing chronological order, then $((i,t_{s-1}),(i,t_s))$, which is referred to as a waiting arc, is included in the set of timed arcs, for each $s = 2,\ldots,r$.

Clearly any path in the TEN corresponds to a timed path in the original network, which can be expressed simply as the timed node sequence in the TEN path, omitting intermediate timed copies of the same node whenever more than two appear consecutively.

In what follows, we will show that for all remaining cases of the TDSPP-PW, there exists an optimal solution to any instance that corresponds to a path in its TEN. Furthermore, we will be able to construct lengths for the arcs in the TEN so that any shortest path, with respect to these lengths, from $(1,0)$ to (n,T) in the TEN corresponds to an optimal solution to the TDSPP-PW instance.

5.2 Preliminaries

Definition 2. *A* travel subpath *of a given timed path is a maximal consecutive sequence of timed nodes in the timed path with no consecutive timed copies of the same node.*

Thus a timed path is the concatenation of a sequence of travel subpaths. Note that a node is a start or end of a travel subpath if and only if the node is 1 and the travel subpath starts with $(1,0)$, or n and the travel subpath ends with (n,T), or if the node is in $N^w(P)$.

To illustrate, consider the example and the feasible timed path $P = ((1,0),(1,2),(2,2.14),(2,2.92),(3,4.05),(4,5))$. Here $N^w(P) = \{1,2\}$ with $\omega(P,1) = 2$ and $\omega(P,2) = 0.78$, and P has three travel subpaths: $((1,0))$, $((1,2),(2,2.14))$ and $((2,2.92),(3,4.05),(4,5))$.

If a timed path consists of three of more travel subpaths then we say that all but the first and last travel subpaths are *intermediate* subpaths. A travel subpath is, itself, a timed path, and so properties of timed paths and terms used to describe them may also be used for travel subpaths.

A key feature of a travel subpath is whether or not it contains a breakpoint. Note that we consider time 0 to be a breakpoint at node 1 and time T to be a breakpoint at node n, regardless of whether the travel time function for an arc leaving or entering these nodes, respectively, has a breakpoint at these respective times.

Definition 3. *A travel subpath* contains a breakpoint *if it starts with* $(1,0)$*, or ends with* (n,T)*, or if there is an arc* $(i,j) \in A$ *and a time* t *that is a breakpoint of* $c_{i,j}(\cdot)$ *for which the timed node*

(i, t) appears in the travel subpath.

Our argument that some solution to a TDSPP-PW instance must occur in its TEN involves *shifting* travel subpaths that do not contain a breakpoint. We thus make use of the following idea.

Definition 4. *A travel subpath $P(\Delta)$ is a Δ-*shifting *of travel subpath $P = ((i_1, t_1), \ldots, (i_K, t_K))$ if $P(\Delta) = ((i_1, t_1'), \ldots, (i_k, t_K'))$ where $t_1' = t_1 + \Delta$ and $t_k' = t_{k-1}' + c_{i_{k-1},i_k}(t_{k-1}')$ for $k = 2, \ldots, K$.*

Note that, from Observation 1, for $P(\Delta)$ a Δ-shifting of P, if $\Delta < 0$ then $t_k' < t_k$ for all $k = 1, \ldots, K$, while if $\Delta > 0$ then $t_k' > t_k$ for all $k = 1, \ldots, K$.

As consequence of properties of compositions of piecewise affine functions, if a travel subpath does not contain a breakpoint, then its travel time is a locally affine function of the departure time on its first node.

Observation 2. *If travel subpath P does not contain a breakpoint, then there exist $\varepsilon^-, \varepsilon^+ > 0$ such that $\tau(P(\Delta))$ is an affine function of Δ for all $\Delta \in (-\varepsilon^-, \varepsilon^+)$. Furthermore, if $\varepsilon^-, \varepsilon^+$ are the maximal such values, then both $P(-\varepsilon^-)$ and $P(\varepsilon^+)$ contain a breakpoint.*

In what follows, when we *shift P back to a breakpoint*, we mean that we replace P by $P(-\varepsilon^-)$ where $\varepsilon^- \geq 0$ is the largest value such that $P(\delta)$ does not contain a breakpoint for all $\delta \in (-\varepsilon^-, 0)$. Similarly, we *shift P forward to a breakpoint* by replacing P with $P(\varepsilon^+)$ where $\varepsilon^+ \geq 0$ is the largest value such that $P(\delta)$ does not contain a breakpoint for all $\delta \in (0, \varepsilon^+)$.

5.3 Complexity Results

In this section, we consider the TDSPP-PW with either waiting cost not more than travel cost ($\alpha \leq 1$) or at least one node at which waiting is not tallied ($M \subset N$). (Recall that the only remaining case of TDSPP-PW, namely that with $\alpha > 1$ and $M = N$, is NP-hard, shown in Theorem 2.) Our approach is structured as follows.

We first show that there is an optimal solution to any TDSPP-PW instance with each of its travel subpaths containing a breakpoint. We then show that if either $\alpha \leq 1$ or there is an optimal solution with zero tallied waiting, then there must be an optimal solution in which each travel subpath consists of two concatenated MATP solutions to/from a breakpoint. Thus each travel subpath corresponds to a sequence of timed arcs of precisely the form of the travel arcs in the TEN for the instance.

It is then straightforward to prove that solving a TDSPP-PW instance with $\alpha \leq 1$ can be done by solving a shortest path problem in its TEN. The case of $\alpha > 1$ and $M \neq N$ requires some additional results, to establish that there is an optimal solution which has zero tallied waiting. We conclude by showing that any instance of the TDSPP-PW with $\alpha > 1$ and $M \subset N$ can be solved in polynomial time, by solving a shortest path in its associated TEN.

Our first result generalizes that of Foschini et al. (2014) for the MDP.

Proposition 5. *For any instance of TDSPP-PW, there exists an optimal timed path such that each of its travel subpaths includes at least one timed node at a breakpoint. Furthermore, if P is any optimal solution, there exists P', also an optimal solution, with a breakpoint in each of its travel subpaths and with $N^w(P') \subseteq N^w(P)$.*

Proof. Suppose that P is an optimal timed path having a travel subpath, S, that does *not* include a breakpoint. Recall that for P to be feasible, it must start with $(1, 0)$ and end with (n, T), which are considered to be breakpoints. So S must be an intermediate travel subpath. Let τ^- be the end time of the travel subpath immediately preceding S and let τ^+ be the start time of the travel subpath immediately after S in P.

