On The Complexity of Maximizing Temporal Reachability via Trip Temporalisation

Filippo Brunelli, Pierluigi Crescenzi, Laurent Viennot

On The Complexity of Maximizing Temporal Reachability via Trip Temporalisation

Filippo Brunelli[1], Pierluigi Crescenzi[2], and Laurent Viennot[1]

[1]Inria, Université de Paris, CNRS, IRIF, F-75013 Paris, France, suported by ANR project Multimod ANR-17-CE22-0016.
[2]Gran Sasso Science Institute, I-67100 L'Aquila, Italy

November 16, 2021

Abstract

We consider the problem of assigning appearing times to the edges of a digraph in order to maximize the (average) temporal reachability between pairs of nodes. Motivated by the application to public transit networks, where edges cannot be scheduled independently one of another, we consider the setting where the edges are grouped into certain walks (called trips) in the digraph and where assigning the appearing time to the first edge of a trip forces the appearing times of the subsequent edges. In this setting, we show that, quite surprisingly, it is NP-complete to decide whether there exists an assignment of times connecting a given pair of nodes. This result allows us to prove that the problem of maximising the temporal reachability cannot be approximated within a factor better than some polynomial term in the size of the graph. We thus focus on the case where, for each pair of nodes, there exists an assignment of times such that one node is reachable from the other. We call this property strong temporalisability. It is a very natural assumption for the application to public transit networks. On the negative side, the problem of maximising the temporal reachability remains hard to approximate within a factor $\sqrt{n}/12$ in that setting. Moreover, we show the existence of collections of trips that are strongly temporalisable but for which any assignment of starting times to the trips connects at most an $O(1/\sqrt{n})$ fraction of all pairs of nodes. On the positive side, we show that there must exist an assignment of times that connects a constant fraction of all pairs in the strongly temporalisable and symmetric case, that is, when the set of trips to be scheduled is such that, for each trip, there is a symmetric trip visiting the same nodes in reverse order.

Keywords:edge labeling edge scheduled network network optimisation temporal graph temporal path temporal reachability time assignment

1 Introduction

A typical question that has been considered in the literature in the field of *network optimization* consists in finding the graph modification which maximises or minimises a specific optimisation criterion (often connected to the notion of reachability and distance between nodes). For example, one well studied problem has been the following one: given a graph and given a budget k, find a set of k edges, which, when added to the graph, minimise the diameter of the new graph (see, for example, [1, 16, 19]). Other *graph operations* that have been studied are node or edge deletions (see, for example, [31] and [29], respectively) or edge contractions (see, for example, [21]), while other *optimisation criteria* are reachability, that is, the number of pairs of nodes (u, v) such that u can reach v (see, for example, [12]), information diffusion, that is, the number of nodes reachable from a specific set of source nodes (see, for example, [13]), or centrality measures, such as the closeness and the betweenness centrality of a node (see, for example, [14] and [5], respectively).

In this paper, we study a network optimisation problem which is related to the notion of reachability in temporal graphs (see Section 1.3 for similar problems). *Temporal graphs* (also called edge-scheduled networks [6], dynamic graphs [24], temporal networks [28], evolving graphs [7], time-stamped graphs [11]-, time-varying graphs [9], link streams [30], or point-availability time-dependent networks [8]) have received increasing attention over the last two decades [26, 27, 32, 36]. Indeed, they have been repeatedly used in order to study classical notions from the field of graph theory (such as degree, path, connectivity, clique, and so on) in a more realistic framework, in which the topology of the graph changes over time. As stated in [30], studying temporal graphs has applications in a wide variety of contexts, ranging from phone calls to contact tracing, from cattle exchanges to messaging, from communication traffic to public transport systems. In one of its formulations (which is the one we adopt in this paper), a temporal graph is a list of *temporal edges* (u, v, t, λ), where u and v are two nodes of the graph, t is the *appearing time* of the temporal edge, and λ is its *travel time* (that is, we can traverse the edge starting from u at time t and arrive in v at time $t + \lambda$, which is the *arrival time* of the temporal edge).

In our network optimisation problem, the graph operation will allow us to transform a weighted directed graph (in which the weight of an edge has to be interpreted as its travel time) into a temporal graph (see below), while the optimisation criterion will be the *temporal reachability* of a temporal graph, that is, the number of pairs of nodes u and v such that v is temporally reachable from u. Note that, in temporal graphs, a (temporal) path, differently from the notion of walk in graphs, requires a natural temporal constraint: that is, the appearing time of an edge in the path has to be at least equal to the arrival time of the edge which precedes it in the path. This additional constraint makes the computation of the nodes reachable from a node a little bit more complicated than in the case of graphs, but still doable in time $\tilde{O}(m)$, where m is the number of temporal edges and the notation $\tilde{O}$ hides poly-logarithmic factors (see, for example, [39]). Note that we do not put any constraint on how long it is possible to wait at a node in-between two temporal edges, as these further constraints can dramatically change the complexity of such temporal path computation [10].

In order to introduce our network optimisation problem, let us first consider a simplified version of it, in which the graph operation is the *edge temporalisation* (a similar problem has been considered in [23], where unweighted undirected multigraphs are considered and the graph operation is called *labeling*). Given a weighted directed multigraph D, this operation simply assigns to each edge (u, v, λ) of D an appearing time t, making it a temporal edge (u, v, t, λ). For example, the weighted directed multigraph D could represent the connections between airports of an air company, where the weight of an edge could represent the duration of the corresponding flight (multiple edges between two nodes could correspond to multiple flights between the two corresponding cities). An edge temporalisation of D would then correspond to an assignment of the leaving time of each flight, and the goal would be to connect as many pairs of cities as possible (clearly, other optimisation criteria would be interesting for this application, such as minimizing the average duration of all indirect connections, but guaranteeing a high reachability of the network seems to be a basic prerequisite). As an example, let us consider the weighted directed multigraph D shown in the left part of Figure 1. In the middle and right parts of the figure, we show two edge temporalisations

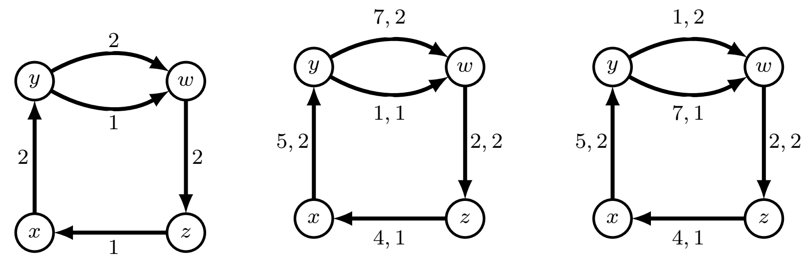

Figure 1: A weighted directed multigraph D (left), and two edge temporalisations of D (middle and right), which differ only for the appearing times of the two edges from y to w (the edge labels denote the appearing time and the travel time, respectively, of the edge). The temporal reachability of the first temporal graph is 15, while the temporal reachability of the second temporal graph is 13.

of D, which differ only for the appearing times assigned to the two edges from y to w. In the temporalisation in the middle part of the figure, the edge $(y, w, 1)$ has been assigned the appearing time 1, while the edge $(y, w, 2)$ has been assigned the appearing time 7. The opposite assignment has been used, instead, by the edge temporalisation in the right part of the figure.

Note that, in the temporal graph in the middle part of the figure, node x can reach nodes x, y, and w, while nodes y, w, and z can reach all nodes: the temporal reachability, in this case, is equal to $3 + 3 \times 4 = 15$. In the temporal graph in the right part of the figure, instead, the situation is similar for all nodes but y. Indeed, node x can reach nodes x, y, and w, node y can reach nodes y and w, and nodes w and z can reach all nodes: the temporal reachability, in this case, is equal to $3 + 2 + 2 \times 4 = 13$, thus showing that the first edge temporalisation is better than the second one.

A natural network optimisation problem, based on edge temporalisation and temporal reachability, is then the following one: given a weighted directed multigraph D, find an edge temporalisation of D which maximises the temporal reachability of the resulting temporal graph. For example, in the case of the weighted directed multigraph shown in the left part of Figure 1, it is easy to verify that the edge temporalisation shown in the middle part of the figure is an optimal one. Note that in [23], it is proved that deciding whether the maximum temporal reachability is equal to the square of the number of nodes is a NP-complete problem, in the setting of unweighted undirected multigraphs.

1.1 The maximum reachability trip temporalisation problem

Our network optimisation problem is mostly inspired by the fact that, in some real-world applications such as public transport systems, the edges of the weighted directed multigraph can be "used" more than one time (for example, by several vehicles) and they are not "independent", in the sense that an appearing time cannot be assigned to an "instance" of an edge independently of the appearing time assigned to "instances" of other edges. For example, in the case of a public transport system, the "instances" of the edges are grouped in "trips", where each trip is a sequence of edges forming a walk in the weighted directed multigraph, corresponding to the journey of a vehicle. When several vehicles travel along the same walk, we assume that a distinct trip is associated to each one of them. We also assume that the waiting time at a stop is negligible and that scheduling a vehicle amounts to assigning an appearing time to the first edge of the corresponding

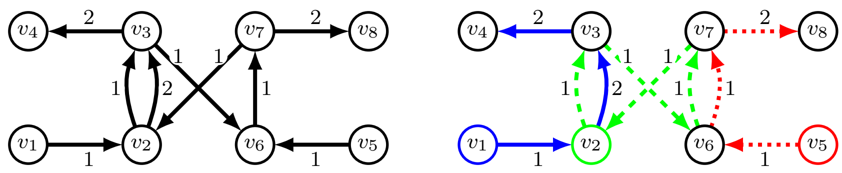

Figure 2: An example of a weighted directed multigraph D (left) and a collection of trips on D (right), where each starting node of a trip has a colored border. The edge $(v_6, v_7, 1)$ of D is "used" by both the green dashed trip T_2 and the red dotted trip T_3. The duration of the blue solid trip T_1 is 5, while the duration of the other two trips is 4. No trip temporalisation exists such that both v_8 is reachable from v_1 and v_4 is reachable from v_5 in the induced temporal graph.

