A polynomial algorithm for minimizing travel time in time-dependent networks with waits

Jérémy Omer, Michael Poss

A polynomial algorithm for minimizing travel time in time-dependent networks with waits

Jérémy Omer*

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

Michael Poss†

LIRMM, University of Montpellier, CNRS, Montpellier, France

July 10, 2019

Abstract

We consider a time-dependent shortest path problem with possible waiting at each node and a global bound W on the total waiting time. The goal is to minimize only the time traveled along the edges of the path, not including the waiting time. We prove that the problem can be solved in polynomial time when the travel time functions are piecewise linear and continuous. The algorithm relies on a recurrence relation characterized by a bound ω for the total waiting time, where $0 \leq \omega \leq W$. We show that only a small numbers of values $\omega_1, \omega_2, \ldots, \omega_K$ need to be considered, which depends on the total number of breakpoints of all travel time functions.
keywords: Time-dependent networks; Shortest paths; Label-setting algorithms; Concavity; Breakpoint; Wait.

1 Introduction

We consider in this paper a variant of the time-dependent shortest path problem (TDSP). Given a graph $G = (V, A)$ with n nodes and m arcs, the objective is to find the path from $o \in V$ to $d \in V$ that has the lowest travel time. It is permitted to wait at the nodes, as long as the total waiting time does not exceed a given bound W, and the waiting time is not considered in the travel time. The problem has diverse applications in transport and logistics such as planning routes for truck drivers while minimizing the fuel consumption.

The study of the TDSP (without waits) dates back to Cooke and Halsey [6] who introduced an extension of Bellman's equations [3] to the time-dependent context. This early work has been followed by articles introducing an assumption, often referred to as the *consistency assumption*, preventing from reaching the head of an arc earlier by departing later from its tail. Also called FIFO (first-in first-out), the assumption leads to polynomial-time algorithms for the problem [8, 11].

Allowing to wait at nodes makes the problem more complex since one must decide how much to wait at each node, in addition to choosing the path. A general approach to the TDSP with

*jeremy.omer@insa-rennes.fr
†michael.poss@lirmm.fr

waiting has been proposed by Cai et al. [4] who consider that each arc has a time-dependent cost, in addition to the time-dependent traversing time, and that waiting incurs a cost, also modeled by a time-dependent function. Their model also requires that the path reaches the destination within a given time horizon. Different dynamic programming algorithms have been proposed for this variant [4, 5, 7], resulting algorithms running in time polynomial in n, m, and the length of the time horizon, making them pseudo-polynomial. Related works consider alternative settings such as minimizing the excess time (over the minimum travel time) [2] or road networks with traffic lights [1].

Other variants of the TDSP with waiting have been considered by [9] and [12]. Both papers consider only traversing time functions and relax the time horizon constraint. On the one hand, if waiting is allowed only at the source node, Foschini et al. [9] show that the problem remains polynomially solvable. On the other hand, if waiting is allowed at each node, and bounds not greater than W are considered on each node, the problem becomes $\mathcal{NP}$-hard [12] and can be solved polynomially in n, m and W. Interestingly, the reduction provided in [12] does not extend to the case where the waiting times are bounded only by one global constraint. The purpose of this paper is to fill this gap: we prove that the TDSP allowing waiting at each node and bounding the waiting time only by one global constraint can be solved in polynomial time.

We now provide a rough description of the algorithm proposed in this article. Similarly to the method developed by Foschini et al. [9], our algorithm exploits the breakpoints of the travel time functions. Yet, our problem needs to handle the possibility of waiting at every node of the graph, unlike in [9] where it is allowed to wait only at the source node. Hence, even when the path is given, the optimization problem considered herein involves n optimization variables, one for each node, while the counterpart from [9] is a one-dimensional optimization problem. We address this difficulty by introducing $T_v(\omega)$, the travel-time of the cheapest path to node v as a function of the bound $\omega \leq W$ on the total waiting time. What is more, we are able to construct a sequence of possible waits $\omega_1 = 0, \omega_2, \ldots, \omega_K = W$ such that $\{T_v(\omega_{k+1}), v \in V\}$ can be computed from $\{T_v(\omega_k), v \in V\}$ by applying a variant of Dijkstra's algorithm. We show that the values $\omega_1, \ldots, \omega_K$ are the possible non-concavities of the function $T_v(\omega)$ and every such non-concavity can be traced back to a distinct breakpoint of the transit times functions. Hence, K is bounded by the total number of breakpoints. The overall algorithm is polynomial in n, m, and the number of breakpoints of the travel time functions.

The rest of the paper is structured as follows. In the next section, we define the problem formally. Section 3 introduces basic properties of the problem. Section 4 analyzes the structure of optimal paths and introduces a key property that must be satisfied by the sequence $\omega_1, \omega_2, \ldots, \omega_K$. Section 5 proposes an algorithm building a sequence satisfying this property.

2 Problem definition

The travel time of each arc is given by a non-negative continuous piecewise linear function $C_e : \mathbb{R}^+ \to \mathbb{R}^+$ defined by r_e pieces. Each piece $s = 1, \ldots, r_e$ is an affine function $c_e^s + \rho_e^s t$ defined on the interval $[\tau_e^{s-1}, \tau_e^s]$, where $c_e^s, \rho_e^s \in \mathbb{R}$, $\tau_e^s \in \mathbb{R}_+$, $\tau_e^0 = 0$ and $\tau_e^{r_e} = +\infty$.

Let $\mathcal{P}_v$ be the set of all paths from o to $v \in V$. We further define $\mathcal{PW}_v(\omega)$ as the sets of paths-with-waits $\pi = (p, \mathrm{w})$, where $p = (v_1(= o), v_2, \ldots, v_{|p|}(= v))$ belongs to $\mathcal{P}_v$, and $\mathrm{w} = (\mathrm{w}_1, \ldots, \mathrm{w}_{|p|-1})$ is the vector of waiting times, which satisfies $\sum_{i=1}^{|p|-1} \mathrm{w}_i \leq \omega$. Since we wish to minimize total travel

time, the cost of (p, w) can be expressed as

$$C(\pi) = \sum_{i=1}^{|p|-1} C_i(t_i), \tag{1}$$

where C_i is a shorthand notation for $C_{v_i v_{i+1}}$ and t_i is the departure time from node v_i, which can be computed using the recursion

$$t_i = t_{i-1} + C_{i-1}(t_{i-1}) + \mathrm{w}_i. \tag{2}$$

Let $T_v(\omega) = \min_{\pi \in \mathcal{PW}_v(\omega)} C(\pi)$ be the minimum travel time among all paths from o to v with total waiting time at most ω. We note that the value of $T_v(\omega)$ is different from the departure time at node v, t_v, which includes the total waiting time up to node v (see (2)). For $\pi^* = (p^*, \mathrm{w}^*) \in \arg\min_{\pi \in \mathcal{PW}_v(\omega)} C(\pi)$, the two quantities are related through the equation $T_v(\omega) = t_v - \sum_{i=1}^{|p^*|} \mathrm{w}_i^*$, or equivalently,

$$C(\pi^*) = t_v - \sum_{i=1}^{|p^*|} \mathrm{w}_i^*. \tag{3}$$

The aim of the article is to provide a polynomial-time algorithm for the optimization problem.

$$T_d(W) = \min_{\pi \in \mathcal{PW}_d(W)} C(\pi). \tag{TDSPW}$$

In most applications, we can assume that entering an arc e at time $t' \geq t$ leads to leaving the arc at time $t' + \mathcal{C}_e(t') \geq t + \mathcal{C}_e(t)$, which can be equivalently stated as follows.