We claim that there is a Δ-shifting of S so that (i) $S(\Delta)$ is contained in $[\tau^-, \tau^+]$, (ii) the path formed by replacing S in P by $S(\Delta)$ is optimal, and (iii) either $S(\Delta)$ contains a breakpoint, or it starts at τ^- or ends at τ^+. In the latter two cases, $S(\Delta)$ is no longer a travel subpath of the new optimal path say, since it is not maximally waiting-free; it concatenates with either the preceding travel subpath or the next travel subpath in P to form a longer travel subpath. Thus the new optimal path has either one more travel subpath containing a breakpoint or it has one fewer travel subpath than the original optimal path, P. This procedure cannot introduce waiting at any node where there was no waiting in P. Applying this procedure repeatedly must end with an optimal path, P', in which every travel subpath contains a breakpoint, and $N^w(P') \subseteq N^w(P)$, as required. We now prove the claim.

First, apply Observation 2 to travel subpath S, so that $\tau(S(\Delta))$ is affine in Δ, given by $\tau(S(\Delta)) = m\Delta + \tau(S)$ for some $m \in \mathbb{R}$, for all $\Delta \in (-\varepsilon^-, \varepsilon^+)$, with $\varepsilon^-, \varepsilon^+ > 0$ and maximal.

Suppose S starts at node i and ends at node j. For $\Delta \in (-\varepsilon^-, \varepsilon^+)$ such that $S(\Delta)$ is contained in $[\tau^-, \tau^+]$, replacing S with $S(\Delta)$ in P to create a new feasible path, P', will result in

$$\tau(P') = \tau(P) + m\Delta, \quad \omega(P', i) = \omega(P, i) + \Delta \quad \text{and} \quad \omega(P', j) = \omega(P, j) - (1+m)\Delta.$$

Thus

$$\begin{aligned}
\tau(P') + \alpha\omega(P') &= \tau(P) + \alpha\omega(P) + \left\{ \begin{array}{ll} m\Delta, & \text{if } i, j \notin M, \\ m\Delta + \alpha(\Delta - (1+m)\Delta), & \text{if } i, j \in M \\ m\Delta + \alpha\Delta, & \text{if } i \in M, j \notin M \\ m\Delta - \alpha(1+m)\Delta, & \text{if } i \notin M, j \in M \end{array} \right\}. \\
&= \tau(P) + \alpha\omega(P) + \left\{ \begin{array}{ll} m\Delta, & \text{if } i, j \notin M, \\ (1-\alpha)m\Delta, & \text{if } i, j \in M \\ (m+\alpha)\Delta, & \text{if } i \in M, j \notin M \\ (m - \alpha(1+m))\Delta, & \text{if } i \notin M, j \in M. \end{array} \right\}.
\end{aligned}$$

In each case, m and α (if relevant) must be such that the coefficient of Δ is zero, otherwise some positive or negative value of Δ creates a new path with better TDSPP-PW objective value than P, which is optimal. In other words, the coefficient of Δ is zero in every case, otherwise Δ can be chosen to contradict optimality of P. Hence $\Delta = -\min\{\varepsilon^-, t - \tau^-\}$ must satisfy the claim, where t is the start time of S. In the case that $\varepsilon^- \leq t - \tau^-$, by Observation 2, the Δ-shifting of S will contain a breakpoint; otherwise it will start at τ^-. (Note: Δ may equally well be set to ε or the value for which $t + \tau(S(\Delta)) \leq \tau^+$, whichever is smaller. In the latter case, $S(\Delta)$ will end at τ^+.) $\square$

Recall that timed nodes in the TEN are created by solving an MATP starting from each breakpoint, forward, to find the earliest time each node can be reached from the breakpoint, and by solving a reverse MATP from each breakpoint, backward, to find the latest departure time at each node from which the breakpoint can be reached. We call a path solving a forward MATP an *earliest arrival time path* and a path solving a backward MATP a *latest departure time path*. By the FIFO property, these paths do not include any waiting.

Lemma 6. *For any instance of the TDSPP-PW satisfying (i) $\alpha \leq 1$ or (ii) there is an optimal solution, P, with $\omega(P) = 0$, there exists an optimal timed path, P', such that each of its travel subpaths consists of a latest departure time path from a node to a breakpoint concatenated with an earliest arrival time path from the same breakpoint to another node. In case (ii), $\omega(P') = 0$.*

Proof. Consider an instance of the TDSPP-PW satisfying the required conditions. If $\alpha \leq 1$, apply Proposition 5, to obtain an optimal timed path, P such that each of its travel subpaths contains a breakpoint. Otherwise, let P^* be an optimal solution with $\omega(P^*) = 0$, so travel subpaths start or end at nodes $N^w(P^*) \subseteq N(P^*) \setminus M$. Apply Proposition 5, to obtain a new optimal solution, P, with each of its travel subpaths containing a breakpoint and $N^w(P) \subseteq N^w(P^*)$. So $\omega(P) = 0$.