trip, since all the other appearing times are a consequence of it: indeed, each edge of the trip appears right after the arrival of the previous one. More precisely, given a trip $T = e_1, \ldots, e_k$, where, for $i \in [k]$,[1] $e_i = (u_i, v_i, \lambda_i)$, assigning a starting time t to T results in the set of temporal edges $f_i = (u_i, v_i, t + \sum_{j=1}^{i-1} \lambda_j, \lambda_i)$, for $i \in [k]$.[2]

The graph operation that we will consider is, then, the *trip temporalisation*. Given a weighted directed multigraph D and a collection of trips on D (that is, a collection of not necessarily distinct walks in D), this operation simply assigns to each trip a starting time. The temporal graph *induced* by a trip temporalisation is the temporal graph whose set of nodes is the same as the set of nodes of D, and whose set of temporal edges is the disjoint union of all the temporal edges resulting from the assignment of the starting times to the trips in the input collection. For example, let us consider the weighted directed multigraph D shown in the left part of Figure 2, and the following collection of trips on D (depicted in the right part of the figure): $T_1 = (v_1, v_2, 1), (v_2, v_3, 2), (v_3, v_4, 2)$ (blue solid trip), $T_2 = (v_2, v_3, 1), (v_3, v_6, 1), (v_6, v_7, 1), (v_7, v_2, 1)$ (green dashed trip), and $T_3 = (v_5, v_6, 1), (v_6, v_7, 1), (v_7, v_8, 2)$ (red dotted trip). Note how the edge $(v_6, v_7, 1)$ is "used" by two different trips (that is, T_2 and T_3): this might correspond to two different vehicles travelling through this edge. Note also that there is no trip temporalisation such that both v_8 is reachable from v_1 and v_4 is reachable from v_5 in the induced temporal graph. Indeed, if v_8 is reachable from v_1, then the starting time assigned to T_1 has to be smaller than the starting time assigned to T_3, while if v_4 is reachable from v_5, then the starting time assigned to T_3 has to be smaller than the starting time assigned to T_1: these two inequalities cannot be both satisfied. Let us consider the trip temporalisation which assigns to T_1 the starting time 1, to T_2 the starting time 6, and to T_3 the starting time 10 (this trip temporalisation intuitively corresponds to scheduling the three trips one after the other). The temporal graph induced by this trip temporalisation is shown in Figure 3. In this temporal graph, the node v_1 can reach all nodes but node v_5, while node v_7 can reach only nodes v_2, v_7, and v_8. The reachability of this temporal graph is equal to 30. On the other hand, it is possible to verify that the trip temporalisation which assigns to T_1 the starting time 9, to T_2 the starting time 5, and to T_3 the starting time 1 induces a temporal graph whose reachability is 32 (see also Table 2 at page 9).

The network optimisation problem, called Maximum Reachability Trip Temporalisation (in short, mrtt), that we will analyse in this paper is, hence, the following one: *given a weighted directed multigraph D and a collection of trips $\mathbb{T}$ on D, find a trip temporalisation of $\mathbb{T}$ which*

[1] In the following, for any $h \in \mathbb{N}$, we denote by $[h]$ the set $\{1, 2, \ldots, h\}$.
[2] As it is standard, we assume that the summation with no summands evaluates to zero.

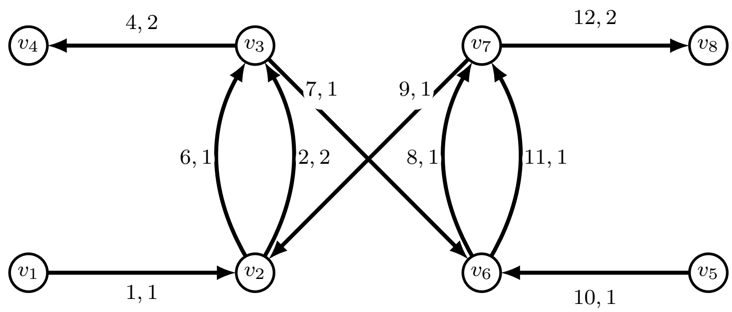

Figure 3: The temporal graph induced by the following temporalisation of the trips depicted in the right part of Figure 2: the starting time of T_1 is 1, of T_2 is 6, and of T_3 is 10.

maximises the reachability of the induced temporal graph. Note that MRTT is a generalisation of the the network optimisation problem based on the edge temporalisation: indeed, the latter one corresponds to the case in which the trip collection is just the collection of all edges in D.

1.2 Our results

Our results are summarised in Table 1, where two other combinatorial problems are also considered. The first decision problem, denoted by O2O-RTT, is the one-to-one version of the MRTT problem, in which the question is whether a trip temporalisation exists making one given node t temporally reachable from another given node s. The second maximisation problem, denoted by SS-MRTT, is the single-source version of the MRTT problem, in which the question is to find a trip temporalisation maximising the number of nodes temporally reachable from a given source node s.

Note that, although we assume negligible waiting times in the sense that edges of a trip must be scheduled one right after the other, our results can be generalized in a setting where, for each pair of consecutive edges of a trip, a fixed waiting time is imposed. The reason is similar to the fact that we can restrict ourselves to a setting where all travel times are 1 as explained in the Section 2.

Quite surprisingly, our first result (see Theorem 1) shows that *the* O2O-RTT *problem is NP-complete*. Using a classical gap technique, we then obtain that if P $\neq$ NP, then *the* MRTT *and the* SS-MRTT *problems cannot be approximated within a factor* $n^{1-\varepsilon}$ *for any* $\varepsilon > 0$*, where* n *is the number of nodes* (see Theorems 2 and 3). We also show that the parameterised version of the O2O-RTT problem with respect to the number k of trips used in the resulting temporal graph, in order to go from s to t, can be solved in time $2^{O(k)} m \log |\mathbb{T}|$ where $m = \sum_{T \in \mathbb{T}} |T|$ is the sum of trip lengths (see Theorem 4).

The above non-approximability results are the main reason for focusing our attention on an interesting restriction of the MRTT problem, that is, the one in which the collection of trips $\mathbb{T}$ satisfies the very natural property of being "temporally" strongly connected in the following sense. A collection of trips $\mathbb{T}$ is *strongly temporalisable* if, for each pair of nodes u and v, there exists a trip temporalisation of $\mathbb{T}$ that allows u to (temporally) reach v. Note that this requirement is a rather weak one, since we are not asking for a unique trip temporalisation, but for a trip temporalisation for each pair of nodes (indeed, this is a requirement which is satisfied in many applications of temporal graphs). We first show that the strong temporalisability property is not sufficient to get high reachability. To this aim, we prove that *there exists an infinite family of trip collections, all*

Maximisation problems	
Problem	**Complexity**
MRTT	Not approximable within a factor $n^{1-\varepsilon}$ for any $\varepsilon > 0$ (Theorem 2 and Theorem 3)
SS-MRTT	

Decision problems	
Problem	**Complexity**
O2O-RTT	NP-complete (Theorem 1)
k-O2O-RTT	Solvable in time $2^{O(k)} m \log \lvert\mathbb{T}\rvert$ (Theorem 4)

	Property	
Problem	*Strongly temporalisable*	*Strongly temporalisable and symmetric*
O2O-RTT	Linear-time solvable (trivial)	
MRTT	Not approximable within a factor less than $\sqrt{n}/12$ (Theorems 6 and 7)	NP-hard (Theorem 8) and r-approximable for some $r > 0$ (Theorem 9)
SS-MRTT		Linear-time solvable (consequence of Fact 4)

Table 1: Our results assuming P $\neq$ NP (n denotes the number of nodes, $m = \sum_{T \in \mathbb{T}} |T|$ denotes the sum of trip lengths and k denotes the number of trips that can be used in a temporal path). The approximability results are obtained by proving that we can get a high temporal reachability (that is, a temporal reachability proportional to the total number of pairs of nodes). A further result is Theorem 5, which intuitively states that the strong temporalisability property is not sufficient to get high reachability. The table leaves as the main open problem the question whether the MRTT and the SS-MRTT problems are approximable within a sub-linear factor, when restricted to strongly temporalisable collections of trips.

strongly temporalisable, such that any trip temporalisation connects at most an $O(1/\sqrt{n})$ fraction of all pairs (see Theorem 5). By using this construction, we then show that if P $\neq$ NP, then *the* MRTT *and the* SS-MRTT *problems cannot be approximated within a factor less than $\sqrt{n}/12$, when restricted to strongly temporalisable trip networks* (see Theorems 6 and 7).

However, the situation changes if we add another quite natural property of a trip collection, that is, symmetricity. A trip collection $\mathbb{T}$ is *symmetric* if, for each trip $T \in \mathbb{T}$, $\mathbb{T}$ includes also the reverse trip, that is, the trip starting from the last node of T, arriving in the first node of T, and passing through all the nodes in T in reverse order. For example, referring to public transport systems, symmetricity is almost always respected, since for any bus/metro/tramway trip, there is usually also the same bus/metro/tramway trip in the opposite direction. It is quite easy to show that, if the collection of trips $\mathbb{T}$ is *symmetric*, then $\mathbb{T}$ is strongly temporalisable if and only if the weighted directed multigraph D is strongly connected (see Corollary 1).

We show that *the* MRTT *problem is NP-hard even if restricted to symmetric and strongly temporalisable collection of trips* (see Theorem 8). However, our final result shows that, *given a symmetric and strongly temporalisable collection of trips, it is possible to find in polynomial time a trip temporalisation achieving a reachability proportional to the total number of pairs* (see Theorem 9). This implies the existence of a constant-factor approximation algorithm in the symmetric and strongly temporalisable setting.