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

The purpose of the article is to prove the following theorem.

Theorem 1. *Problem (TDSPW) can be solved in polynomial time with respect to n, m and $\max_{e \in E} r_e$.*

In what follows, we shall refer to the left and right derivatives of a one-variable function $f(x)$ as $\partial_- f(x)$ and $\partial_+ f(x)$.

3 Preliminaries

Let $v \in V$ and $\omega \leq W$, our algorithm is based on the recurrence relation that relates $T_v(\omega)$ to the minimum travel time to the predecessors of v. To derive the relation, we introduce $\widetilde{T}_v(\omega)$ as the minimum travel time among all paths from o to v with total waiting time *exactly* ω, not including the time waited at v. Formally,

$$\widetilde{T}_v(\omega) = \min_{\pi := (p,\mathrm{w}) \in \mathcal{PW}_v(\omega)} \left\{ C(\pi) : \sum_{i=1}^{|p|-1} \mathrm{w}_i = \omega \right\}.$$

For this quantity, one can verify that the following recurrence holds.

$$\widetilde{T}_v(\omega) = \min_{(u,v) \in E} \left\{ \min_{\omega' \leq \omega} \widetilde{T}_u(\omega') + C_{uv}(\widetilde{T}_u(\omega') + \omega) \right\}.$$

Using the consistency assumption, we observe that $\widetilde{T}_u(\omega') + C_{uv}(\widetilde{T}_u(\omega') + \omega)$ is minimum if $\widetilde{T}_u(\omega')$ is minimum. Since $T_u(\omega) = \min_{\omega' \leq \omega} \widetilde{T}_u(\omega')$, the above recurrence relation can be equivalently written as

$$\widetilde{T}_v(\omega) = \min_{(u,v)\in E} T_u(\omega) + C_{uv}(T_u(\omega) + \omega). \tag{4}$$

As a consequence, we can distinguish two cases in the computation of $T_v(\omega)$, depending on whether it is achieved with a total wait equal to or lower than ω. If there exists $\omega' < \omega$ such that $T_v(\omega) = \widetilde{T}_v(\omega')$, then

$$T_v(\omega) = T_v(\omega'). \tag{5}$$

Otherwise, $T_v(\omega) = \widetilde{T}_v(\omega)$, so we can directly rewrite (4) as

$$T_v(\omega) = \min_{(u,v)\in E} T_u(\omega) + C_{uv}(T_u(\omega) + \omega). \tag{6}$$

Grouping relations (5) and (6), $T_v(\omega)$ satisfies the recurrence relation

$$T_v(\omega) = \min\left\{ \inf_{0\leq\omega'<\omega} T_v(\omega'), \min_{(u,v)\in E} T_u(\omega) + C_{uv}(T_u(\omega) + \omega) \right\}. \tag{7}$$

The bottom line of our algorithm is to build a sequence with polynomial length $0 = \omega_1, \omega_2, \ldots, \omega_K = W$ such that relation (7) can be simplified to

$$T_v(\omega_{k+1}) = \min\left\{ T_v(\omega_k), \min_{(u,v)\in E} T_u(\omega_{k+1}) + C_{uv}(T_u(\omega_{k+1}) + \omega_{k+1}) \right\}, \tag{8}$$

which can be equivalently formulated as

$$T_v(\omega_{k+1}) = \min\left\{ T_v(\omega_k), \widetilde{T}_v(\omega_{k+1}) \right\}. \tag{9}$$

Indeed, for such sequence, $\{T_u(\omega_{k+1})\}_{u\in V}$ can be computed in polynomial time from $\{T_u(\omega_k)\}_{u\in V}$ using the variant of Dijkstra's algorithm presented in Algorithm 1.

Proposition 1. *Let $\omega_k, \omega_{k+1} \in [0, W]$ such that $\forall v \in V$, ω_{k+1} satisfies (8). Then, the label setting algorithm defined by Algorithm 1 returns $T_v(\omega_{k+1}), \forall v \in V$, in at most $\mathcal{O}(n\log(n) + m)$ operations.*

Proof. The algorithm is an adaptation of Dijkstra's algorithm for shortest paths and the proof is similar. The time complexity of the algorithm is the same as that of Dijkstra's algorithm, and using a Fibonacci heap [10] yields the desired complexity. □

We conclude the section by introducing two useful properties. First, we show that the consistency assumption implies that the right-derivative of $T(\omega)$ is bounded below by -1.

Proposition 2. *$\partial_+ T_v(\omega) \geq -1$ for all $\omega \in [0, W[$.*

Proof. Consider $\tilde{\omega} > 0$ and let $\tilde{\pi} = (\tilde{p}, \tilde{\mathrm{w}}) \in \mathcal{PW}_v(\tilde{\omega})$ be a path-with-waits from o to v that achieves $T_v(\tilde{\omega})$, so $C(\tilde{p}, \tilde{\mathrm{w}}) = T_v(\tilde{\omega})$. Let v_j be the last node along $\tilde{p}$ for which $\tilde{\mathrm{w}}_j > 0$ and define $\mathrm{w} \in \mathbb{R}^n$ as $\mathrm{w}_i = \tilde{\mathrm{w}}_i$ for $i \neq j$ and $\mathrm{w}_j = \tilde{\mathrm{w}}_j - \delta$ for some $0 < \delta \leq \tilde{\mathrm{w}}_j$, as well as $\omega = \sum_{i=1}^{|\tilde{p}|-1} \mathrm{w}_i = \tilde{\omega} - \delta$. We claim that

$$\frac{C(\tilde{p}, \tilde{\mathrm{w}}) - C(\tilde{p}, \mathrm{w})}{\delta} \geq -1. \tag{10}$$

input : $\omega_k, \omega_{k+1} \in [0, W], \omega_k < \omega_{k+1}$
$T_v(\omega_k), \forall v \in V$

$T_v \leftarrow T_v(\omega_k), \forall v \in V$
$S \leftarrow \{o\}$ // set of marked vertices
$u \leftarrow o$ // last marked vertex
while $S \neq V$ **do**
  for $(u, v) \in E$ **do** $T_v \leftarrow \min\{T_v; T_u + C_{uv}(T_v + \omega_{k+1})\}$ // update minimum travel times
  $u \leftarrow \arg\min_{v \in V \setminus S}\{T_v\}$ // mark the vertex with minimum travel time
  $S \leftarrow S \cup \{u\}$
output: $T_v, v \in V$

Algorithm 1: Adaptation of Dijkstra's label setting algorithm for the computation of $T(\omega_{k+1})$ provided $T(\omega_k)$

Therefore,

$$\frac{T_v(\omega+\delta) - T_v(\omega)}{\delta} = \frac{C(\tilde{p}, \tilde{\mathrm{w}}) - \min\limits_{\pi \in \mathcal{PW}_v(\omega)} C(\pi)}{\delta} \geq \frac{C(\tilde{p}, \tilde{\mathrm{w}}) - C(\tilde{p}, \mathrm{w})}{\delta} \geq -1.$$

and the results follows by taking the limit.

Let us now prove the claim (10), defining $\tilde{t}$ and t as the vectors of departure times for $\tilde{\mathrm{w}}$ and w, respectively. By definition, $t_j = \tilde{t}_j - \delta$ and, since $\rho_{min} \geq -1$, $t_\ell \leq \tilde{t}_\ell$ for any $j \leq l \leq |\tilde{p}|$. In particular, $t_{|\tilde{p}|} \leq \tilde{t}_{|\tilde{p}|}$, and the claim follows from (3).