Suppose that S is a travel subpath of P that is *not* the concatenation of a latest arrival path and an earliest arrival path. Suppose S starts with (i, t_i), includes breakpoint (k, t_k) and ends with (j, t_j). Let S^b be a latest departure path from i to (k, t_k), and say it starts with (i, t'_i). By optimality of S^b, it must be that $t'_i \geq t_i$. Let S^f be an earliest arrival path from (k, t_k) to j, and say it ends with (j, t'_j). It must be that $t'_j \leq t_j$. Let P' denote the path formed by replacing S in P by S^b concatenated with S^f, which do not include any waiting. Then

$$
\begin{aligned}
\tau(P') &= \tau(P) - (t'_i - t_i) - (t_j - t'_j), \\
\omega(P', i) &= \omega(P, i) + (t'_i - t_i), \text{ and} \\
\omega(P', j) &= \omega(P, j) + (t_j - t'_j).
\end{aligned}
$$

Now define $\alpha_h = \alpha$ if $h \in M$ and $\alpha_h = 0$ otherwise, for $h = i, j$. By the conditions of the lemma, either $\alpha \leq 1$ or $\omega(P) = 0$, so since $\omega(P, i),\ \omega(P, j) > 0$, it must be that $i, j \notin M$, so $\alpha_i = \alpha_j = 0$. Thus, in every case, $\alpha_i,\ \alpha_j \leq 1$. As a consequence,

$$
\begin{aligned}
\tau(P') + \alpha\omega(P') &= \tau(P) - (t'_i - t_i) - (t_j - t'_j) + \alpha\omega(P) + \alpha_i(t'_i - t_i) + \alpha_j(t_j - t'_j) \\
&= \tau(P) + \alpha\omega(P) - (1 - \alpha_i)(t'_i - t_i) - (1 - \alpha_j)(t_j - t'_j) \\
&\leq \tau(P) + \alpha\omega(P)
\end{aligned}
$$

since $t'_i \geq t_i$ and $t'_j \leq t_j$. Since P is optimal for the TDSPP-PW instance, it must be that P' is optimal, too. Furthermore, $N^w(P') = N^w(P)$, so if $\omega(P) = 0$ then $\omega(P') = 0$ too. This procedure can be repeated until an optimal solution satisfying the conditions of the lemma is generated. $\square$

Theorem 6. *The variant of TDSPP-PW in which waiting costs less than traveling, i.e., $0 < \alpha \leq 1$, is solvable in polynomial time.*

Proof. By Lemma 6, there exists an optimal timed path P consisting of a sequence of travel subpaths, each of which is the concatenation of a latest departure time path to a breakpoint and an earliest arrival time path from the same breakpoint. By construction, all timed nodes and timed arcs in these paths are included in the TEN. Furthermore, if a travel subpath of P ends with (i, t) and the travel subpath immediately after it in P starts with (i, t'), then by definition of the travel subpaths, $t' > t$. And by construction of the TEN, there must be a sequence of waiting arcs in the TEN forming a path from (i, t) to (i, t').

Define the length of each travel arc in the TEN to be its travel time. Define the length of each waiting arc in the TEN, of the form $((i, t), (i, t'))$ to be $\alpha(t' - t)$ if $i \in M$ and zero otherwise. Then clearly any path from $(1, 0)$ to (n, T) in the TEN corresponds to a feasible timed path for the

TDSPP-PW and the length of the TEN path is precisely the corresponding timed path's TDSPP-PW objective value. Thus solving a shortest path problem in the TEN with the given lengths must yield an optimal solution to the TDSPP-PW. The TEN has $\mathcal{O}(2nK)$ nodes, for K the number of breakpoints, and hence the TDSPP-PW can be solved in polynomial time. $\square$

We now turn our attention to the case of TDSPP-PW with waiting cost greater than travel cost. Throughout the remainder of this section, "an instance" means an instance of the TDSPP-PW with $\alpha > 1$ and $M \subset N$ (so $M \neq N$).

We begin by establishing that any optimal solution, P^*, to an instance must have zero tallied waiting, i.e., $\omega(P^*) = 0$ for any optimal timed path P^*. We do this in two steps. First, we show that if every node at which a timed path waits is in the tally set, then the path cannot be optimal.

Lemma 7. *If P is a feasible solution for an instance with the property that $\omega(P, i) = 0$ for all $i \in N(P)$ with $i \notin M$, then either P is waiting-free or P is not optimal.*

Proof. If P is as described, then it must be that $\tau(P) + \omega(P) = T$. Suppose P is not waiting-free. Then it must wait at a node in M, so $\omega(P) > 0$. Thus the TDSPP-PW objective value of P is $\tau(P) + \alpha\omega(P) > T$, since $\alpha > 1$. Now $M \neq N$ so there exists $i \in N \setminus M$. Let S^f be the solution to the MATP with origin 1 and destination i, starting at time 0, giving earliest arrival time at i of t. Let S^b be the solution to the reverse MATP that determines the latest departure time from i so as to arrive at n at time T; let this time be t'. Then, by Assumption 3, $t' \geq t$ and the concatenation of S^f followed by S^b is a feasible timed path; denote this path by P'. By FIFO, the only node at which P' waits is node $i \notin M$. Thus $\omega(P') = 0$ and $\tau(P') + \alpha\omega(P') = \tau(P') \leq T < \tau(P) + \alpha\omega(P)$. Hence P cannot be optimal. $\square$

We now show that if P^* is an optimal solution that is not waiting-free, then it cannot wait at *any* node in M. The proof makes use of the observation that if a travel subpath of a feasible timed path, P, ends at a node other than the destination, say the travel subpath ends at node $i \neq n$, then the timed path must wait at i, i.e., $\omega(P, i) > 0$. Also, if $\omega(P, i) > 0$ for some feasible P and some node i, it must be that a travel subpath either ends at i, starts at i, or both.

Proposition 6. *If P^* is an optimal solution for an instance, then $\omega(P^*) = 0$.*

Proof. If P^* is waiting-free, then the result follows. Otherwise, suppose, for contradiction, that $\omega(P^*) > 0$, so there exists $i \in M$ with $\omega(P^*, i) > 0$. By Lemma 7, there must exist $j \notin M$ with $\omega(P^*, j) > 0$. There are two cases to consider: either i appears before j in the path, or vice versa. The two cases are symmetric, and the proof for one case easily adapted to the other, so we only give the proof for the case that i appears before j in P^*.

We claim that there must exist a pair of nodes $\hat{i} \in N(P^*) \cap M$ and $\hat{j} \in N(P^*) \setminus M$ so that a travel subpath of P^* starts at $\hat{i}$ and ends at $\hat{j}$, with $\omega(P^*, \hat{i})$, $\omega(P^*, \hat{j}) > 0$. To substantiate the claim, observe that some travel subpath of P^* must start at node i. If the end of that travel subpath is also in M, say it ends at node $i' \in M$, then $i' \neq n$ (since node $j \notin M$ must appear later), so it must be that $\omega(P^*, i') > 0$, and we may replace i with i', which appears later in P^*. This procedure can be repeated until the end of the travel subpath starting at i is *not* in M, say it ends at node $j' \notin M$. Then either $j' = j$ and the claim follows, or $j' \neq n$ since j' must appear before j in P^*. Thus, since j' ends a travel subpath and is not the destination node, it must be that $\omega(P^*, j') > 0$. Taking $\hat{i} = i$ and $\hat{j} = j'$ satisfies the claim.