All our hardness results are proved starting from the 3-SAT problem, which is NP-complete [22]. Moreover, it is easy to show that the decision and maximisation problems we consider are in NP or in NPO (that is, the class of NP optimisations problems [3]), respectively. Indeed, given an instance of the problem and a trip temporalisation, checking that a node t is reachable from a node s in the induced temporal graph can be done in polynomial time (see, for example, [39]).

1.3 Related work

Problems similar to the one considered in this paper have already been analysed [28, 33, 13, 18, 34, 37, 35]. For instance, in [18] the authors consider the problem of deleting edges from a given temporal graph in order to reduce its reachability, motivated by the context of epidemiology. Later on, the temporal dimension of a temporal network has suggested the following quite natural network modification: given a graph, assign the appearing time of its edges in order to minimise the reachability of the resulting temporal graph. This problem has been proved NP-complete in [17] (the authors pose as an open problem the existence of a constant-factor approximation algorithm). Another closely related work [15] studies the problem of minimising the average reachability (as well as other similar objectives which could be interesting goals in the context of transport networks) in a temporal graph by delaying some edges. Various NP-hardness results as well as a polynomial-time algorithm are given, depending on the type of delay operations that are permitted. The authors leave as open the complexity of maximising reachability which is similar to our goal.

As far as we know, the trip temporalisation problem has never been studied before. The topic of temporalising edges to increase reachability is connected to gossip and broadcasting protocols (see [25] for a survey). However, apart from the already mentioned [23] where the undirected setting makes the problem slightly different from here, the objective is usually different, that is, minimising the time for a message to reach all nodes.

2 Preliminary definitions and results

A *weighted directed multigraph* (or just *weighted multidigraph*) $D = (V, E)$ consists of a set V of *nodes* and a set $E \subseteq V \times V \times \mathbb{R}^+$ of *(weighted) edges* (for each edge (u, v, w), we say that u is the *tail*, v is the *head*, and w is the weight of the edge). A *walk* T in a weighted multidigraph $D = (V, E)$ from a node u to a node v is a sequence $e_1, \ldots, e_k$ of edges in E such that, for each i with $i \in [k-1]$, the head of e_i is equal to the tail of e_{i+1}, u is the tail of e_1, and v is the head of e_k. The *duration* $\delta(T)$ is defined as the sum of the weights of all the edges of T. A node v is said to be *reachable* from a node u in D, if there exists a walk in D from u to v (in the following, we will assume that a node is reachable from itself).

A *trip network* is a weighted multidigraph $D = (V, E)$ (also called the *underlying multidigraph* of the trip network) along with a collection $\mathbb{T} = \{T_1, \ldots, T_{|\mathbb{T}|}\}$ of walks in D (also called *trips*).[3] In the following, without loss of generality, we will assume that any node in V and any edge in E appears in at least one trip in $\mathbb{T}$. Note that the disjoint union of the trips in $\mathbb{T}$ defines a weighted multidigraph M that we call the *induced multidigraph* of $(D, \mathbb{T})$ (we assume that the edges of M

[3]Here, the term "collection" is used to mean a multiset, that is, a set in which order is ignored but multiplicity is significant. The cardinality $|A|$ of a multiset A denotes the sum of the multiplicities of the distinct elements in A: in our case, $|\mathbb{T}|$ denotes the number of (not necessarily distinct) trips in the trip collection $\mathbb{T}$.

have an additional label specifying which trip they belong to, and usually represented in the figures by means of different line colors and styles).

A *temporal graph* $G = (V, \mathbb{E})$ consists of a set V of *nodes* and a set $\mathbb{E} \subseteq V \times V \times \mathbb{R} \times \mathbb{R}^+$ of *temporal edges*. Given a temporal edge $e = (u, v, t, \lambda)$, we say that u is the *tail*, v is the *head*, t is the *starting* time, $\lambda > 0$ is the *travel* time, and $t + \lambda$ is the *arrival* time of e. A *temporal path* from a node u to a node v in a temporal graph G is a sequence of temporal edges $e_1, \ldots, e_k$ such that the tail of e_1 is u, the head of e_k is v, and, for each $i \in [k-1]$, the tail of e_{i+1} is equal to the head of e_i and the starting time of e_{i+1} is at least equal to the arrival time of e_i (as travel times are strictly positive, the path is strict in the sense that the starting time of e_{i+1} is greater than the starting time of e_i, for each $i \in [k-1]$). A node v is *temporally reachable* from a node u if there exists a temporal path from u to v (in the following, we will assume that a node is temporally reachable from itself). The set of nodes temporally reachable from a node u in G is denoted as $\mathcal{R}_G(u)$. The *temporal reachability* of a node u in G is defined as $|\mathcal{R}_G(u)|$, while the *temporal reachability* of G is defined as $\sum_{u \in V} |\mathcal{R}_G(u)|$.

Given a trip network $(D, \mathbb{T})$, a *temporalisation* τ of the trip network assigns a real number $\tau(T)$ to each trip T in $\mathbb{T}$, indicating the starting time of T. Such a temporalisation induces the temporal graph $G[D, \mathbb{T}, \tau] = (V, \mathbb{E})$ defined as follows. For each $T = e_1, \ldots, e_k$ in $\mathbb{T}$, with $e_i = (u_i, v_i, w_i)$, $\mathbb{E}$ contains the temporal edges $(u_i, v_i, \tau(T) + \sum_{j=1}^{i-1} w_j, w_i)$: we say that these temporal edges are *induced* by T (with respect to the temporalisation τ). A node v is said to be τ-reachable from a node u if $v \in \mathcal{R}_{G[D,\mathbb{T},\tau]}(u)$. The *$\tau$-reachability* of a node u is the temporal reachability of u in $G[D, \mathbb{T}, \tau]$, and the *τ-reachability* of the trip network is the temporal reachability of $G[D, \mathbb{T}, \tau]$. Our main optimisation problem is the following one.

Maximum Reachability Trip Temporalisation (mrtt). Given a trip network $(D, \mathbb{T})$, find a temporalisation τ of the trip network which maximises its τ-reachability.

We will also study the restriction of the mrtt problem to the case in which, for each pair of nodes, there exists a temporalisation allowing to reach one from the other. More precisely, given a trip network $(D, \mathbb{T})$ and two nodes s and t, $(D, \mathbb{T})$ is said to be *(s, t)-temporalisable* if there exists a temporalisation τ of $(D, \mathbb{T})$ such that t is τ-reachable from s, and it is said to be *strongly temporalisable* if, for any two nodes s and t, $(D, \mathbb{T})$ is (s, t)-temporalisable. Moreover, we will also consider symmetric trip networks in the following sense. We say that a trip network $(D, \mathbb{T})$ is *symmetric* if all trips in $\mathbb{T}$ can be grouped into disjoint pairs $(T, \reflectbox{$T$})$ such that $\reflectbox{$T$}$ is the reverse of T (T and $\reflectbox{$T$}$ are two distinct trips in $\mathbb{T}$): $\reflectbox{$T$}$ visits the same nodes as T, but in reverse order.

We will often refer to a particular kind of temporalisations. Given a trip network $(D, \mathbb{T})$, a *schedule* of the trip network is an ordering of the trips in $\mathbb{T}$. Note that a schedule S immediately induces a temporalisation τ_S of the trip network defined as follows. If $S = T_1, \ldots, T_{|\mathbb{T}|}$, then $\tau_S(T_1) = 0$ and $\tau_S(T_{i+1}) = \sum_{j=1}^{i} \delta(T_j)$, for $i \in [|\mathbb{T}| - 1]$. A node v is said to be S-reachable from a node u if it is τ_S-reachable. The *S-reachability* of the trip network is defined as its τ_S-reachability (see Table 2 where, for any possible schedule S, we indicate the S-reachability of the trip network shown in Figure 2).

Fact 1 *Let $(D, \mathbb{T})$ be a trip network and S be a schedule of $(D, \mathbb{T})$. Let C be a weighted multidigraph obtained starting from D by arbitrarily modifying only the weights of the edges of D. The S-reachability of $(D, \mathbb{T})$ is equal to the S-reachability of $(C, \mathbb{T})$.*

	T_1, T_2, T_3	T_1, T_3, T_2	T_2, T_1, T_3	T_2, T_3, T_1	T_3, T_1, T_2	T_3, T_2, T_1
v_1	$V \setminus \{v_5\}$	$V \setminus \{v_5, v_8\}$	$\{v_1, v_2, v_3, v_4\}$	$\{v_1, v_2, v_3, v_4\}$	$V \setminus \{v_5, v_8\}$	$\{v_1, v_2, v_3, v_4\}$
v_2	$V \setminus \{v_1, v_5\}$	$V \setminus \{v_1, v_5, v_8\}$	$V \setminus \{v_1, v_5\}$	$V \setminus \{v_1, v_5\}$	$V \setminus \{v_1, v_5, v_8\}$	$V \setminus \{v_1, v_5, v_8\}$
v_3	$V \setminus \{v_1, v_5\}$	$V \setminus \{v_1, v_5, v_8\}$	$V \setminus \{v_1, v_5\}$	$V \setminus \{v_1, v_5\}$	$V \setminus \{v_1, v_5, v_8\}$	$V \setminus \{v_1, v_5, v_8\}$
v_5	$\{v_5, v_6, v_7, v_8\}$	$V \setminus \{v_1, v_3, v_4\}$	$\{v_5, v_6, v_7, v_8\}$	$\{v_5, v_6, v_7, v_8\}$	$V \setminus \{v_1, v_3, v_4\}$	$V \setminus \{v_1\}$
v_6	$\{v_2, v_6, v_7, v_8\}$	$\{v_2, v_6, v_7, v_8\}$	$\{v_2, v_6, v_7, v_8\}$	$V \setminus \{v_1, v_5\}$	$\{v_2, v_6, v_7, v_8\}$	$V \setminus \{v_1, v_5\}$
v_7	$\{v_2, v_7, v_8\}$	$\{v_2, v_7, v_8\}$	$V \setminus \{v_1, v_5, v_6\}$	$V \setminus \{v_1, v_5, v_6\}$	$\{v_2, v_7, v_8\}$	$V \setminus \{v_1, v_6, v_5\}$
	30	28	29	31	28	32

Table 2: The possible schedules of the trip network of Figure 2. For each schedule S and for each source node v, the corresponding cell shows the set of nodes S-reachable from v (the last row shows the value of the S-reachability). Note that in the underling multidigraph of the trip network the number of pairs of nodes u and v such that v is reachable from u is equal to 38.