□

Next we show that we can restrict our attention to acyclic solutions.

Proposition 3. *TDSPW admits an acyclic optimal solution.*

Proof. Consider a solution (p, w) where $p = (v_1, \ldots, v_{|p|})$ contains a cycle c. Then, $v_{i_1} = v_{i_2}$ for some $i_1 < i_2$. Denote $\sigma_j = \sum_{i=1}^{j} \mathrm{w}_i, \forall j \in \{2, \ldots, |p|\}$. Assumption 1 ensures that $\partial_+ C_{uv} \geq -1, \forall (u, v) \in E$. As a consequence,

$$T_{v_{i_1}}(\sigma_{i_1}) + C_{i_2}(T_{v_i}(\sigma_{i_1}) + \sigma_{i_2}) \leq T_{v_{i_1}}(\sigma_{i_1}) + C(c) + C_{i_2}(T_{v_i}(\sigma_{i_1}) + \sigma_{i_2} + C(c)) = T_{v_{i_2+1}}(\sigma_{i_2})$$

Stated otherwise, the cycle from v_{i_1} to v_{i_2} can be removed without increasing the total travel time nor the total waiting by waiting an additional $\sigma_{i_2} - \sigma_{i_1}$ at $v_{i_1} = v_{i_2}$.

□

4 Structure of optimal paths

The sequence $\omega_1, \ldots, \omega_K$ will be constructed so that for all $k = 1, \ldots, K$ there is some arc (u, v) such that a breakpoint of C_{uv} is reached at $T_u(\omega_k) + \omega_k$. The practical algorithm that identifies this sequence is based on the structure of the paths-with-waits that realize the minimum travel time $T_v(\omega)$. For this, if $\pi = (p, \mathrm{w}) \in \mathcal{PW}_v(\omega)$, we define $\pi(i) = (p(i), \mathrm{w}(i))$ and $\overline{\pi}(i) = (\overline{p}(i), \overline{\mathrm{w}}(i))$ as the subpaths-with-waits of π respectively ending and starting at the i-th node of p.

Definition 1. *Let $v \in V$, $0 \leq \omega \leq W$ and $\pi := (p, \mathrm{w}) \in \mathcal{PW}_v(\omega)$ where $p = (v_1 := o, v_2, \ldots, v_{|p|})$ and $\mathrm{w} = (\mathrm{w}_1, \ldots, \mathrm{w}_{|p|})$. If there is $i \leq |p| - 1$ such that $\sum_{k=1}^{i} \mathrm{w}_k = \omega$ and $\sum_{k=1}^{i-1} \mathrm{w}_k = \omega' < \omega$, then $\overline{\pi}(i)$ is the* saturated subpath of π. *By extension the nodes $v_i, \ldots, v_{|p|} = v$ are said to be* saturated *in π and v_i is the* first saturated node *in π.*

From this definition, one can notice that the first saturated node of a path-with-waits is also the last one where the wait is positive. The introduction of this saturated subpath allows for a more precise characterization of optimal paths.

Proposition 4. *Let $v \in V$, $0 \leq \omega \leq W$ such that $T_v(\omega) = \widetilde{T}_v(\omega)$. Then, there exists $\pi := (p, \mathrm{w}) \in \mathcal{PW}_v(\omega)$ such that*

- *$C(\pi) = T_v(\omega)$, $\sum_{k=1}^{|p|-1} \mathrm{w}_k = \omega$, and*
- *$C(\pi(j)) = T_{v_j}(\omega)$ for every saturated node v_j in π.*

In particular, $T_{v_j}(\omega) = \widetilde{T}_{v_j}(\omega)$ for every saturated node v_j after the first saturated node.

Proof. Let $\pi := (p, \mathrm{w}) \in \mathcal{PW}_v(\omega)$ such that $C(\pi) = T_v(\omega)$ and $\sum_{k=1}^{|p|-1} \mathrm{w}_k = \omega$. Assume that there is v_j, a saturated node in π, such that $C(\pi(j)) > T_{v_j}(\omega)$. There is $\pi^j := (p^j, \mathrm{w}^j) \in \mathcal{PW}_{v_j}(\omega)$ such that $C(\pi^j) = T_{v_j}(\omega)$ and $\sum_{\ell=1}^{|\pi^j|-1} \mathrm{w}^j_\ell = \omega^j \leq \omega$. We can then build a path-with-waits π' from o to v with total wait ω, by appending $\bar{\pi}(j)$ to π^j and waiting an additional $\omega - \omega^j$ at v_j. Node v_j is saturated in both π and π', so the departure time from v_j is $C(\pi(j)) + \omega$ in π and $T_{v_j}(\omega) + \omega$ in π'. And since π and π' are identical from v_j to v, the consistency assumption and $C(\pi(j)) + \omega > T_{v_j}(\omega) + \omega$ yields $C(\pi) \geq C(\pi')$. From $C(\pi) = T_v(\omega)$, we deduce that $C(\pi') = T_v(\omega)$. We thus obtain the desired path-with-waits by applying the above approach recursively on the saturated nodes until $C(\pi(j)) = T_{v_j}(\omega)$ for every saturated node v_j.

The last assertion follows from the fact that the total wait along $\pi(j)$ is exactly equal to ω.

□

Remark 1. *We observe that if i is the first saturated node in π, there is no wait from v_{i+1} to v in π. If $C(\pi) = T_v(\omega)$, this means that $\overline{p}(i)$ is a shortest path (without wait) from v_i to v among those departing from v_i at $T_{v_i}(\omega) + \omega$.*

Remark 2. *In the proof of Proposition 4, we use the consistency assumption to show that $C(\pi(j)) + \omega > T_{v_j}(\omega) + \omega$ implies $C(\pi) \geq C(\pi')$. Actually, it can happen that $C(\pi) = C(\pi')$ only if there is an arc $(v_\ell, v_{\ell+1})$ in π such that $\ell \geq j$ and $C_{\ell,\ell+1}$ has a slope equal to -1 at $C(\pi(\ell)) + \omega$. As a consequence, if there is no such arc in the saturated subpath of π, we can show that $C(\pi) = T_v(\omega)$ directly implies that $C(\pi(j)) = T_{v_j}(\omega)$ for every saturated node v_j.*

Next we prove a property which, if satisfied by sequence $\omega_1, \ldots, \omega_K$, directly implies that (8) holds. The proof of the following result requires a few new notations. Let us denote the index of the right and left-breakpoint of function C_e at $T_u(\omega_k) + \omega_k$ as $s_e(\omega_k)$ and $s_e^-(\omega_k)$, respectively. Formally,

$$s_e(\omega_k) = \min_s \{s : T_u(\omega_k) + \omega_k < \tau_e^s\} \quad \text{and} \quad s_e^-(\omega_k) = s_e(\omega_k) - 1.$$

To keep concise notations, we denote shortly $\tau_e^{s_e(\omega_k)}$ as $\tau_e(\omega_k)$, $\tau_e^{s_e^-(\omega_k)}$ as $\tau_e^-(\omega_k)$, $\rho_e^{s_e(\omega_k)}$ as $\rho_e(\omega_k)$, and $c_e^{s_e(\omega_k)}$ as $c_e(\omega_k)$. We also define $\tau_u(\omega_k) = \min_{(u,v)\in E} \tau_{uv}(\omega_k)$ and $\tau_u^-(\omega_k) = \max_{(u,w)\in E} \tau_{uw}^-(\omega_k)$.