Let $\hat{\imath}, \hat{\jmath}$ be as claimed. Let $(\hat{\imath}, s), (\hat{\imath}, s')$ with $s' > s$ denote the consecutive node-time pairs in P^* that appear before the consecutive node-time pairs $(\hat{\jmath}, t), (\hat{\jmath}, t')$ with $t' > t$ and let Q denote the travel subpath of P^* that starts with $(\hat{\imath}, s')$ and ends with $(\hat{\jmath}, t)$. Let $\Delta = s - s' < 0$ and consider $Q(\Delta)$, the Δ-shifting of Q: it is a waiting-free timed path that starts with $(\hat{\imath}, s)$ and ends at $\hat{\jmath}$ at some time earlier than t. The path, P, formed by replacing Q in P^* by $Q(\Delta)$ is feasible for the instance. Furthermore, $\tau(Q(\Delta)) < t - s = \tau(Q) + s' - s$ and $\omega(P, i) = 0$. Thus, since $\hat{\jmath} \notin M$, we have that

$$\begin{aligned}
\tau(P) + \alpha\omega(P) &= (\tau(P^*) - \tau(Q) + \tau(Q(\Delta))) + \alpha(\omega(P^*) - \omega(P^*, i) + \omega(P, i)) \\
&< (\tau(P^*) + s' - s) + \alpha(\omega(P^*) - (s' - s)) \\
&= \tau(P^*) + \alpha\omega(P^*) + (1 - \alpha)(s' - s) \\
&< \tau(P^*) + \alpha\omega(P^*)
\end{aligned}$$

since $\alpha > 1$ and $s' < s$. This contradicts the optimality of P^*. □

We are now able to show, as a consequence of the above result, that there is an optimal solution for an instance that has a corresponding path in its TEN. We do so by first establishing that every travel subpath of an optimal solution must solve an MDP over the time interval containing both the travel subpath and any waiting before or after it in the optimal solution.

Proposition 7. *For any instance, there is a path in its TEN that corresponds to an optimal timed path for the instance and that does not use any waiting arcs at nodes in M.*

Proof. Let P be an optimal timed path for the instance with $\omega(P) = 0$, known to exist by Proposition 6. By Lemma 6, there exists another optimal timed path, P', such that every travel subpath consists of a latest departure time path to a breakpoint concatenated with an earliest arrival path from the same breakpoint. Thus every travel subpath is contained in the TEN for the instance, which also includes waiting arcs to link all timed copies of the same node. Since Lemma 6 also guarantees that $\omega(P') = 0$, no waiting arc at a node in M is used. The result follows. □

Theorem 7. *The variant of TDSPP-PW in which waiting is penalized at nodes $M \subset N$ and the waiting cost is greater than the travel cost, i.e., $\alpha > 1$, is solvable in polynomial time.*

Proof. Consider an instance of this variant of TDSPP-PW, and its associated TEN. Remove all waiting arcs at each node in M. Define the length of each travel arc in the TEN to be its travel time. Define the length of each (remaining) waiting arc to be zero. Now the length of every path from $(1, 0)$ to (n, T) in the TEN is its total travel time, which is also its TDSPP-PW objective, since no waiting at arcs in M is possible. Every path in the TEN corresponds to a feasible solution of the TDSPP-PW, and, by Proposition 7, the optimal TDSPP-PW solution has a corresponding path in the TEN. The result follows. □

6 Cases of TDSPP-LW Solvable in Polynomial Time

6.1 Waiting not allowed ($W = 0$) at a subset of nodes $M \subset N$

When $W = 0$, i.e., waiting at nodes in M is not allowed, the TDSPP-LW resembles TDSPP-PW with $\alpha > 1$, where waiting at nodes in M is discouraged. Indeed, the proof that this variant is solvable in polynomial time relies on showing that this variant of TDSPP-LW has a relaxation which is a variant of TDSPP-PW, and that solving this relaxation yields an optimal solution to the TDSPP-LW.

Theorem 8. *The variant of TDSPP-LW in which waiting is not allowed ($W = 0$) at a subset of nodes $M \subset N$ is solvable in polynomial time.*

Proof. Any feasible path for an instance of TDSPP-LW with $W = 0$ is a feasible path for the corresponding instance of TDSPP-PW with $\alpha > 1$ (i.e., with the same tally set M) having the same objective value. Hence, TDSPP-PW with $\alpha > 1$ is a relaxation of TDSPP-LW with $W = 0$. By Theorem 7, solving the variant of TDSPP-PW with $M \subset N$ with $\alpha > 1$ can be done in polynomial time and, by Proposition 7, yields an optimal TDSPP-PW solution P^* that does not use any waiting arcs at nodes in M. In other words, for $i \in M$, we have that $\omega(P^*, i) = 0$, hence P^* is feasible for the TDSPP-LW with $W = 0$. Since P^* is an optimal solution to a relaxation of TDSPP-LW and is also feasible, it is also an optimal solution of TDSPP-LW, and has been found in polynomial time. □

6.2 Waiting constrained at $M \subset N$ with $M = N \setminus \{n\}$ and a positive limit on total waiting time ($W > 0$)

The proofs in this section only use that $n \notin M$, hence also apply to the case where $M = N \setminus \{1, n\}$. For the case where $M = N \setminus \{1\}$, apply the transformation used in the proof of Proposition 1 to return an instance with $M = N \setminus \{n\}$.