Proof. The fact simply follows from the fact that a temporal path in $G[D, \mathbb{T}, \tau_S]$ is also a temporal path in $G[C, \mathbb{T}, \tau_S]$, since edges of different trips cannot be interleaved inside a temporal path obtained through a schedule, where all edges of a trip T are assigned smaller starting times than all edges of the trips scheduled after T. □

For the sake of simplicity and without loss of generality, in the following we will present our results by referring to trip networks in which the weight of all edges are equal to 1: indeed, as a consequence of Fact 1, *all* our results will apply to general trip networks as well (either because they are hardness results or because the lower bounds on the temporal reachability are obtained by referring to schedules). Under this assumption, a trip $T_i = (u_0, u_1, 1), \dots, (u_{k-1}, u_k, 1)$ will also be indicated as $T_i = \langle u_0, \dots, u_k \rangle$. Note, however, that, in general, the maximum τ-reachability obtainable through a temporalisation can be higher than the maximum S-reachability obtainable through a schedule (see, for example, Figure 4), and that the presence of weights can, in general, increase the maximum τ-reachability of a trip network (see, for example, Figure 5).

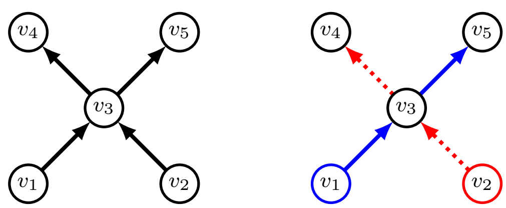

Figure 4: An example of a trip network $(D, \mathbb{T})$, where the underlying digraph D is depicted on the left (all edges have weight 1, so that D is a simple digraph) and $\mathbb{T}$ (depicted in the induced multidigraph on the right) contains the trips $T_1 = \langle v_1, v_3, v_5 \rangle$ (blue solid trip) and $T_2 = \langle v_2, v_3, v_4 \rangle$ (red dotted trip), such that the maximum τ-reachability obtainable through a temporalisation is higher than the maximum S-reachability obtainable through a schedule. Indeed, the two possible schedules both achieve a reachability equal to 12, while a temporalisation that assign the same starting time to T_1 and T_2 achieves a reachability equal to 13 (which is also the number of pairs of nodes u and v such that v is reachable from u in the underlying digraph).

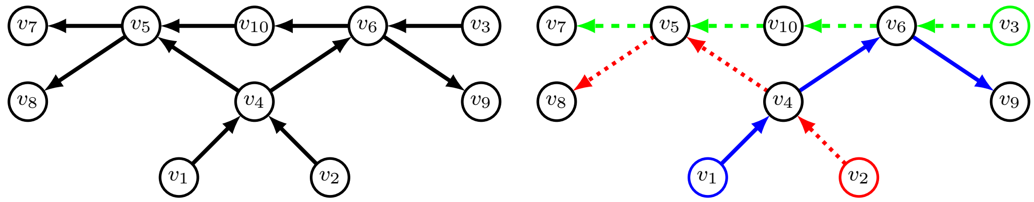

Figure 5: An example of a trip network $(D, \mathbb{T})$, where the underlying digraph D is depicted on the left and $\mathbb{T}$ contains the three trips (depicted on the right) T_1 (blue solid trip), T_2 (green dashed trip), and T_3 (red dotted trip), such that the presence of weights can increase the maximum τ-reachability obtainable through a temporalisation. Indeed, if all weights are equal to 1, no temporalisation τ can make the four nodes v_7, v_8, v_9, and v_{10} all τ-reachable from the three nodes v_1, v_2, and v_3: hence, for any temporalisation τ, the τ-reachability is less than the number R of pairs of nodes u and v such that v is reachable from u in D. On the contrary, if the edge from v_4 to v_5 has weight 3, then there exists a temporalisation whose reachability is equal to R (such a temporalisation assigns 1 to the trips T_1 and T_2, and 2 to T_3).

3 The maximum reachability trip temporalisation problem

We first consider the following one-to-one version of the MRTT problem, called ONE-TO-ONE REACHABILITY TRIP TEMPORALISATION (in short, O2O-RTT): given a trip network $(D, \mathbb{T})$ and two nodes s and t, is $(D, \mathbb{T})$ (s, t)-temporalisable? Quite surprisingly, even this restricted version of the MRTT problem seems to be difficult to be solved in polynomial time.

Theorem 1 *The* O2O-RTT *problem is NP-complete.*

Proof. We reduce in polynomial time 3-SAT to O2O-RTT. Let us consider a 3-SAT formula Φ, with n variables $x_1, \ldots, x_n$ and m clauses $c_1, \ldots, c_m$. We first define the directed graph $D = (V, E)$ as the union of the following gadgets.

Intermediate and final nodes. V contains two nodes v_{n+1} and w_{m+1}.

Variable gadgets (see Figure 6(a)). For each variable x_i with $i \in [n]$, let p_i be the number of clauses that contains the literal x_i, and n_i the number of clauses that contains the literal $\neg x_i$ (without loss of generality, we may assume that both p_i and n_i are positive numbers). Then, V contains the following $p_i + n_i + 1$ nodes: $v_i, f_i^1, \ldots, f_i^{p_i}, t_i^1, \ldots, t_i^{n_i}$. Moreover, E contains the following $p_i + n_i + 2$ directed edges: (v_i, f_i^1), (f_i^h, f_i^{h+1}) for $h \in [p_i - 1]$, $(f_i^{p_i}, v_{i+1})$, (v_i, t_i^1), (t_i^h, t_i^{h+1}) for $h \in [n_i - 1]$, and $(t_i^{n_i}, v_{i+1})$.

Clause gadgets (see Figure 6(b)). For each clause c_j with $j \in [m]$, V contains the following four nodes: w_j, l_j^1, l_j^2, l_j^3. Moreover, E contains the following six edges: (w_j, l_j^h) and (l_j^h, w_{j+1}), for $h \in [3]$.

Variable-clause edge. E contains the edge (v_{n+1}, w_1).

Clause-variable edges (see Figure 6(c)). For each clause c_j with $j \in [m]$, for each variable x_i with $i \in [n]$, for $h \in [3]$, and for $k \in [n_i]$, E contains the edge (l_j^h, t_i^k) if the h-th literal of c_j is $\neg x_i$ and c_j is the k-th clause in which the literal $\neg x_i$ occurs. Analogously, for each clause c_j with

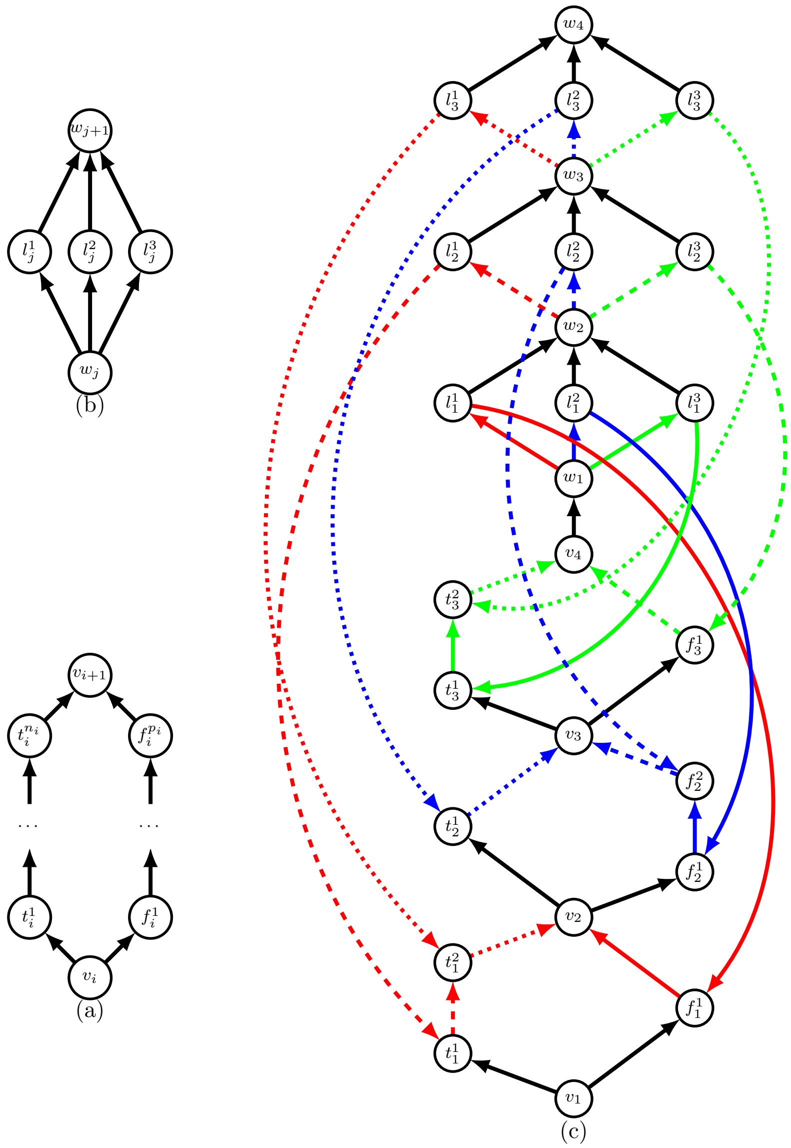

Figure 6: The reduction from 3-SAT to O2O-RTT. The variable gadget (a) corresponding to the variable x_i (p_i is the number of clauses that contains the literal x_i, while n_i is the number of clauses that contains the literal $\neg x_i$), the clause gadget (b) corresponding to the clause c_j, and the trip network (c) corresponding to the 3-SAT formula $(x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_3)$ (the trips in red, blue, and green are the three trips corresponding to the first clause (solid edges), to the second clause (dashed edges), and to the third clause (dotted edges), and each black edge forms a trip of length one).