Proposition 5. *Let $\omega_{k+1} \geq \omega_k$ be such that for all $u \in V$,*

$$T_u(\omega_{k+1}) + \omega_{k+1} \leq \tau_u(\omega_k). \tag{11}$$

Then for all $v \in V$, $\omega \mapsto T_v(\omega)$ is a continuous piecewise linear and concave function on $[\omega_k, \omega_{k+1}]$.

Proof. We prove the result by contradiction, so assume that there is $v \in V$ such that $\omega \mapsto T_v(\omega)$ is not concave on $[\omega_k, \omega_{k+1}]$. We have already observed that T_v is a continuous piecewise linear function, so it is not concave if and only if there is $\bar{\omega} \in [\omega_k, \omega_{k+1}]$ such that T_v is not differentiable at $\bar{\omega}$ and

$$\partial_- T_v(\bar{\omega}) < \partial_+ T_v(\bar{\omega}) \tag{12}$$

In particular, (12) involves that $\partial_- T_v(\bar{\omega}) < 0$, so $\widetilde{T}_v(\bar{\omega}) = T_v(\bar{\omega})$. According to Proposition 4, this means that we can build $\pi = (p, \mathrm{w}) \in \mathcal{PW}_v(\bar{\omega})$ such that $C(\pi) = T_v(\bar{\omega})$, $\sum_{\ell=1}^{|p|-1} \mathrm{w}_\ell = \bar{\omega}$, and, denoting $p = (v_1(= o), v_2, \ldots, v_{|p|} = v)$, $C(\pi(j)) = T_{v_j}(\bar{\omega})$ for every saturated node v_j in π. To exhibit the contradiction, we study the function $f : [-\mathrm{w}_i, W - \bar{\omega}] \to \mathbb{R}^+$ defined by $f(\epsilon) = C(p, \mathrm{w} + \epsilon\delta(i))$. where v_i is the first saturated node in π, and $\delta(i)$ is the i-th vector of the canonical basis of $\mathbb{R}^{|p|}$ ($\delta_i(i) = 1$ and $\delta_j(i) = 0$, for $j \neq i$). Stated otherwise, $f(\epsilon)$ is the cost of the path-with-waits from o to v obtained from π by waiting an additional ϵ at node i. This path-with-wait is denoted as $\pi^\epsilon (= (p, w + \epsilon\delta))$. Since the arc cost functions are piecewise linear and continuous, so is f. As a consequence, we can compute the left and right partial derivative of f at 0 as

$$\begin{cases} \partial_- f(0) = \lim\limits_{\epsilon \to 0-} \dfrac{C(p, \mathrm{w} + \epsilon\delta(i)) - C(p, \mathrm{w})}{\epsilon} = \lim\limits_{\epsilon \to 0-} \dfrac{C(p, \mathrm{w} + \epsilon\delta(i)) - T_v(\bar{\omega})}{\epsilon} \\ \partial_+ f(0) = \lim\limits_{\epsilon \to 0+} \dfrac{C(p, \mathrm{w} + \epsilon\delta(i)) - C(p, \mathrm{w})}{\epsilon} = \lim\limits_{\epsilon \to 0+} \dfrac{C(p, \mathrm{w} + \epsilon\delta(i)) - T_v(\bar{\omega})}{\epsilon} \end{cases}$$

Observing that $(p, w + \epsilon\delta(i))$ is a path-with-waits from o to v with total wait $\bar{\omega} + \epsilon$, we get $C(p, \mathrm{w} + \epsilon\delta(i)) \geq T_v(\bar{\omega} + \epsilon)$ which yields

$$\begin{cases} \partial_- f(0) \leq \lim\limits_{\epsilon \to 0-} \dfrac{T_v(\bar{\omega} + \epsilon) - T_v(\bar{\omega})}{\epsilon} = \partial_- T_v(\bar{\omega}) \\ \partial_+ f(0) \geq \lim\limits_{\epsilon \to 0+} \dfrac{T_v(\bar{\omega} + \epsilon) - T_v(\bar{\omega})}{\epsilon} = \partial_+ T_v(\bar{\omega}) \end{cases}$$

Together with (12), we get

$$\partial_- f(0) \leq \partial_- T_v(\bar{\omega}) < \partial_+ T_v(\bar{\omega}) \leq \partial_+ f(0),$$

which implies in particular that f is non-differentiable at 0.

By definition, $f(\epsilon)$ is the sum of composites of the piecewise-linear arc cost functions $C_j, j = 1, \ldots, |p| - 1$. For $j = 1, \ldots, i-1$ the term related to C_j is taken at $C(\pi(j)) + \sum_{k=1}^{j-1} \mathrm{w}_k$, which does not depend on ϵ. For $j = i, \ldots, [p| - 1$ the term related to C_j is taken at $C(\pi^\epsilon(j)) + \bar{\omega} + \epsilon$. Let v_j be a saturated node ($j \in \{i, \ldots, |p|\}$: using that $C(\pi(j)) = T_{v_j}(\bar{\omega})$ and $\omega_k \leq \bar{\omega} \leq \omega_{k+1}$, we get $\tau^-_{v_j}(\omega_k) \leq C(\pi(j)) + \bar{\omega} \leq \tau_{v_j}(\omega_k)$. Recalling that C_j is differentiable on $]\tau^-_{v_j}(\omega_k), \tau_{v_j}(\omega_k)[$, for all $v, v' \in V$, we get that f is non-differentiable at 0 only if there is a saturated node v_j such that the piece where C_j is evaluated changes at 0, i.e.,

i. $T_{v_j}(\bar{\omega}) + \bar{\omega} = \tau^-_{v_j}(\omega_k)$ and $\exists \alpha > 0 : C(\pi^{-\epsilon}(j)) + \bar{\omega} - \epsilon < \tau^-_{v_j}(\omega_k)$ for all $0 < \epsilon < \alpha$; or

ii. $T_{v_j}(\bar{\omega}) + \bar{\omega} = \tau_{v_j}(\omega_k)$ and $\exists \alpha > 0 : C(\pi^{\epsilon}(j)) + \bar{\omega} + \epsilon > \tau_{v_j}(\omega_k)$ for all $0 < \epsilon < \alpha$.

If $T_{v_j}(\bar{\omega}) + \bar{\omega} = \tau^-_{v_j}(\omega_k)$, $\tau^-_{v_j}(\omega_k) \leq T_{v_j}(\omega_k) + \omega_k$ yields $T_{v_j}(\bar{\omega}) + \bar{\omega} = T_{v_j}(\omega_k) + \omega_k$, hence $\partial_- T_{v_j}(\bar{\omega}) = -1$. Function T_{v_j} is piecewise linear, so there is $\alpha > 0$ such that for all $0 < \epsilon < \alpha$, $T_{v_j}(\bar{\omega} - \epsilon) = T_{v_j}(\bar{\omega}) + \epsilon$. Using that $C(\pi^{-\epsilon}(j)) \geq T_{v_j}(\bar{\omega} - \epsilon)$, we obtain $C(\pi^{-\epsilon}(j)) + \bar{\omega} - \epsilon \geq T_{v_j}(\bar{\omega}) + \bar{\omega} \geq \tau^-_{v_j}(\omega_k)$, so item i. is never true.