Definition 5. *A* non-trivial *travel subpath is a travel subpath that starts and ends at timed copies of different nodes. In our context, the only possible trivial travel subpaths are* $(1, 0)$ *and* (n, T)*.*

Definition 6. *A timed path P that begins at timed node (i, t_i) and ends at timed node (j, t_j) can be* completed *by prepending a latest departure time path from 1 to i (arriving at node i at time t_i) and appending an earliest arrival time path from j to n (departing from node j at time t_j), and, if the resulting timed path is within the time horizon, adding timed node $(1, 0)$ at the beginning and timed node (n, T) at the end, to form a feasible timed path P' for the MTTP. If in addition, $\omega(P') \leq W$ then P' is feasible for the TDSPP-LW. If the resulting timed path has timed nodes outside the time horizon, then the timed path P cannot be extended to a feasible solution for the TDSPP-LW and is removed from consideration.*

We begin by introducing three progressively more specific properties which an optimal timed path P may have. We then show that if an optimal timed path exists, it is always possible to convert it to an optimal path with each of these properties. Importantly, we show that if an optimal path with the most specific property exists, it can be found in polynomial time.

Property 1. *All travel subpaths of P except possibly the last non-trivial travel subpath contain a breakpoint. If $\omega(P) < W$, then the last non-trivial travel subpath also contains a breakpoint.*

Property 2. *Each travel subpath of P consists of a latest departure time path to a timed node concatenated with an earliest arrival time path from the same timed node, where the timed node is a breakpoint if one exists in the travel subpath.*

Property 3. *Let R_1, R_2 and R_3 be the first, second-to-last (if it exists), and last non-trivial travel subpaths of P, respectively, then R_1 starts at timed node $(1, t_1)$ with $t_1 \in \{t_1^1, t_1^2, \dots t_1^K\}$ where t_1^k is the latest departure time from node 1 for arrival at the k^{th} breakpoint (given an arbitrary ordering of the breakpoints), R_2 ends at timed node (j, t_j) with $t_j \in \{t_j^1, t_j^2, \dots t_j^K\}$ where t_j^k is the earliest arrival time at node j when departing at the k^{th} breakpoint, and R_3 ends at timed node (n, t_n) with $t_n \in \{t_n^1, t_n^2, \dots t_n^K\}$ where t_n^k is the earliest arrival time at node n when departing at the k^{th} breakpoint.*

If P contains only one non-trivial travel subpath, then $R_1 = R_3$ solves the MATP from timed node $(1, t_1)$ via a breakpoint to timed node (n, t_n), where the allowable sets for t_1 and t_n are constructed while also considering $(1, W)$ as a breakpoint. Moreover, if the breakpoint is the k^{th} breakpoint, then $t_1 = t_1^k$ and $t_n = t_n^k$. In other words, P is the completion of a timed path with only one timed node (i, t), which is a breakpoint.

In the case where P contains more than one non-trivial travel subpath, if $\omega(P) < W$, the subpath of P that starts with R_1 and ends with R_3 solves the MTTP from node 1 to node n with time horizon $[t_1, t_n]$. Otherwise, if $\omega(P) = W$, the subpath of P that starts with R_1 and ends with R_2 solves the MTTP from node 1 to node j with time horizon $[t_1, t_j]$, where, in case there are only two non-trivial subpaths, we set $R_2 = R_3$.

Lemma 8. *Suppose there exists an optimal timed path, P, such that Property 3 holds, then an optimal timed path can be found in polynomial time.*

Proof. If P contains only one non-trivial travel subpath, then P must be one of the K timed paths found by completing the K timed path containing only one timed node which is a breakpoint. Selecting the resulting timed path that is feasible for the TDSPP-LW with the least total travel time must return an optimal solution since one of the solutions returns P. This can be done in $\mathcal{O}(2K(MATP))$ time. Thus in this case, an optimal solution can be found in polynomial time.

Otherwise, if P contains more than one non-trivial travel subpath, we split into the two cases $\omega(P) < W$ and $\omega(P) = W$. Let R_1, R_2 and R_3 be the first, second-to-last, and last non-trivial travel subpaths of P, respectively, then by Property 3, R_1 starts at timed node $(1, t_1)$ with $t_1 \in \{t_1^1, t_1^2, \dots t_1^K\}$ where t_1^k is the latest departure time from node 1 for arrival at the k^{th} breakpoint (given an arbitrary ordering of the breakpoints), R_2 ends at timed node (j, t_j) with $t_j \in \{t_j^1, t_j^2, \dots t_j^K\}$ where t_j^k is the earliest arrival time at node j when departing at the k^{th} breakpoint, and R_3 ends at timed node (n, t_n) with $t_n \in \{t_n^1, t_n^2, \dots t_n^K\}$ where t_n^k is the earliest arrival time at node n when departing at the k^{th} breakpoint.

If $\omega(P) < W$, the sets $\{t_1^1, t_1^2, \dots t_1^K\}$ and $\{t_n^1, t_n^2, \dots, t_n^K\}$ can be found in $\mathcal{O}(2K(MATP))$ time, where $MATP$ is the complexity of solving the MATP. Since the subpath of P that start with R_1 and ends with R_3 solves the MTTP from node 1 to node n with time horizon $[t_1, t_n]$, it must be one of the K^2 MTTP solutions found by solving a MTTP for each pair $[t_1^\alpha, t_n^\beta]$ where $\alpha, \beta \in \{1, 2, \dots, K\}$. Completing each of the K^2 MTTP solutions and selecting the completed timed path that is feasible for the TDSPP-LW with the least total travel time must return an optimal solution. To see why, if each of the K^2 MTTP have unique solutions, one of the solutions contain a subpath of P which when completed returns P, otherwise, an alternative solution must have the same travel time as P as it solves the MTTP and has at most as much waiting time since it must arrive at node n no later than P. This can all be done in $\mathcal{O}(K^2(MTTP + MATP))$ time, where $MTTP$ is the complexity of solving the MTTP. Thus in this case, an optimal solution can be found in polynomial time.