$j \in [m]$, for each variable x_i with $i \in [n]$, for $h \in [3]$, and for $k \in [p_i]$, E contains the edge (l_j^h, f_i^k) if the h-th literal of c_j is x_i and c_j is the k-th clause in which the literal x_i occurs.

We now define the trip collection $\mathbb{T}$ on D. For each clause c_j with $j \in [m]$ and for $h \in [3]$, $\mathbb{T}$ contains the trip $\langle w_j, l_j^h, t_i^k, o_i^k \rangle$, if $(l_j^h, t_i^k) \in E$ and o_i^k is defined as the unique out-neighbour of t_i^k (that is, $o_i^k = t_i^{k+1}$ if $k < n_i$, and $o_i^k = v_{i+1}$ if $k = n_i$), and the trip $\langle w_j, l_j^h, f_i^k, o_i^k \rangle$, if $(l_j^h, f_i^k) \in E$

and o_i^k is defined as the unique out-neighbour of f_i^k (that is, $o_i^k = f_i^{k+1}$ if $k < p_i$, and $o_i^k = v_{i+1}$ if $k = p_i$). Each of the other $2n + 3m + 1$ edges, that are not yet included in a trip, forms a one-edge trip. Figure 6(c) shows an example of the reduction in the case of the Boolean formula $(x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_3)$.

Let τ be a temporalisation of the trip network $(D, \mathbb{T})$ and let $G = G[D, \mathbb{T}, \tau]$ be the temporal graph induced by τ. Note that, each edge in D belongs to exactly one trip in $\mathbb{T}$, which means that, for each edge $e \in E$, there is exactly one temporal edge in G with the same head and tail of e. Note also that, due to the topology of D, if $w_{m+1} \in \mathcal{R}_G(v_1)$, then the first part of the temporal path P from v_1 to w_{m+1} consists in moving from v_1 to v_{n+1} by passing, for each $i \in [n]$, through the node v_i and either through the nodes $t_i^1, \ldots, t_i^{n_i}$ or through the nodes $f_i^1, \ldots, f_i^{p_i}$. The second part of the temporal path P consists in moving from v_{n+1} to w_1 and, then, from w_1 to w_{m+1} by passing, for each $j \in [m]$, through the node w_j and exactly one l_j-node. Indeed, we can assume that P does not go back from a l_j-node to a variable node to which it is connected, since otherwise P should have to pass again through the edge (v_{n+1}, w_1), contradicting the fact that, as observed above, there is only one temporal edge in G corresponding to this edge. Moreover, if P uses a temporal edge with tail w_j and head l_j^h, for some $h \in [3]$, and if $(l_j^h, t_i^k) \in E$ (respectively, $(l_j^h, f_i^k) \in E$), for some $i \in [n]$ and $k \in [n_i]$ (respectively, $k \in [p_i]$), then P must have passed, in its first part, through the f_i-nodes (respectively, t_i-nodes) corresponding to the variable x_i. Otherwise, as P is a temporal path, the edge outgoing t_i^k (respectively, f_i^k) with head o_i^k would have an appearing time smaller than that of (w_j, l_j^h), contradicting the fact that τ is a temporalisation of the trip $\langle w_j, l_j^h, t_i^k, o_i^k \rangle$ (respectively, $\langle w_j, l_j^h, f_i^k, o_i^k \rangle$).

Let us now prove that $(D, \mathbb{T})$ is (v_1, w_{m+1})-temporalisable if and only if there exists an assignment α to the variables that satisfies the Boolean formula Φ. Let us first suppose that $(D, \mathbb{T})$ is (v_1, w_{m+1})-temporalisable, that is, there exists a temporalisation τ of $(D, \mathbb{T})$ such that $w_{m+1} \in \mathcal{R}_G(v_1)$, where $G = G[D, \mathbb{T}, \tau]$ is the temporal graph induced by τ. Let P be a temporal path from v_1 to w_{m+1} in G. For each variable x_i with $i \in [n]$, we set $\alpha(x_i) = \text{TRUE}$ if and only if P passes through the t_i-nodes corresponding to x_i. We now prove that any clause c_j, with $j \in [m]$, is satisfied by α. Let l_j^h, for some $h \in [3]$, be the node which P goes to, when moving from w_j. If $(l_j^h, f_i^k) \in E$ (respectively, $(l_j^h, t_i^k) \in E$), for some $i \in [n]$ and $k \in [p_i]$ (respectively, $k \in [n_i]$), then the h-th literal of c_j is x_i (respectively, $\neg x_i$), and, because of the previous observations, P passes through the t_i-nodes (respectively, f_i-nodes) corresponding to x_i: this implies that $\alpha(x_i) = \text{TRUE}$ (respectively, $\alpha(x_i) = \text{FALSE}$) and, hence, that the clause c_j is satisfied.

Let us now suppose that there exists an assignment α to the variables that satisfies the formula Φ, and let us consider the following walk P in D, which starts from v_1 and arrives in w_{m+1}. The first part of P arrives at v_{n+1} and consists in moving from v_i to v_{i+1}, for $i \in [n]$, by passing through the t_i-nodes (respectively, f_i-nodes) corresponding to x_i, if $\alpha(x_i) = \text{TRUE}$ (respectively, $\alpha(x_i) = \text{FALSE}$). We know that, for each clause c_j with $j \in [m]$, at least one literal of c_j is satisfied: suppose that the first such literal is the h_j-th one, for some $h_j \in [3]$. Note that the choice of h_j implies that the first part of P does not use any edge of the trip T containing the edge $(c_j, l_j^{h_j})$: indeed, if the corresponding literal is x_i (respectively, $\neg x_i$), T goes through the f_i-nodes (respectively, t_i-nodes), while P goes through the t_i-nodes (respectively, f_i-nodes) as $\alpha(x_i) = \text{TRUE}$ (respectively, $\alpha(x_i) = \text{FALSE}$). The second part of P starts from v_{n+1}, moves to w_1, arrives at w_{m+1}, and consists in moving from w_j to w_{j+1}, for $j \in [m]$, by passing through the node $l_j^{h_j}$. Because of the definition of $\mathbb{T}$ and the choice of $h_1, \ldots, h_m$, each edge of P belongs to a different trip in $\mathbb{T}$. We can then consider a schedule S of $(D, \mathbb{T})$ in which the trips corresponding to the edges in P are scheduled in the same order as they appear in P itself. The walk P thus induces a temporal path in $G[D, \mathbb{T}, \tau_S]$ from v_1 to w_{m+1}. Thus, $(D, \mathbb{T})$ is (s,t)-temporalisable and the theorem has been proved. □

Note that, as a consequence of its proof, the above theorem holds even with the following restrictions: the in-degree and the out-degree of the nodes in D are bounded by 3, and $\mathbb{T}$ contains only *simple* trips (that is, trips that do not pass through the same node more than once), which are pairwise edge-disjoint.

We are now ready to prove the inapproximability result of the MRTT problem. Note that, given an instance $(D, \mathbb{T})$ with n nodes, any solution of the problem has a value greater than or equal to n (since we have assumed that any node is temporally reachable from itself). Since the number of pairs of nodes is n^2, the MRTT problem is trivially n-approximable. The following theorem states that a better approximation is not achievable, unless P = NP.

Theorem 2 *Unless* P = NP*, the* MRTT *problem is not $r(\cdot)$-approximable, for any $\epsilon \in (0,1)$ and for any non-decreasing function r in $O(n^{1-\epsilon})$, where n is the number of nodes.*

Proof. The proof makes use of the well-known gap technique (see Section 3.1.4 of [3]). Suppose by contradiction that there exists a $r(\cdot)$-approximation algorithm $\mathcal{A}$ for the MRTT problem, for some $\epsilon \in (0,1)$ and for some function $r(n)$ of the number n of nodes that satisfies $r(n) \leq cn^{1-\epsilon}$ for some constant c. We will now show that it is possible to exploit such an algorithm in order to solve in polynomial time the O2O-RTT problem, which would imply that P = NP (because of Theorem 1). Let us consider an instance $\langle (D = (V,E), \mathbb{T}), s, t \rangle$ of the O2O-RTT problem, where $V = \{s = v_1, \ldots, v_n = t\}$. Without loss of generality, we assume that $n > c+1$. We define an instance $(D' = (V', E'), \mathbb{T}')$ of MRTT as follows.

- $V' = V \cup \{v_{n+i} : i \in [2K]\}$ with $K = \left\lceil (cn)^{1/\epsilon}(n+2)^{\frac{2-\epsilon}{\epsilon}} \right\rceil$.
- $E' = E \cup \{(v_{n+i}, s), (t, v_{n+K+i}) : i \in [K]\}$.
- $\mathbb{T}'$ is the union of $\mathbb{T}$ with all the one-edge trips corresponding to the edges in $E' \setminus E$.

Consider an optimal temporalisation τ^* of $(D', \mathbb{T}')$: the maximum reachability is thus $\texttt{opt} = \sum_{u \in V} |\mathcal{R}_{G[D', \mathbb{T}', \tau^*]}(u)|$. Moreover, let x be the value of the reachability achieved by the temporalisation computed by the approximation algorithm $\mathcal{A}$ with input $(D', \mathbb{T}')$: hence, $\frac{\texttt{opt}}{r(n')} \leq x \leq \texttt{opt}$ where $n' = n + 2K$.