If $T_{v_j}(\bar{\omega}) + \bar{\omega} = \tau_{v_j}(\omega_k)$, then $T_{v_j}(\bar{\omega}) + \bar{\omega} = T_{v_j}(\omega_{k+1}) + \omega_{k+1}$ by definition of ω_{k+1}. A path-with-waits from o to v_j that realizes $T_{v_j}(\omega_{k+1})$ with total wait ω_{k+1} will then arrive at $T_{v_j}(\omega_{k+1}) + \omega_{k+1} = T_{v_j}(\bar{\omega}) + \bar{\omega}$. So if we continue p from there without additional waits, v will also be reached at $T_v(\bar{\omega}) + \bar{\omega}$, just like in π. Denoting this new path as π', we get that $C(\pi') = T_v(\bar{\omega}) + \bar{\omega} - \omega_{k+1}$. Finally, $C(\pi') \geq T_v(\omega_{k+1})$, hence $T_v(\omega_{k+1}) + \omega_{k+1} \leq T_v(\bar{\omega}) + \bar{\omega}$. This is only possible if $\partial_+ T_v(\omega) = -1$ for all $\omega \in [\bar{\omega}, \omega_{k+1}]$, which is in contradiction with $\partial_- T_v(\bar{\omega}) < \partial_+ T_v(\bar{\omega})$. We conclude that item ii. is never true either, hence f is differentiable at 0: a contradiction.

□

Corollary 1. *Let $\omega_{k+1} \geq \omega_k$ be such that (11) holds. Then for all $v \in V$,*

$$T_v(\omega_{k+1}) = \min\left\{T_v(\omega_k), \widetilde{T}_v(\omega_{k+1})\right\}. \tag{13}$$

Proof. By concavity of T_v on $[\omega_k, \omega_{k+1}]$, T_v is either constant $[\omega_k, \omega_{k+1}]$ or there is $\bar{\omega} \in [\omega_k, \omega_{k+1}]$ such that T_v is decreasing on $[\bar{\omega}, \omega_{k+1}]$. In the latter case, we get that $T_v(\omega) = \widetilde{T}_v(\omega)$ for all $\omega \in [\bar{\omega}, \omega_{k+1}]$. In particular, $T_v(\omega_{k+1}) = \widetilde{T}_v(\omega_{k+1})$. To summarize, $T_v(\omega_{k+1}) = \min\left\{T_v(\omega_k), \widetilde{T}_v(\omega_{k+1})\right\}$.
□

The above result allows to get more specific in the characterization of the first saturated node of the path exhibited in Proposition 4 when $\omega = \omega_{k+1}$ for some $k \in \{1, \ldots, K-1\}$.

Corollary 2. *Let $0 \leq \omega_k < \omega_{k+1} \leq W$ such that (11) holds and consider $v \in V$ such that $T_v(\omega_{k+1}) = \widetilde{T}_v(\omega_{k+1})$. Then, there exists $\pi := (p, \mathrm{w}) \in \mathcal{PW}_v(\omega_{k+1})$ such that*

- *$C(\pi) = T_v(\omega_{k+1})$ and $\sum_{k=1}^{|p|-1} \mathrm{w}_k = \omega_{k+1}$;*
- *$C(\pi(i)) = T_{v_i}(\omega_k)$, where v_i is the first saturated node in π;*
- *and $C(\pi(j)) = \widetilde{T}_{v_j}(\omega_{k+1})$ for every other saturated node v_j.*

Proof. Proposition 4 guarantees that there is $\pi := (p, \mathrm{w}) \in \mathcal{PW}_v(\omega)$ such that $C(\pi) = T_v(\omega_{k+1})$, $\sum_{k=1}^{|p|-1} \mathrm{w}_k = \omega_{k+1}$, and $C(\pi(j)) = T_{v_j}(\omega_{k+1})$ for every saturated node v_j. By definition of v_i, we know that the total wait up to it in π, ω^i, is less than ω_{k+1}. As a consequence, $C(\pi(i)) = T_{v_i}(\omega_{k+1})$ yields $T_{v_i}(\omega) = T_{v_i}(\omega_{k+1})$ for all $\omega \in [\omega^i, \omega_{k+1}]$. By concavity of T_{v_i} on $[\omega_k, \omega_{k+1}]$, we conclude that $T_{v_i}(\omega) = T_{v_i}(\omega_{k+1})$ for all $\omega \in [\omega_k, \omega_{k+1}]$, hence $C(\pi(i)) = T_{v_i}(\omega_k)$.

□

The above two results immediately yield the concluding result of the section, which is essential to the justification of the polynomial algorithm described in next section. Indeed, $T_v(\omega_{k+1})$ can now be expressed as the concatenation of a path-with-waits up to some node u for a total wait not greater than ω_k, and a path-without-waits leaving u at $T_u(\omega_k) + \omega_{k+1}$. In particular, this means that given $T_u(\omega_k)$ for all $u \in V$, we will be able to focus on such paths during the search for ω_{k+1}.

Theorem 2. *Let $0 \leq \omega_k < \omega_{k+1} \leq W$ such that (11) holds and consider $v \in V$. Then,*

$$T_v(\omega_{k+1}) = \min_{u \in V \setminus \{v\}} \left\{T_v(\omega_k), T_u(\omega_k) + C\left(p_{u \to v}(\omega_k)\right)\right\},$$

where $p_{u \to v}(\omega_k)$ is the shortest path (without waits) from u to v, leaving u at $T_u(\omega_k) + \omega_{k+1}$.

5 Building the sequence

We describe next how to construct the sequence $\omega_1, \ldots, \omega_K = W$. From Proposition 5, we wish to define the next iterate ω_{k+1} as the largest value (smaller than W) such that for each $v \in V$ we have

$$T_v(\omega_{k+1}) + \omega_{k+1} \leq \tau_v(\omega_k). \tag{14}$$

Defining

$$\omega_v = \max\{\omega : T_v(\omega) + \omega \leq \tau_v(\omega_k)\},$$

we obtain $\omega_{k+1} = \min_{v \in V} \omega_v$ or ($\min_{v \in V} \omega_v > W$ and $\omega_{k+1} = W$).

Remark 3. *By definition of $\tau_v(\omega_k), v \in V$, we know that $\tau_v(\omega_k)$ is the breakpoint of a cost function of an arc leaving from v and $\tau_v(\omega_k) > T_v(\omega_k) + \omega_k$. So, after setting $\omega_{k+1} = \min_{v \in V} \omega_v$, we will get $\tau_{v^*}(\omega_{k+1}) > T_{v^*}(\omega_{k+1}) + \omega_{k+1} = \tau_{v^*}(\omega_k)$ for any $v^* \in \arg\min_{v \in V}\{\omega_v\}$. Since the function $\omega \mapsto T_v(\omega) + \omega$ is non-decreasing for all $v \in V$, we deduce that each breakpoint of each arc cost function can correspond to at most one element of the sequence $\omega_1, \ldots, \omega_K$. More formally, let $e = (v^*, u) \in E, s \in \{1, \ldots, r_e\}$ such that $\tau_{v^*}(\omega_k) = \tau_e^s$. Then, $\tau_{v^*}(\omega_l) > \tau_e^s$ for all $l > k$. As a consequence, $\omega_1, \ldots, \omega_K = W$ has at most as many elements as the total number of breakpoints in the arc cost functions, i.e., $\sum_{e \in E} r_e$.*