If $\omega(P) = W$, suppose node j is known, then the sets $\{t_1^1, t_1^2, \dots t_1^K\}$ and $\{t_j^1, t_j^2, \dots, t_j^K\}$ can be found in $\mathcal{O}(2K(MATP))$ time, where $MATP$ is the complexity of solving the MATP. Since the subpath of P that start with R_1 and ends with R_2 solves the MTTP from node 1 to node j with time horizon $[t_1, t_j]$, it must be one of the K^2 MTTP solutions found by solving a MTTP for each pair $[t_1^\alpha, t_j^\beta]$ where $\alpha, \beta \in \{1, 2, \ldots, K\}$. Append waiting of length $W - \omega(P')$ at node j to each of the MTTP solutions found and complete the resulting timed paths. Selecting the completed timed path that is feasible for the TDSPP-LW with the least total travel time must return an optimal solution. To see why, if each of the K^2 MTTP have unique solutions, one of the solutions contain a subpath of P after appending waiting (since it is known that $\omega(P) = W$) and completing returns P, otherwise, an alternative solution must have the same travel time and tallied waiting time as the subpath of P as it solves the MTTP the sum of tallied waiting and travel time must be t_j, hence appending waiting also returns an optimal solution. This can be done in $\mathcal{O}(K^2(MTTP + MATP))$, where $MTTP$ is the complexity of solving the MTTP. Since node j is not known beforehand, this process needs to be repeated for each node $j \in \{2, 3, \ldots, n-1\}$ which introduces another factor of $\mathcal{O}(n)$ for a run-time of $\mathcal{O}(nK^2(MTTP + MATP))$. Thus in this case, an optimal solution can also be found in polynomial time. □

It now remains to show that such a P satisfying Property 3 exists, we shall do so by first showing that there exists P satisfying Properties 1 and 2.

Lemma 9. *Any optimal timed path, P^*, can be converted into an optimal timed path, P, for which Property 1 holds.*

Proof. Let S_1 be the last non-trivial travel subpath of P^*, and suppose P^* contains at least one travel subpath $S_2 \neq S_1$ that does not contain a breakpoint. We claim that it is possible to apply Δ-shifting such that (i) tallied waiting does not increase, (ii) travel time does not change and (iii) after Δ-shifting, the travel subpath $S_2(\Delta)$ contains a breakpoint or merges with another travel subpath. It is clear that after such a Δ-shifting, the resulting timed path is still feasible by (i), optimal by (ii) and contains one less travel subpath without a breakpoint by (iii), thus, by applying Δ-shifting repeatedly there exists an optimal timed path, P', such that all travel subpaths except possibly the last travel subpath contain a breakpoint.

Let P' be an optimal timed path such that all travel subpaths except possibly the last travel subpath contain a breakpoint. If $\omega(P') = W$, then set $P \leftarrow P'$ and we are done. Otherwise, if $\omega(P') < W$, consider Δ-shifting the last non-trivial travel subpath of P', S_3. The gradient of the function $\tau(S_3(\Delta))$ must be zero, otherwise there exists a Δ such that Δ-shifting S_3 reduces travel time while remaining feasible, since both positive and negatives values of Δ are allowed due to waiting on either side of S_3 as it does not contain a (n, T) which is a breakpoint. Now, perform Δ-shifting on S_3, selecting a value of Δ such that the resulting travel subpath must either (1) contain a breakpoint, in which case we are done, or (2) merges with another travel subpath which must contain a breakpoint since all other travel subpaths contain a breakpoint, in which case we are done or (3) the resulting optimal timed path P satisfies $\omega(P) = W$, in which case we are also done. Hence, in any case, the desired optimal timed path, P, exists.

We now prove the earlier claim. By Observation 2, there exists a range for Δ such that $\tau(S_k(\Delta))$ is an affine function of Δ in that range. Let m_k be the gradient of $\tau(S_k(\Delta))$. Note that if $m_2 = 0$, there exists a value of Δ such that Δ-shifting travel subpath S_2 by itself satisfies all of (i), (ii) and (iii), and thus the claim holds.

Assume now that $m_2 \neq 0$ and simultaneously shift S_1 by Δ and S_2 by $-\frac{m_1}{m_2}\Delta$. Note that the value $-\frac{m_1}{m_2}\Delta$ has been carefully chosen such that (ii) holds. Selecting $\Delta < 0$ and simultaneously shifting can only decrease the value of $\omega(P^*)$. To see why, consider the interval from the start of S_2 to the end of S_1, since travel time does not change the waiting in the interval does not change, waiting at node n may be introduced, but is not tallied, therefore, tallied waiting time can only decrease so (i) holds. If selecting a value of Δ such that either a travel subpath merges or contains a breakpoint, results in the travel subpath $S_2(-\frac{m_1}{m_2}\Delta)$ containing a breakpoint or merging with an existing travel subpath that contains a breakpoint then (iii) holds and we are done. Otherwise, if the value of Δ results in $S_1(\Delta)$ containing a breakpoint or merging with an existing travel subpath, repeat the process, replacing S_1 with the merged travel subpath or simply updating the value of m_1 and the value of Δ until (iii) holds, which can occur, at most, a number of times equal to the sum of all breakpoints and the number of travel subpaths of P^*. In the worst case, the travel subpath containing a Δ-shifting of S_1 merges with the travel subpath containing a Δ-shifting of S_2 and we have $S_1 = S_2$ and (iii) holds. □

Lemma 10. *Any optimal timed path, P^*, for which Property 1 holds, can be converted into an optimal timed path, P, for which Property 1 and 2 hold.*

Proof. Suppose P^* contains at least one travel subpath S that does not consist of a latest departure time path to a timed node concatenated with an earliest arrival time path from the same timed node, where the timed node is selected to be a breakpoint if one exists in the travel subpath.

Select a breakpoint in S if one exists, otherwise, select any timed node in S. Replace S with S' in P, with S' being a latest departure time path from the first (untimed) node of S to the selected timed node concatenated with an earliest arrival time path from the same timed node to the last (untimed) node of S. The resulting timed path P' must satisfy Property 1, since S' contains a breakpoint if S contains a breakpoint. By construction, it must be that $\tau(S') < \tau(S)$, however, P' must not be feasible otherwise that would contradict the optimality of P^*, so it must be the case that $\omega(P') > W \geq \omega(P)$. Moreover, the decrease in travel time must be equal to the increase in waiting time, and since waiting at node n, which is not tallied, can only increase via this procedure, it must be the case that $\tau(P) - \tau(P') \geq \omega(P') - \omega(P)$.