Let us upper bound $\texttt{opt}$ in the case in which $(D, \mathbb{T})$ is not (s,t)-temporalisable. To this aim, we upper bound the number of nodes τ'-reachable from each node in V', for any temporalisation τ' of $(D', \mathbb{T}')$ (in the following, $G' = G[D', \mathbb{T}', \tau']$ is the temporal graph induced by τ').

- $|\mathcal{R}_{G'}(v_1)| \leq n - 1$ (since t is not τ'-reachable from $s = v_1$).
- For each $i \in [n-1]$, $|\mathcal{R}_{G'}(v_{i+1})| \leq n + K$ (since all nodes v_{n+i} with $i \in [K]$ have in-degree equal to zero in D').
- For each $i \in [K]$, $|\mathcal{R}_{G'}(v_{n+i})| \leq n$ (since t is not τ'-reachable from s).

- For each $i \in [K]$, $|\mathcal{R}_{G'}(v_{n+K+i})| = 1$ (since all these nodes have out-degree equal to zero in D').

Thus, if $(D, \mathbb{T})$ is not (s,t)-temporalisable, we have that $x \leq \texttt{opt} \leq (n-1) + (n-1)(n+K) + Kn + K = n^2 + 2nK - 1$. On the other hand, if $(D, \mathbb{T})$ is (s,t)-temporalisable, then it is easy to produce a temporalisation τ' of $(D', \mathbb{T}')$ such that $v_{n+K+i} \in \mathcal{R}_{G[D',\mathbb{T}',\tau']}(v_{n+j})$, for any $i, j \in [K]$. Hence, in this case we have that $x \geq \frac{\texttt{opt}}{r(n')} \geq \frac{K^2}{r(n')}$.

If we prove that $\frac{K^2}{r(n')} > n^2 + 2nK - 1$, then we have that $x > n^2 + 2nK - 1$ if and only if $(D, \mathbb{T})$ is (s,t)-temporalisable. This would imply that the O2O-RTT problem is solvable in polynomial time, and the theorem is proved. Let us then show that $\frac{K^2}{r(n')} > n^2 + 2nK - 1$. Since

$$\frac{K^2}{r(n')} \geq \frac{K^2}{c(n+2K)^{1-\epsilon}} > \frac{K^2}{c(n+2)^{1-\epsilon}K^{1-\epsilon}} = \frac{K^{1+\epsilon}}{c(n+2)^{1-\epsilon}},$$

it is sufficient to prove that $K^{1+\epsilon} \geq c(n+2)^{1-\epsilon}n(n+2K)$. Since, by definition, $K \geq (cn)^{1/\epsilon}(n+2)^{\frac{2-\epsilon}{\epsilon}}$, that is, $K^\epsilon \geq cn(n+2)^{2-\epsilon}$, we have that $K^{1+\epsilon} \geq cnK(n+2)^{2-\epsilon} > cn(n+2K)(n+2)^{1-\epsilon}$, and the proof is completed. □

We can state a third hardness result (whose proof is given in the appendix), concerning an optimisation problem which is, somehow, in between the O2O-RTT problem and the MRTT problem, that is, the following SINGLE SOURCE MAXIMUM REACHABILITY TRIP TEMPORALISATION (in short, SS-MRTT) problem: given a trip network $(D, \mathbb{T})$ and a node s, find a temporalisation τ of $(D, \mathbb{T})$ which maximises the τ-reachability of s. Note that, given an instance of this problem, any solution has a value greater than or equal to 1. Since the maximum number of nodes temporally reachable from s is n, we have that the SS-MRTT problem is trivially n-approximable. The following theorem states that a better approximation is not achievable, unless P = NP.

Theorem 3 *Unless* P = NP*, the* SS-MRTT *problem is not $r(\cdot)$-approximable, for any $\epsilon \in (0,1)$ and for any non-decreasing function r in $O(n^{1-\epsilon})$, where n is the number of nodes.*

3.1 Bounding the number of used trips

In this section, we study the O2O-RTT problem parameterised by the number of trips needed to go from the source to the destination.

Given a trip network $(D, \mathbb{T})$, a temporalisation τ and $k \in \mathbb{N}$, a node v is said to be (k,τ)*-reachable* from u if there exists a temporal path P in $G = G[D, \mathbb{T}, \tau]$ from u to v which is composed of edges induced by at most k different trips in $\mathbb{T}$: let $\mathbb{P}$ be the set of such trips. Note that, without loss of generality, we can suppose that the edges induced by the same trip are contiguous in P. Indeed, if $P = e_1, \ldots, e_p$, let $T_1 \in \mathbb{P}$ be one of the trips that induces e_1 and let e_h be the last edge in P which is induced by T_1. The temporalisation of T_1 induces a temporal path $e'_1 = e_1, e'_2, \ldots, e'_{l-1}, e'_l = e_h$ in G. We can then consider the temporal path $P' = e'_1, \ldots, e'_l, e_{h+1}, \ldots, e_p$, which has the property that all the edges which are induced by T_1 are contiguous. We can now apply the same argument by considering the trip $T_2 \in \mathbb{P}$ as one of the trips that induces e_{h+1}, and go on like this until all the edges induced by each trip considered are contiguous in the final temporal path from u to v.

We now consider the O2O-RTT problem parameterised by the number k of trips used in the resulting temporal graph, in order to go from s to t. More precisely, given a trip network $(D, \mathbb{T})$, a source node s, and a target node t, the parameterised problem k-O2O-RTT consists in deciding whether there exists a temporalisation τ such that v is (k,τ)-reachable from u. By using the color coding technique developed in [2], we can obtain the following result.

Theorem 4 *The k-O2O-RTT problem can be solved in $2^{O(k)} m \log |\mathbb{T}|$ time where $m = \sum_{T \in \mathbb{T}} |T|$ is the sum of trip lengths.*

Proof. Let us consider a trip network $(D=(V,E),\mathbb{T})$, a source node s and a target node t, and let M be the induced multidigraph of $(D,\mathbb{T})$. Note that m is also the number of edges in M. We suppose that for each trip T, we are given the list of its edges in their respective order in T. We also let $V(T)$ denote the set of nodes appearing in T and write $u \prec_T v$ when $u, v \in V(T)$ and u precedes v in T. Let $\chi : \mathbb{T} \to [k]$ be any color assignment to the trips in $\mathbb{T}$. For $i \in [k]$, a *(i,χ)-path* P in M is a path which is the concatenation of exactly i subtrips of distinct trips in $\mathbb{T}$ with pairwise distinct colors (in the following, $\chi(P)$ will denote the set of colors "used" by such a path P). Note that the existence of a (i,χ)-path in M from s to t with $i \in [k]$ implies the existence of a schedule S such that t is (k,τ_S)-reachable from s: indeed, we can first schedule the i trips of the path in the order they appear in it, and then the remaining trips of $\mathbb{T}$ in any order. In order to test the existence of a *(i,χ)-path* in M from s to t for $i \in [k]$, we can use the dynamic programming technique [4]. For any node $v \in V$, let $A_\chi[v,i]$ denote the collection of sets C of colors for which there exists a (i,χ)-path P in M from s to v such that $\chi(P) = C$. Clearly, there exists a *(i,χ)-path* in M from s to t if and only if $A_\chi[t,i] \neq \emptyset$. We have that, for any node $v \in V$,

$$A_\chi[v,1] = \{\{\chi(T)\} : T \in \mathbb{T} \wedge s \prec_T v\}.$$

Moreover, for any $i \in [k-1]$,

$$\begin{aligned} A_\chi[v,i+1] &= \bigcup_{T \in \mathbb{T} : v \in V(\mathbb{T})} \{C \cup \{\chi(T)\} : u \prec_T v \wedge C \in A_\chi[u,i] \wedge \chi(T) \notin C\} \\ &= \bigcup_{T \in \mathbb{T} : v \in V(\mathbb{T})} \{C \cup \{\chi(T)\} : C \in \mathcal{C}(T,v) \wedge \chi(T) \notin C\} \end{aligned}$$

where $\mathcal{C}(T,v) = \bigcup_{u \prec_T v} A_\chi[u,i]$. We can compute $A_\chi[v,1]$ for all $v \in V$ by scanning all trips that contain s. These trips can be obtained in $O(m)$ time by scanning all trips in $\mathbb{T}$. Moreover, we can execute the above update rule for all $v \in V$ by scanning once each trip $T \in \mathbb{T}$ as follows. We first set $A_\chi[v,i+1] := \emptyset$ for all $v \in V$. Then, for each trip T, we iterate over the edges of T in their respective order in T. For each edge (u,v) of T, we compute $\mathcal{C}(T,v) = \mathcal{C}(T,u) \cup A_\chi[u,i]$ and update $A_\chi[v,i+1] := A_\chi[v,i+1] \cup \{C \cup \{\chi(T)\} : C \in \mathcal{C}(T,v) \wedge \chi(T) \notin C\}$ (if (u,v) is the first edge of T we simply use $\mathcal{C}(T,v) = A_\chi[u,i]$ as u is then the only node preceding v in T). Note that both the computation of $\mathcal{C}(T,v)$ and the update of $A_\chi[v,i+1]$ take $O(2^k)$ time since, for any node $u \in V$ and for $j \in [k]$, $|A_\chi[u,j]| \leq 2^k$. Each update step is thus performed in $O(2^k m)$ time and the whole computation requires $2^{O(k)} m$ time.