We see that for each $v \in V$,

$$T_v(\omega_v) + \omega_v = \tau_v(\omega_k). \tag{15}$$

If $\omega_v > W$ for all $v \in V$, none of the constraints (11) restricts the possible values for the next iterate ω_{k+1}, which is thus set to W. Otherwise, equality (15) holds for some $v^* \in \arg\min_{v \in V} \omega_v$. By definition, (11) holds for $\omega_{v^*} (= \omega_{k+1})$ so we can use Theorem 2 to rewrite (15) associated to v^* as

$$\omega_{v^*} = \tau_{v^*}(\omega_k) - \min_{u \in V} \left(T_u(\omega_k) + C(p_{u \to v^*})\right). \tag{16}$$

Recall that, for $u \in V$, $p_{u \to v^*}$ is the path (without waits) from u to v^*, the cost of which is minimum among all paths leaving u at $T_u(\omega_k) + \omega_{v^*}$. According to Proposition 4, there is

$$u^* \in \arg\min_{u \in V} \left\{T_u(\omega_k) + C(p_{u \to v^*})\right\},$$

such that every node v in $p_{u^* \to v^*}$ is reached at $T_v(\omega_{v^*}) + \omega_{v^*}$, and we know from (15) that $p_{u^* \to v^*}$ reaches v^* at time $\tau_{v^*}(\omega_k)$. What is more, $T_v(\omega_{v^*}) + \omega_{v^*} \leq \tau_v(\omega_k), \forall v \in V$, by definition of ω_{v^*}. The latter means in particular that for every node v in $p_{u^* \to v^*}$, the path leaves v at a time $t_{v \to v^*}$ that falls in $[T_v(\omega_k) + \omega_k, \tau_v(\omega_k)] \subseteq [\tau_v^-(\omega_k), \tau_v(\omega_k)]$. Hence, we can compute the departing time of $p_{u^* \to v^*}$ from node u^* by focusing on the paths without waits from u^* to v^* that reach v^* at $\tau_{v^*}(\omega_k)$ and depart from every intermediate node v in the time interval $[T_v(\omega_k) + \omega_k, \tau_v(\omega_k)]$.

For $u \in V, \bar{v} \in V$, we denote $\Pi_{u \to \bar{v}}(\omega_k)$ as the set of paths (without waits) from u to $\bar{v}$ that reach $\bar{v}$ at $\tau_{\bar{v}}(\omega_k)$ and depart from every intermediate node v at some time in $[T_v(\omega_k) + \omega_k, \tau_v(\omega_k)]$. Let also $t_{u \to \bar{v}}$ be the departing time from u in the shortest path among $\Pi_{u \to \bar{v}}(\omega_k)$, i.e.,

$$t_{u \to \bar{v}} = \tau_{\bar{v}}(\omega_k) - \min\{C(p) : p \in \Pi_{u \to \bar{v}}(\omega_k)\}.$$

In particular, we have seen that $p_{u^* \to v^*} \in \Pi_{u^* \to v^*}(\omega_k)$, so $t_{u^* \to v^*} = \tau_{v^*}(\omega_k) - C(p_{u^* \to v^*})$. We will show that for all $\bar{v} \in V$, $t_{u \to \bar{v}}$ can be computed by back-propagating the arrival time $\tau_{\bar{v}}(\omega_k)$ from $\bar{v}$ to u for all $u \in V$.

With the above definitions, we finally get $u^* \in \arg\max_{u \in V}\{t_{u\to v^*} - T_u(\omega_k)\}$ from which we conclude

$$\omega_{v^*} = t_{u^*\to v^*} - T_{u^*}(\omega_k).$$

Algorithm 2 details the back-propagation which computes for each $v \in V$ the departure time $t_{v\to\bar{v}}$ of the shortest path (without waits) from v to $\bar{v}$, reaching $\bar{v}$ at $T_{\bar{v}}(\omega_k)$. Let t_u and t_v denote the departure times at nodes u and v, respectively, and let us assume that $t_u \in [T_u(\omega_k) + \omega_k, \tau_u(\omega_k)]$. Then, if arc (u, v) belongs to the shortest path-without-waits from u to $\bar{v}$, we can relate t_u and t_v through the formula

$$t_v = t_u + C_{uv}(t_u) = t_u + \rho_{uv}(\omega_k)t_u + c_{uv}(\omega_k). \tag{17}$$

Steps 2–2 describe a backward label-setting algorithm based on formula (17), with a difficulty in the case $\rho_{uv}(\omega_k) = -1$. When that happens, (17) becomes $t_v = c_{uv}(\omega_k)$, so the departure time at v does not depend on the departure time at u. Thus, if $c_{uv}(\omega_k) \le t_v$, we can wait at node u until reaching $\tau_u(\omega_k)$, which implies that $\omega_u \le \omega_{\bar{v}}$. Otherwise, $c_{uv}(\omega_k) > t_v$ and it is impossible to reach node v at time t_v by taking arc (u, v). We can thus skip that iteration in the for loop. Steps 2 and 2 ensure that $t_u \in [T_u(\omega_k) + \omega_k, \tau_u(\omega_k)]$ throughout the algorithm as proved below in Lemma 1.

input : $\omega_k, \bar{v} \in V, \tau_u(\omega_k), T_u(\omega_k), \forall u \in V,\ c_e(\omega_k), \rho_e(\omega_k), \forall e \in E$
$V' \leftarrow V$
$S \leftarrow \emptyset$ `// set of marked nodes`
$t_u \leftarrow -\infty, \forall u \in V'$
$t_{\bar{v}} \leftarrow \tau_{\bar{v}}(\omega_k)$
$v \leftarrow \bar{v}$
while $v \neq o$ *and* $S \neq V'$ **do**
  $S \leftarrow S \cup \{v\}$ `// mark node` v
  for $(u, v) \in E$ `// back-propagation loop of` t_v `to its predecessors`
  do
    if $\rho_{uv}(\omega_k) > -1$ **then** $t_u \leftarrow \max\left\{t_u, \frac{t_v - c_{uv}(\omega_k)}{1+\rho_{uv}(\omega_k)}\right\}$
    else
      if $c_{uv}(\omega_k) \le t_v$ **then** $\bar{\omega}_{\bar{v}} \leftarrow +\infty$, **STOP** `//` $\tau_u(\omega_k)$ `is reached before` $\tau_{\bar{v}}(\omega_k)$
      else continue `//` t_v `will not be reached, skip arc` (u, v)
  select v in $\arg\max_{u \in V' \setminus S}\{t_u\}$
  if $t_v > \tau_v(\omega_k)$ **then** $\bar{\omega}_{\bar{v}} \leftarrow +\infty$, **STOP** `//` $\tau_v(\omega_k)$ `is reached before` $\tau_{\bar{v}}(\omega_k)$
  if $t_v < T_v(\omega_k) + \omega_k$ **then** $V' \leftarrow V' \setminus \{v\}$, go to step 2 `// Impossible to backtrack to` v `late enough`
$\bar{\omega}_{\bar{v}} \leftarrow \max_{u \in V'}(t_u - T_u(\omega_k))$
output: $\bar{\omega}_{\bar{v}}$

Algorithm 2: Back-propagation of $\tau_v(\omega_k)$.

Lemma 1. *Consider the node v selected at an execution of step 2 of Algorithm 2, i.e., $v \in \arg\max_{u \in V' \setminus S}\{t_u\}$. We have that*

1. *if $T_v(\omega_k) + \omega_k \le t_v \le \tau_v(\omega_k)$, $t_v = t_{v\to\bar{v}}$;*
2. *if $t_v < T_v(\omega_k) + \omega_k$, then $\Pi_{v\to\bar{v}}(\omega_k) = \emptyset$;*

3. *if* $t_v > \tau_v(\omega_k)$, *then* $\omega_{\bar{v}} \geq \omega_v$.