Let R be the last non-trivial travel subpath of P' and apply Δ-shifting to R with $\Delta < 0$. Select Δ with $|\Delta|$ as small as possible such that either (i) the resulting timed path, P'', satisfies $\omega(P'') = W$ or (ii) $R(\Delta)$ contains a breakpoint or (iii) $R(\Delta)$ merges with an existing travel subpath. In cases (ii) and (iii), repeat the procedure until case (i) holds, which can occur, at most, a number of times equal to the sum of all breakpoints and the number of travel subpaths of P'. Note that in each iteration, the increase in travel time must be equal to the decrease in waiting time, and since waiting at node n, which is not tallied, can only increase via this procedure, it must be the case that $\tau(P'') - \tau(P') \leq \omega(P') - \omega(P'')$. Since in the last iteration, P'' is feasible and $\omega(P'') \leq \omega(P') + \tau(P') - \tau(P'') \leq \omega(P) + \tau(P) - \tau(P'')$, we have $\tau(P'') \leq \tau(P) + \omega(P) - W \leq \tau(P)$, hence by setting $P \leftarrow P''$, we return an optimal timed path such that Property 2 hold. Property 1 holds since every travel subpath except the last non-trivial travel subpath contains a breakpoint and $\omega(P) = W$. □

Lemma 11. *Any optimal timed path, P^*, for which Property 1 and 2 hold, can be converted into an optimal timed path, P, for which Property 3 holds.*

Proof. Let S_1 and S_3 be the first and last non-trivial travel subpaths of P^* respectively, and suppose S_1 starts at timed node $(1, t_1)$, S_3 ends at timed node (n, t_n). Note that due to Property 1 and Property 2, it must be the case that $t_1 \in \{t_1^1, t_1^2, \ldots t_1^K\}$ where t_1^k is the latest departure time from node 1 for arrival at the k^{th} breakpoint (given an arbitrary ordering of the breakpoints) and $t_n \in \{t_n^1, t_n^2, \dots t_n^K\}$ where t_n^k is the earliest arrival time at node n when departing at the k^{th} breakpoint.

Select an optimal solution to the MTTP from node 1 to node n with time horizon $[t_1, t_n]$ such that the solution satisfies Property 1 and Property 2. (The algorithm of He et al. (2019) for solving MTTP produces such a solution.) It must be the case that for any travel subpath of the solution that ends at a node (j, t_j), we have that $t_j \in \{t_j^1, t_j^2, \dots t_j^K\}$, where t_j^k is the earliest arrival time at node j when departing at the k^{th} breakpoint. Complete the solution to the MTTP to obtain a timed path P', which must have $\tau(P') \leq \tau(P)$, since the subpath of P starting with S_1 and ending with S_3 is feasible for the MTTP.

Let R_1 and R_3 be the first and last non-trivial travel subpaths of P' respectively, and suppose R_1 starts at timed node $(1, t_1')$ and R_3 ends at timed node (n, t_n').

If $\omega(P') < W$ then we are done since the subpath of P' that starts with R_1 and ends with R_3 solves the MTTP from node 1 to node n with time horizon $[t_1', t_n']$ since it solves the MTTP from node 1 to node n with larger time horizon $[t_1, t_n] \supseteq [t_1', t_n']$. If P' contains only one non-trivial travel subpath R, then since R is a feasible timed path for the MATP from node 1 to node n starting at t_1' is must also be optimal for that MATP.

Otherwise if $\omega(P') \geq W$, apply Δ-shifting to R_3 with $\Delta \leq 0$. Select Δ with $|\Delta|$ as small as possible such that either (i) the resulting timed path, P'', satisfies $\omega(P'') = W$ or (ii) $R_3(\Delta)$ contains a breakpoint or (iii) $R_3(\Delta)$ merges with an existing travel subpath. In cases (ii) and (iii), repeat the procedure until case (i) holds, which can occur, at most, a number of times equal to the sum of all breakpoints and the number of travel subpaths of P'. Note that in each iteration, the increase in travel time must be equal to the decrease in waiting time, and since waiting at node n which is not tallied can only increase via this procedure, it must be the case that $\tau(P'') - \tau(P') \leq \omega(P') - \omega(P'')$. Since in the last iteration, P'' is feasible and $\omega(P'') \leq \omega(P') + \tau(P') - \tau(P'') \leq \omega(P) + \tau(P) - \tau(P'')$ so $\tau(P'') \leq \tau(P) + \omega(P) - W \leq \tau(P)$, hence by setting $P \leftarrow P''$, we return an optimal timed path for the TDSPP-LW. To see why Property 3 holds for P, note that any subpath of P' that starts at a timed node (i, t_i) and ends at a timed node (j, t_j) solves the MTTP from node i to node j with time horizon $[t_i, t_j]$, otherwise replacing that subpath with the MTTP solution provides a timed path with better objective value than P' which is a contradiction. Thus, if R_2 is the second-to-last travel subpath of P and ends at timed node (j, t_j), then the subpath of P (and also of P') that starts with R_1 and ends with R_2 solves an MTTP from node 1 to node j with time horizon $[t_1, t_j]$. If P' contains only one non-trivial travel subpath R that starts at timed node $(1, W)$ (since this is the only node where tallied waiting can occur and $\omega(P) = W$) and ends at timed node (n, t) then R must solve an MTTP from node 1 to node n with time horizon $[W, t]$ and since R is a feasible timed path for the MATP from node 1 to node n starting at W it must also be optimal for that MATP. □

Theorem 9. *The variant of TDSPP-LW in which waiting is constrained at intermediate nodes $M = N \setminus \{n\}$ and the limit on total waiting is positive ($W > 0$) is solvable in polynomial time.*

Proof. If an optimal timed path exists, consecutively applying Lemma 9, 10, and 11 guarantees the existence of an optimal timed path, P, that satisfies Property 3. Lemma 8 specifies how an optimal solution can be found in polynomial time given the existence of such a timed path, P. □

We have recently been made aware of an alternative proof of Theorem 9, independently derived, provided in Omer and Poss (2019a).

7 Summary

We summarize the known complexity results for variants of TDSPP-PW and TDSPP-LW in Table 3.

Table 3: Complexity results for variants of TDSPP-PW and TDSPP-LW. PT stands for polynomial time and NPH stands for NP-Hard. The results in bold are the additions to Table 2 derived in Sections 2-3. Equivalence of the $N \setminus M = \{n\}$ and $N \setminus M = \{1\}$ cases in the third row follow from Corollary 1.