Observe now that if there exists a temporalisation of $(D,\mathbb{T})$ such that t is (k,τ)-reachable from s, then there must exist a color assignment χ such that M includes a (i,χ)-path from s to t for some $i \in [k]$. In order to find such a path, we can use an appropriate set of perfect hash functions from $[|\mathbb{T}|]$ to $[k]$. Indeed, it is possible to design $2^{O(k)} \log |\mathbb{T}|$ hash functions such that any subset of k trips has image $\{1,\ldots,k\}$ for at least one function [38]. This implies that any subset of i trips has i pairwise distinct colors as image for at least one function as such a set can be completed in a set of k trips. Each hash function can be coded with $O(k + \log\log|\mathbb{T}|)$ bits, and can be generated in $O(k^3 \log|\mathbb{T}|)$ time [20]. The number of such functions is $O(2^k \log|\mathbb{T}|)$ and their computation takes $2^{O(k)} \log^2|\mathbb{T}|$ time. As they can be accessed in $O(1)$ time, testing the coloring obtained through each function yields a $2^{O(k)} m \log|\mathbb{T}|$-time algorithm for solving the k-O2O-RTT problem (we use $\log|\mathbb{T}| = O(|\mathbb{T}|)$ and $|\mathbb{T}| \leq m$). The theorem is thus proved. □

4 Strongly temporalisable trip networks

We now switch to strongly temporalisable trip networks where one-to-one reachability is assumed for all pairs of nodes. This clearly implies that the O2O-RTT problem is trivially solvable when restricted to strongly temporalisable trip networks, since the answer is always yes (actually, one-to-one reachability is always satisfied under strong temporalisability). On the other hand, we will prove that both the MRTT and the SS-MRTT problem cannot be approximated within a factor less than $\frac{\sqrt{n}}{12}$ (unless P = NP). To this aim, we first show that the strong temporalisability by itself is not enough to ensure the existence of a temporalisation τ with a τ-reachability which is a constant fraction of all pairs of nodes.

Theorem 5 *For any $r > 3$, there exists a strongly temporalisable trip network $(D_r, \mathbb{T}_r)$ with $n = r^2 + 2r$ nodes, such that any temporalisation τ of $(D_r, \mathbb{T}_r)$ has τ-reachability $O(n^{\frac{3}{2}})$.*

Proof. We first define the trip network $(D_r, \mathbb{T}_r)$ through the gadgets that compose it (see Figure 7). We then prove that the trip network is strongly temporalisable and, finally, we prove that, for any temporalisation τ, the τ-reachability is $O(n^{\frac{3}{2}})$.

Upper gadget $V^U, E^U, \mathbb{T}^U$. The set V^U contains the nodes $c_1, \ldots, c_r$, and the nodes $u_1, \ldots, u_r$. These nodes are connected through the following set of directed edges:

$$E^U = \{(c_{i+1}, c_i) : i \in [r-1]\} \cup \{(c_i, c_{i+2}) : i \in [r-2]\} \cup \{(c_1, u_i), (u_i, c_2) : i \in [r]\}.$$

On this gadget, we have the following collection of trips: $\mathbb{T}^U = \{T_i^U : i \in [r]\}$, where

$$T_i^U = \langle c_1, u_i, c_2, c_1, c_3, c_2, \ldots, c_{r-1}, c_{r-2}, c_r, c_{r-1} \rangle$$

(each edge (c_{i+1}, c_i) is followed by (c_i, c_{i+2}), see also Figure 7, where the upper red solid trip is T_3^U).

Lower gadget $V^L, E^L, \mathbb{T}^L$. The set V^L contains the nodes $c_{r+1}, \ldots, c_{2r}$, and the nodes $l_1, \ldots, l_r$. These nodes are connected through the following set of directed edges:

$$\begin{aligned} E^L &= \{(c_{r+i+1}, c_{r+i}) : i \in [r-1]\} \cup \{(c_{r+i}, c_{r+i+2}) : i \in [r-2]\} \\ &\quad \cup \{(c_{2r-1}, l_i), (l_i, c_{2r}) : i \in [r]\}. \end{aligned}$$

On this gadget, we have the following collection of trips: $\mathbb{T}^L = \{T_i^L : i \in [r]\}$, where

$$T_i^L = \langle c_{r+2}, c_{r+1}, c_{r+3}, c_{r+2}, \ldots, c_{2r-2}, c_{2r}, c_{2r-1}, l_i, c_{2r} \rangle$$

(each edge (c_{r+i+1}, c_{r+i}) is followed by (c_{r+i}, c_{r+i+2}), see also Figure 7, where the lower red solid trip is T_3^L).

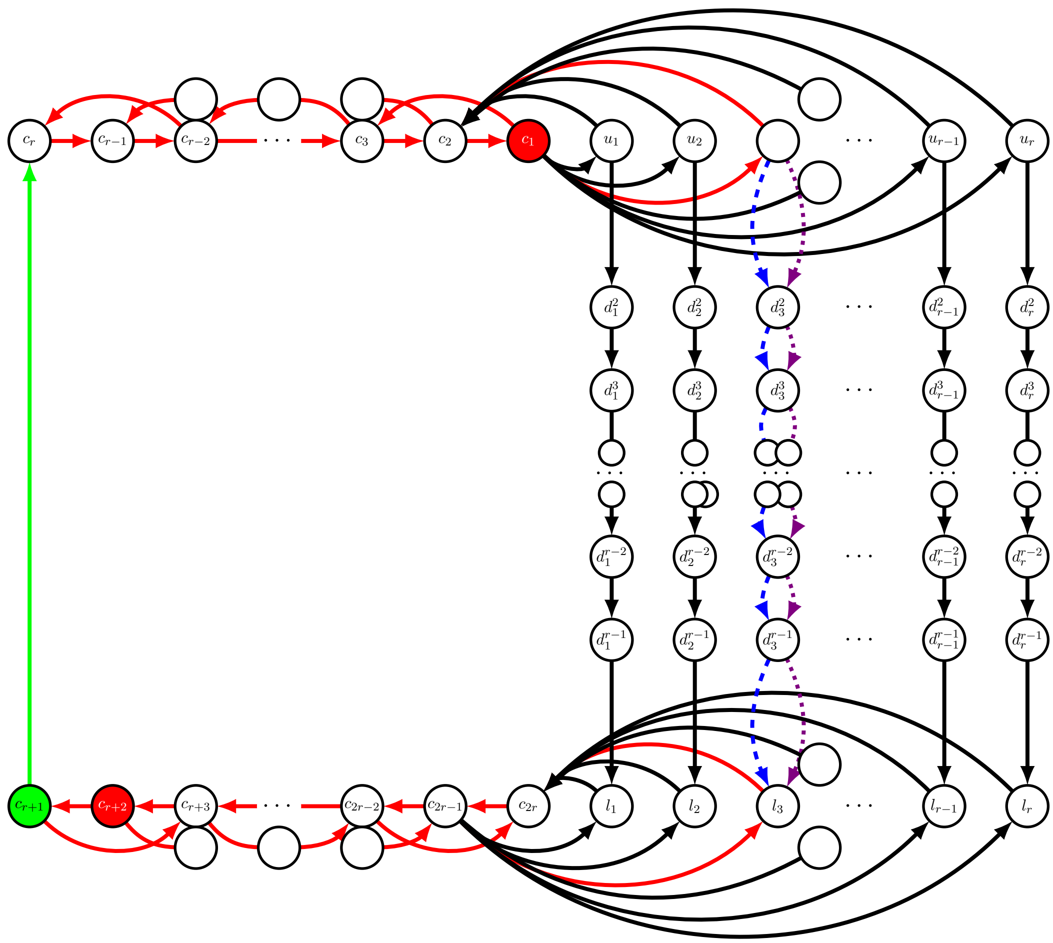

Figure 7: An example of a strongly temporalisable trip network, such that any temporalisation cannot connect a constant fraction of the total pairs of nodes, obtained via the construction described in the proof of Theorem 5. The two red solid trips, starting from the two red nodes, correspond to the trip T_3^U and T_3^L, respectively, in the construction (the first time a trip passes through a node with more than one red outgoing edge, it continues towards the node with the smaller index). The blue dashed (respectively, violet dotted) trip, starting from the half blue (respectively, violet) node, corresponds to the trip $T_3^{\downarrow \mathrm{l}}$ (respectively, $T_3^{\downarrow \mathrm{r}}$) in the construction. Finally, the green solid trip, starting from the green node, corresponds to the trip $T^{\uparrow}$ in the construction.

Descending gadgets $\boldsymbol{V_i^{\downarrow}, E_i^{\downarrow}}, \mathbb{T}^{\downarrow}$. For any i with $i \in [r]$, we refer to node u_i and l_i as d_i^1 and d_i^r, respectively. The set $V_i^{\downarrow}$ contains the nodes $d_i^2, \ldots, d_i^{r-1}$. These nodes are connected among them and to the previous gadgets through the following set of directed edges:

$$E_i^{\downarrow} = \{(d_i^j, d_i^{j+1}) : j \in [r-1]\}.$$

On this gadget, we have the following collection of trips: $\mathbb{T}^{\downarrow} = \{T_i^{\downarrow \mathrm{l}}, T_i^{\downarrow \mathrm{r}} : i \in [r]\}$, where

$$T_i^{\downarrow \mathrm{l}} = T_i^{\downarrow \mathrm{r}} = \langle u_i = d_i^1, d_i^2, \ldots, d_i^{r-1}, d_i^r = l_i \rangle$$

(see Figure 7, where the blue dashed trip is $T_3^{\downarrow \mathrm{l}}$ and the violet dotted trip is $T_3^{\downarrow \mathrm{r}}$).

Ascending gadget $e^{\uparrow}$, $T^{\uparrow}$. This gadget contains the edge $e^{\uparrow} = (c_{r+1}, c_r)$, which connects the lower gadget to the upper gadget, and is also a one-edge trip $T^{\uparrow}$. Note that this gadget does not introduce any new nodes.