Proof. One can verify that the three items are satisfied if the conditions checked at steps 2 and 2 have been false since the beginning of the execution of Algorithm 2. Indeed, in such case, the algorithm is a plain back-propagation for the computation of shortest paths with affine arc costs.

The complete proof of items 1 and 2 is by induction on the number of executions of step 2. At the first execution of this step,

1. if $T_v(\omega_k) + \omega_k \leq t_v \leq \tau_v(\omega_k)$, the above observation yields $t_v = t_{v\to\bar{v}}$.
2. $t_v < T_v(\omega_k) + \omega_k$: assume by contradiction that there is $p \in \Pi_{v\to\bar{v}}(\omega_k)$, and take v^- the predecessor of $\bar{v}$ in this path. At the first iteration, Algorithm 2 backpropagates along the arcs going to $\bar{v}$, so p departs from v^- at t_{v^-}. Then, by definition, $v \in \arg\max\{t_u\}$, so $t_v \geq t_{v^-}$. Since path p must depart from v before departing from v^-, the above yields that it departs from v before $t_v < T_v(\omega_k) + \omega_k$, which is in contradiction with the definition of $\Pi_{v\to\bar{v}}(\omega_k)$.

We now consider a later execution of step 2, where $v \in \arg\max_{u\in V'\setminus S}\{t_u\}$, assuming that the two items hold at each previous iteration. For every node u selected at a previous execution of step 2, the definition of the algorithm involves that $u \in S$ if $T_u(\omega_k) + \omega_k \leq t_u \leq \tau_u(\omega_k)$ and $u \in V \setminus V'$ if $t_u < T_u(\omega_k) + \omega_k$. This means that for all marked nodes $u \in S$, $t_u = t_{u\to\bar{v}}$, and that no path in $\Pi_{u\to\bar{v}}(\omega_k)$ goes through a node of $V \setminus V'$. Moreover, $t_u > \tau_u(\omega_k)$ did not happen, otherwise the algorithm would have been terminated.

To show item 1, observe that we would have obtained the same value for t_v if the back-propagation had been run on the subgraph induced by V'. Moreover, the nodes of $V \setminus V'$ are not involved in the paths of $\Pi_{v\to\bar{v}}(\omega_k)$, so $t_{v\to\bar{v}}$ can be computed by considering the subgraph induced by V'. In this subgraph, the conditions checked at steps 2-2 would have been false at every previous iteration, so if $T_v(\omega_k) + \omega_k \leq t_v \leq \tau_v(\omega_k)$, t_v is the result of a classical back-propagation. As a consequence, $t_v = t_{v\to\bar{v}}$.
The proof of item 2 is then similar to that given in the initialization of the induction.

To prove item 3., assume that $t_v > \tau_v(\omega_k)$ at some execution of step 2. Let p be the path from v to $\bar{v}$ constructed by the back-propagation and denote as v^+ the successor of v in p. Let also $\pi^v \in \mathcal{PW}_v(\omega_v)$ be a path-with-wait from o to v that reaches v at $\tau_v(\omega_k) < t_v$ with a total wait equal to ω_v (its existence is guaranteed by the definition of ω_v in (14)). Taking path arc (v, v^+) immediately after π^v (no wait at v), v^+ is reached at

$$t^+ = \tau_v(\omega_k) + \rho_{vv^+}(\omega_k)\tau_v(\omega_k) + c_{vv^+}(\omega_k).$$

By $\tau_v(\omega_k) < t_v$, we know that v^+ is reached earlier in this path than in p, i.e., $t^+ \leq t_{v^+}$. If we then wait $t_{v^+} - t^+$ at v^+, we can take the end of p from v^+ to reach $\bar{v}$ at $\tau_{\bar{v}}(\omega_k)$. Now, if π is the path-with-wait from o to $\bar{v}$ constructed above, we get that π reaches $\bar{v}$ at $\tau_{\bar{v}}(\omega_k)$ with a total wait

$$\omega_\pi = \omega_v + t_{v^+} - t^+ \geq \omega_v. \tag{18}$$

Stated otherwise, we have $\tau_{\bar{v}}(\omega_k) = C(\pi) + \omega_\pi$ where $C(\pi) \geq T_{\bar{v}}(\omega_\pi)$, so $T_{\bar{v}}(\omega_\pi) + \omega_\pi \leq \tau_{\bar{v}}(\omega_k)$. Recall that by definition, $\omega_{\bar{v}} = \max\{\omega : T_{\bar{v}}(\omega) + \omega \leq \tau_{\bar{v}}(\omega_k)\}$, hence $\omega_\pi \leq \omega_{\bar{v}}$. Using (18), we conclude that $\omega_{\bar{v}} \geq \omega_v$.

□

The main algorithm for $TDSPW$ is given in Algorithm 3. The algorithm first computes $T_v(\omega_{k+1})$ given $T_v(\omega_k)$ for all $v \in V$, using the adaptation of Dijkstra's label setting algorithm

initialization: $k = 1, \omega_1 = 0, \omega_0 = -1, T_v(\omega_0) = +\infty \; \forall v \in V \setminus \{o\}, T_o(\omega_0) = 0$
execute Algorithm 1 to compute $T_v(\omega_k) \; \forall v \in V$
if $\omega_k = W$ **then STOP**
for $\bar{v} \in V$ **do** compute $\tau_{\bar{v}}(\omega_k)$
for $\bar{v} = 1, \ldots, n$ **do** execute Algorithm 2 to compute $\bar{\omega}_{\bar{v}}$
$\omega_{k+1} = \min\left(W, \min_{\bar{v} \in V} \bar{\omega}_{\bar{v}}\right)$
$k + 1 \leftarrow k$
go to **step 3**
output : $T_v(W), \forall v \in V$

Algorithm 3: Computing $T_d(W)$.

given in Algorithm 1. The validity of Algorithm 1 is only guaranteed if (9) holds, which is guaranteed by Corollary 1 if at each iteration of Algorithm 3, we can set $\omega_{k+1} := \min\{W, \min_{u \in V} \omega_u\}$, as discussed at the beginning of the section. The computation of ω_{k+1} is done in steps 3 to 3 of Algorithm 3. The validity of the approach is justified below.