	TDSPP-PW		TDSPP-LW	
	$0 < \alpha \leq 1$	$\alpha > 1$	$W = 0$	$W > 0$
$M = N$	PT (Prop. 2)	**NPH (Thm. 2)**	**NPH (Thm. 3)**	**NPH (Thm. 3)**
$N \setminus M = \{1, n\}$	**PT (Thm. 6)**	PT (Prop. 3)	PT (MDP)	**PT (Thm. 9)**
$N \setminus M = \{n\}$ or $\{1\}$	**PT (Thm. 6)**	PT (Prop. 4)	PT (MATP)	**PT (Thm. 9)**
$M \subset N$	**PT (Thm. 6)**	**PT (Thm. 7)**	**PT (Thm. 8)**	**NPH (Thm. 3)**

8 Final Remarks

In this paper, we have determined the complexity of some variants of time-dependent shortest path problems. Below, we provide our perspective on the practical value of these results.

The type of time-dependent shortest path problem we have studied arises in situations where there is a trade-off between waiting and traveling. This may occur, for example, in the transport of perishable goods, when product decay happens faster when traveling. To minimize decay from product origin to destination, given time-dependent travel times, it may be advantageous to wait at intermediate locations. However, a limit on total waiting may be imposed. Another example occurs when planning a sightseeing itinerary. Tourists prefer exploring sights over traveling, so minimizing total travel time is a primary objective. The minimum time allotted for exploring a sight can be included in the (time-dependent) travel times between sights, and a constraint on waiting at less desirable sights can be imposed to allow for more time at the more desirable sights.

Another potential application lies in the case where the network is linear, i.e., when the route is known and only the travel time needs to be evaluated, which occurs frequently as a subproblem in vehicle routing problems. Being able to solve TDSPP-LW and TDSPP-PW introduces greater flexibility in the parent problem and allows more difficult variants to be solved.

Note also that the "travel time" in this paper is not restricted to modeling just travel time, but can also include other time-dependent properties, such as service time, processing time, and mandatory waiting time (that is not tallied).

Although the time-complexity of several polynomial-time variants may be discouraging at face value, the factors relating to n and K stem from having to solve many subproblems (MTTP and MATP) that are very similar. It is possible that exploiting this similarity computationally may result in practically viable algorithms. Exploring such ideas is left for future research.

Acknowledgment

This material is based upon work supported by the National Science Foundation under Grant No. 1662848.

References

D. Bertsimas, A. Delarue, P. Jaillet, and S. Martin. Travel time estimation in the age of big data. *Operations Research*, 2019.

I. Chabini. Discrete dynamic shortest path problems in transportation applications: Complexity and algorithms with optimal run time. *Transportation Research Record: Journal of the Transportation Research Board*, (1645):170–175, 1998.

B. Coifman. Estimating travel times and vehicle trajectories on freeways using dual loop detectors. *Transportation Research Part A: Policy and Practice*, 36(4):351–364, 2002.

K. L. Cooke and E. Halsey. The shortest route through a network with time-dependent internodal transit times. *Journal of mathematical analysis and applications*, 14(3):493–498, 1966.

B. S. Cooper and R. V. Cowlagi. Path-planning with waiting in spatiotemporally-varying threat fields. *PloS one*, 13(8):e0202145, 2018.

B. C. Dean. Algorithms for minimum-cost paths in time-dependent networks with waiting policies. *Networks: An International Journal*, 44(1):41–46, 2004a.

B. C. Dean. Shortest paths in fifo time-dependent networks: Theory and algorithms. *Rapport technique, Massachusetts Institute of Technology*, 2004b.

B. Ding, J. X. Yu, and L. Qin. Finding time-dependent shortest paths over large graphs. In *Proceedings of the 11th international conference on Extending database technology: Advances in database technology*, pages 205–216. ACM, 2008.

M. A. Figliozzi. The time dependent vehicle routing problem with time windows: Benchmark problems, an efficient solution algorithm, and solution characteristics. *Transportation Research Part E: Logistics and Transportation Review*, 48(3):616–636, 2012.

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

A. Franceschetti, D. Honhon, G. Laporte, and T. Van Woensel. A shortest-path algorithm for the departure time and speed optimization problem. *Transportation Science*, 52(4):756–768, 2018.

E. He, N. Boland, G. Nemhauser, and M. Savelsbergh. Computational complexity of time-dependent shortest path problems. *Optimization Online*, pages Eprint ID 2019–02–7083, 2019.

S. Ichoua, M. Gendreau, and J.-Y. Potvin. Vehicle dispatching with time-dependent travel times. *European journal of operational research*, 144(2):379–396, 2003.

E. Kanoulas, Y. Du, T. Xia, and D. Zhang. Finding fastest paths on a road network with speed patterns. In *Data Engineering, 2006. ICDE'06. Proceedings of the 22nd International Conference on*, pages 10–10. IEEE, 2006.

J. Letchner, J. Krumm, and E. Horvitz. Trip router with individualized preferences (trip): Incorporating personalization into route planning. In *Proceedings of the National Conference on Artificial Intelligence*, volume 21, page 1795. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2006.

K. Nachtigall. Time depending shortest-path problems with applications to railway networks. *European Journal of Operational Research*, 83(1):154–166, 1995.

M. Nykänen and E. Ukkonen. The exact path length problem. *Journal of Algorithms*, 42(1):41–53, 2002.

J. Omer and M. Poss. A polynomial algorithm for minimizing travel time in time-dependent networks with waits. working paper or preprint, 2019a. URL `https://hal.archives-ouvertes.fr/hal-01836007`.

J. Omer and M. Poss. Time-dependent shortest paths with discounted waits. *Networks*, 2019b.

A. Orda and R. Rom. Shortest-path and minimum-delay algorithms in networks with time-dependent edge-length. *Journal of the ACM (JACM)*, 37(3):607–625, 1990.

Y. Wang and N. L. Nihan. Freeway traffic speed estimation with single-loop outputs. *Transportation Research Record*, 1727(1):120–126, 2000.