To justify that ω_{k+1} is well defined in Algorithm 3, we prove that $\min_{\bar{v} \in V} \bar{\omega}_{\bar{v}} = \omega_{v^*}$, where $\omega_{v^*} = \min_{v \in V} \omega_v$ as in the discussion introducing the section. Lemma 1 and the discussion that precedes it justify that Algorithm 2 returns $\bar{\omega}_{v^*} = \omega_{v*}$. In contrast, if $\bar{v} \notin \arg\min_{u \in V}\{\omega_u\}$, it is not necessarily true that $\bar{\omega}_{\bar{v}} = \omega_{\bar{v}}$, but we will prove below that $\bar{\omega}_{\bar{v}} \geq \omega_{v*}$. The above statements combined guarantee that

$$\min_{\bar{v} \in V} \bar{\omega}_{\bar{v}} = \omega_{v^*}. \tag{19}$$

Let $\bar{v} \in V$: to prove that $\bar{\omega}_{\bar{v}} \geq \omega_{v*}$, first observe that item 3 of Lemma 1 guarantees that $\omega_{\bar{v}} \geq \omega_{v^*}$ if Algorithm 2 returns $\bar{\omega}_{\bar{v}} = +\infty$. We then consider any $\bar{v} \in V$ such that $\bar{\omega}_{\bar{v}} < +\infty$. By definition of ω_{v^*}, $T_v(\omega_{v^*}) + \omega_{v^*} \leq \tau_v(\omega_k)$ for each $v \in V$, so the application of Corollary 1 yields

$$T_{\bar{v}}(\omega_{v^*}) = \min\{T_{\bar{v}}(\omega_k), \widetilde{T}_{\bar{v}}(\omega_{v^*})\}.$$

At step 2 of Algorithm 2, we initialize $t_{\bar{v}}$ to $\tau_{\bar{v}}(\omega_k)$, and this value is not modified in the rest of the algorithm, hence $\bar{\omega}_{\bar{v}} \geq \tau_{\bar{v}}(\omega_k) - T_{\bar{v}}(\omega_k)$. As a consequence,

$$T_{\bar{v}}(\omega_{v^*}) = T_{\bar{v}}(\omega_k) \implies \bar{\omega}_{\bar{v}} \geq \tau_{\bar{v}}(\omega_k) - T_{\bar{v}}(\omega_{v^*}) \geq \omega_{v^*}.$$

Now, assume that $T_{\bar{v}}(\omega_{v^*}) = \widetilde{T}_{\bar{v}}(\omega_{v^*})$, and let $\pi = (p, w)$ be a path-with-waits constructed as in Corollary 2 so that

- $C(\pi) = T_v(\omega_{v^*})$ and $\sum_{k=1}^{|p|-1} \mathrm{w}_k = \omega_{v^*}$;
- $C(\pi(i)) = T_{v_i}(\omega_k)$, where v_i is the first saturated node in π;
- and $C(\pi(j)) = T_{v_j}(\omega_{v^*})$ for every other saturated node v_j.

In particular, π connects v_i to $\bar{v}$ without waits (after v_i) with a cost equal to $T_{\bar{v}}(\omega_{v^*}) - T_{v_i}(\omega_k)$. We claim that the subpath of p leaving from v_i, $\bar{p}(i)$, is in $\Pi_{v_i \to \bar{v}}(\omega_k)$, meaning that if we backpropagate $\tau_{\bar{v}}(\omega_k)$ from $\bar{v}$ to v_i along $\bar{p}(i)$, we reach every intermediate node v_j in the time interval $[T_{v_j}(\omega_k) + \omega_k, \tau_{v_j}(\omega_k)]$. Since every node from v_i to $\bar{v}$ is also reached this time interval in π, the cost of that backpropagated path is also $T_{\bar{v}}(\omega_{v^*}) - T_{v_i}(\omega_k)$. Now, by definition of $t_{v_i \to \bar{v}}$, $\tau_{\bar{v}}(\omega_k) - t_{v_i \to \bar{v}}$ is the minimum cost of a path without wait from v_i to $\bar{v}$ among those that reach $\bar{v}$ at $\tau_{\bar{v}}(\omega_k)$, hence

$$\tau_{\bar{v}}(\omega_k) - t_{v_i \to \bar{v}} \leq T_{\bar{v}}(\omega_{v^*}) - T_{v_i}(\omega_k).$$

Using Lemma 1 ($t_{v_i} = t_{v_i \to \bar{v}}$ when v_i is marked) and the definition of $\bar{\omega}_{\bar{v}}$ in Algorithm 2, we conclude that

$$\bar{\omega}_{\bar{v}} \geq t_{v_i \to \bar{v}} - T_{v_i}(\omega_k) \geq \tau_{\bar{v}}(\omega_k) - T_{\bar{v}}(\omega_{v^*}) \geq \omega_{v^*},$$

where the last inequality follows from $\tau_{\bar{v}}(\omega_k) \geq T_{\bar{v}}(\omega_{v^*}) + \omega_{v^*}$.
To prove the claim, observe that π reaches $\bar{v}$ at $T_{\bar{v}}(\omega_{v^*}) + \omega_{v^*}$, so if we back-propagate $T_{\bar{v}}(\omega_{v^*}) + \omega_{v^*}$ from $\bar{v}$ to v_i along $\bar{p}(i)$, we reach every intermediary node v_j at $C(\pi(j)) = T_{v_j}(\omega_{v^*}) + \omega_{v^*}$. Using that $\tau_{\bar{v}}(\omega_k) \geq T_{\bar{v}}(\omega_{v^*}) + \omega_{v^*}$, we observe that if we back-propagate $\tau_{\bar{v}}(\omega_k)$ from $\bar{v}$ to v_i along $\bar{p}(i)$, we necessarily reach the intermediary nodes v_j later than when back-propagating $T_{\bar{v}}(\omega_{v^*}) + \omega_{v^*}$, i.e., later than $T_{v_j}(\omega_{v^*}) + \omega_{v^*}$. Moreover, if one intermediary node is reached later than $\tau_{v_j}(\omega_k)$, then Algorithm 2 necessarily stops at step 2, which is not possible if $\bar{\omega}_{\bar{v}} < +\infty$. By $T_{v_j}(\omega_{v^*}) + \omega_{v^*} \geq T_{v_j}(\omega_k) + \omega_k$, we get that every intermediary node v_j is also reached in the time interval $[T_{v_j}(\omega_k) + \omega_k, \tau_{v_j}(\omega_k)]$ when back-propagating $\tau_{\bar{v}}(\omega_k)$ from $\bar{v}$ to v_i along $\bar{p}(i)$.

Combining Propositions 1 and equation (19) proves the validity of Algorithm 3. Its time complexity follows.

Theorem 3. *Algorithm 3 runs in* $\mathcal{O}\left((\sum_{e \in E} r_e)(n+1)(n\log(n)+m)\right)$.

Proof. Equation (19) shows that the sequence $\omega_1, \ldots, \omega_K$ is constructed as expected, so Remark 3 justifies that K is at most equal to the total number of breakpoints in the arc cost functions. The number of iterations K then satisfies $K \leq \sum_{e \in E} r_e$. At each iteration, Algorithms 1 is run once whereas Algorithm 2 needs to be called n times. Furthermore, Algorithms 1 and 2 can be both implemented in $\mathcal{O}(n\log(n)+m)$ using a Fibonacci heap.

□

References

[1] R. K. Ahuja, J. B. Orlin, S. Pallottino, and M. Grazia Scutellà. Minimum time and minimum cost-path problems in street networks with periodic traffic lights. *Transportation Science*, 36(3):326–336, 2002.

[2] R. K. Ahuja, J. B. Orlin, S. Pallottino, and M. G. Scutella. Dynamic shortest paths minimizing travel times and costs. *Networks*, 41(4):197–205, 2003.

[3] R. Bellman. On a routing problem. *Quarterly of applied mathematics*, 16(1):87–90, 1958.

[4] X. Cai, T. Kloks, and C.-K. Wong. Time-varying shortest path problems with constraints. *Networks*, 29(3):141–150, 1997.

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

[6] 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.

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

[8] S. E. Dreyfus. An appraisal of some shortest-path algorithms. *Operations Research*, 17(3):395–412, 1969.

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

[10] M. L. Fredman and R. E. Tarjan. Fibonacci heaps and their uses in improved network optimization algorithms. *Journal of the ACM (JACM)*, 34(3):596–615, 1987.

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

[12] J. Omer and M. Poss. Time-dependent shortest path with discounted waits. *Networks*, 2018. In press.