To conclude the definition of the network, we set $D_r = (V, E)$ where

$$V = V^U \cup V^L \cup \bigcup_{i=1}^{r} V_i^{\downarrow}$$

(note that $|V| = 2r + 2r + r(r-2) = r^2 + 2r$) and

$$E = E^U \cup E^L \cup \bigcup_{i=1}^{r} E_i^{\downarrow} \cup \{e^{\uparrow}\},$$

and we set

$$\mathbb{T}_r = \mathbb{T}^U \cup \mathbb{T}^L \cup \mathbb{T}^{\downarrow} \cup \{T^{\uparrow}\}.$$

Note that the descending gadgets contain the majority of the nodes in the trip network, and that it is possible to visit them by entering from the upper nodes and by travelling all the way down (the second descending trip $T_i^{\downarrow \mathrm{r}}$ is necessary in order to ensure that the trip network is strongly temporalisable).

$(D_r, \mathbb{T}_r)$ is strongly temporalisable. In order to ease the reading of the proof let us first define the following r (partial) schedules of the trip networks $(D_r, \mathbb{T}^U)$ and $(D_r, \mathbb{T}^L)$, respectively.

- Schedule S_i^U for $i \in [r]$. This schedule has the purpose of making u_i S_i^U-reachable from c_r. The schedule S_i^U consists in having the trips $\mathbb{T}^U$ scheduled in any order that has T_i^U as the last one. Starting from c_r, it is possible to go to c_{r-1} through the edge (c_r, c_{r-1}) of the first trip scheduled, to c_{r-2} through the edge (c_{r-1}, c_{r-2}) of the second trip scheduled, and so on. In this way, it is possible to reach c_1 using the first $r-1$ trips scheduled. Finally, from c_1 it is possible to go to u_i through the edge (c_1, u_i) of the trip T_i^U. Note that this also shows that, for any $h, k \in [r]$ with $h > k$, c_k is S_i^U-reachable from c_h.

- Schedule S_i^L for $i \in [r]$. This schedule is similar to the previous one, but applied to the lower gadget. It allows node c_{r+1} to be S_i^L-reachable from l_i. The schedule S_i^L consists in having the trips $\mathbb{T}^L$ scheduled in any order that has T_i^L as the first one. Starting from l_i, it is possible to go to c_{2r} through the edge (l_i, c_{2r}) of the trip T_i^L. At this point, it is possible to reach c_{r+1} from c_{2r} by using one edge of each of the remaining $r-1$ trips. Note that this also shows that, for any $h, k \in [r]$ with $h > k$, c_{k+r} is S_i^L-reachable from c_{h+r}.

By using the (partial) schedules above, we can now easily show that, for any two nodes u and v in V, there exists a schedule S such that v is S-reachable from u. These schedules are specified in Table 3 for each possible pair of nodes. For example, in order to reach $d_h^{l_2} \in V^{\downarrow}$ from $d_h^{l_1} \in V^{\downarrow}$ with $l_2 < l_1$, we first schedule the trip $T_h^{\downarrow \mathrm{l}}$ in order to reach $l_h \in V^L$, we then use the (partial) schedule S_h^L in order to reach c_{r+1}, we then schedule the trip $T^{\uparrow}$ in order to reach c_r, we then use

<table>
<tr><th></th><th colspan="5">Destination</th></tr>
<tr><th>Source</th><th>$c_k \in V^U$</th><th>$u_k \in V^U$</th><th>$c_k \in V^L$</th><th>$l_k \in V^L$</th><th>$d_k^{l_2} \in V_k^{\downarrow\mathrm{l}}$</th></tr>
<tr><td>$c_h \in V^U$</td><td>S_1^U if $k < h$
T_1^U if $k > h$</td><td>S_k^U</td><td>$S_1^U, T_1^{\downarrow\mathrm{l}}, S_1^L$</td><td colspan="2">$S_k^U, T_k^{\downarrow\mathrm{l}}$</td></tr>
<tr><td>$u_h \in V^U$</td><td>T_h^U</td><td>T_h^U, T_k^U</td><td>$T_h^{\downarrow\mathrm{l}}, S_h^L$</td><td colspan="2">$T_k^{\downarrow\mathrm{l}}$ if $k = h$
$T_h^U, T_k^U, T_k^{\downarrow\mathrm{l}}$ if $k \neq h$</td></tr>
<tr><td>$c_h \in V^L$</td><td>$S_1^L, T^{\uparrow}, S_1^U$</td><td>$S_1^L, T^{\uparrow}, S_k^U$</td><td>$S_1^U$ if $k < h$
T_1^L if $k > h$</td><td>T_k^L</td><td rowspan="2">$S_h^L, T^{\uparrow}, S_k^U, T_k^{\downarrow\mathrm{l}}$</td></tr>
<tr><td>$l_h \in V^L$</td><td colspan="2">$S_h^U, T^{\uparrow}, S_k^U$</td><td>$S_h^L$</td><td>$T_h^L, T_k^L$</td></tr>
<tr><td>$d_h^{l_1} \in V_h^{\downarrow\mathrm{l}}$</td><td colspan="2">$T_h^{\downarrow\mathrm{l}}, S_h^L, T^{\uparrow}, S_k^U$</td><td>$T_h^{\downarrow\mathrm{l}}, S_h^L$</td><td>$T_h^{\downarrow\mathrm{l}}$ if $k = h$
$T_h^{\downarrow\mathrm{l}}, T_h^L, T_k^L$ if $k \neq h$</td><td>$T_h^{\downarrow\mathrm{l}}, S_h^L, T^{\uparrow}, S_k^U, T_k^{\downarrow\mathrm{l}}$ if $k \neq h$
$T_h^{\downarrow\mathrm{l}}$ if $k = h$ and $l_2 > l_1$
$T_h^{\downarrow\mathrm{l}}, S_h^L, T^{\uparrow}, S_k^U, T_k^{\downarrow\mathrm{r}}$ if $k = h$ and $l_2 < l_1$</td></tr>
</table>

Table 3: The different cases in the proof that the trip network $(D_r, \mathbb{T}_r)$, defined in the proof of Theorem 5 and illustrated in Figure 7, is strongly temporalisable. For each node u labeling the row and for each node v labeling the column, the corresponding cell specifies which (partial) schedule S has to be used in order to guarantee that v is S-reachable from u (the trips that do not appear can be scheduled in any order).

the (partial) schedule S_h^U in order to reach u_h, and we finally schedule the trip $T_h^{\downarrow\mathrm{r}}$ to reach $d_h^{l_2}$ (note how, in this case, we need the second descending trip).

Any temporalisation τ has $O(n\sqrt{n})$ τ-reachability. Given any temporalisation τ of the trip network $(D_r, \mathbb{T}_r)$, let $T_{i_{\min}}^L$ be one of the trips with minimum starting time according to τ among all the trips in the lower gadget, and let $T_{i_{\max}}^U$ be one of the trips with maximum starting time according to τ among all the trips in the upper gadget. We will prove the following claim.

Claim 1 *For any pair of nodes $(d_{h_1}^{l_1}, d_{h_2}^{l_2})$ with $1 < l_1, l_2 < r$, $h_1, h_2 \in [r]$, $h_1 \neq h_2$, and $h_1 \neq i_{\min} \vee h_2 \neq i_{\max}$, $d_{h_2}^{l_2}$ is not τ-reachable from $d_{h_1}^{l_1}$.*

Note that the number of pairs of nodes satisfying the conditions in the claim is at least $(r-1)(r-2)^3 > (r-1)(r^3 - 6r^2) = r^4 - 7r^3 + 6r^2 > r^4 - 7r^3$, thus implying that, since $n = r^2 + 2r$, the τ-reachability is at most $(r^2+2r)^2 - (r^4 - 7r^3) = r^4 + 4r^3 + 4r^2 - r^4 + 7r^3 = 11r^3 + 4r^2 < 15r^3$. Since $n = r^2 + 2r$, we have that $r < \sqrt{n}$, and that the τ-reachability is at most $15n\sqrt{n}$. The theorem thus follows.

It thus remains to prove the claim. To this aim, let $(d_{h_1}^{l_1}, d_{h_2}^{l_2})$ be such that $1 < l_1, l_2 < r$, $h_1, h_2 \in [r]$, $h_1 \neq h_2$, and $h_1 \neq i_{\min} \vee h_2 \neq i_{\max}$. Note that, since $h_1 \neq h_2$, each walk in D_r from $d_{h_1}^{l_1}$ to $d_{h_2}^{l_2}$ has to pass through the nodes l_{h_1}, c_{r+1}, c_r, and u_{h_2} in this order. Note also that any walk from l_{h_1} to c_{r+1} contains at least r edges. Since travelling on more than one edge of the same trip in the lower gadget results in going back towards l_{h_1}, all the r trips in $\mathbb{T}^L$ have to be used in order to go from l_{h_1} to c_{r+1}. As in any temporal path from l_{h_1} to c_{r+1} in $G[D_r, \mathbb{T}_r, \tau]$, the starting times of the temporal edges must increase, this implies that all starting times of trips in $\mathbb{T}^L$ must be pairwise distinct and that the trip with the earliest starting time is $T_{h_1}^L$ as it is the only one containing the edge (l_{h_1}, c_{2r}). If $h_1 \neq i_{min}$, then τ fails to make c_{r+1} τ-reachable from l_{h_1}. If $h_2 \neq i_{max}$, a similar reasoning allows us to show that τ fails to make u_{h_2} reachable from c_r by considering a temporal path from c_r to u_{h_2} in $G[D_r, \mathbb{T}_r, \tau]$, and the trips in $\mathbb{T}^U$. Hence, $h_1 \neq i_{\min} \vee h_2 \neq i_{\max}$ and $h_1 \neq h_2$ implies that τ fails to make $d_{h_2}^{l_2}$ τ-reachable from $d_{h_1}^{l_1}$, and the claim is proved. □

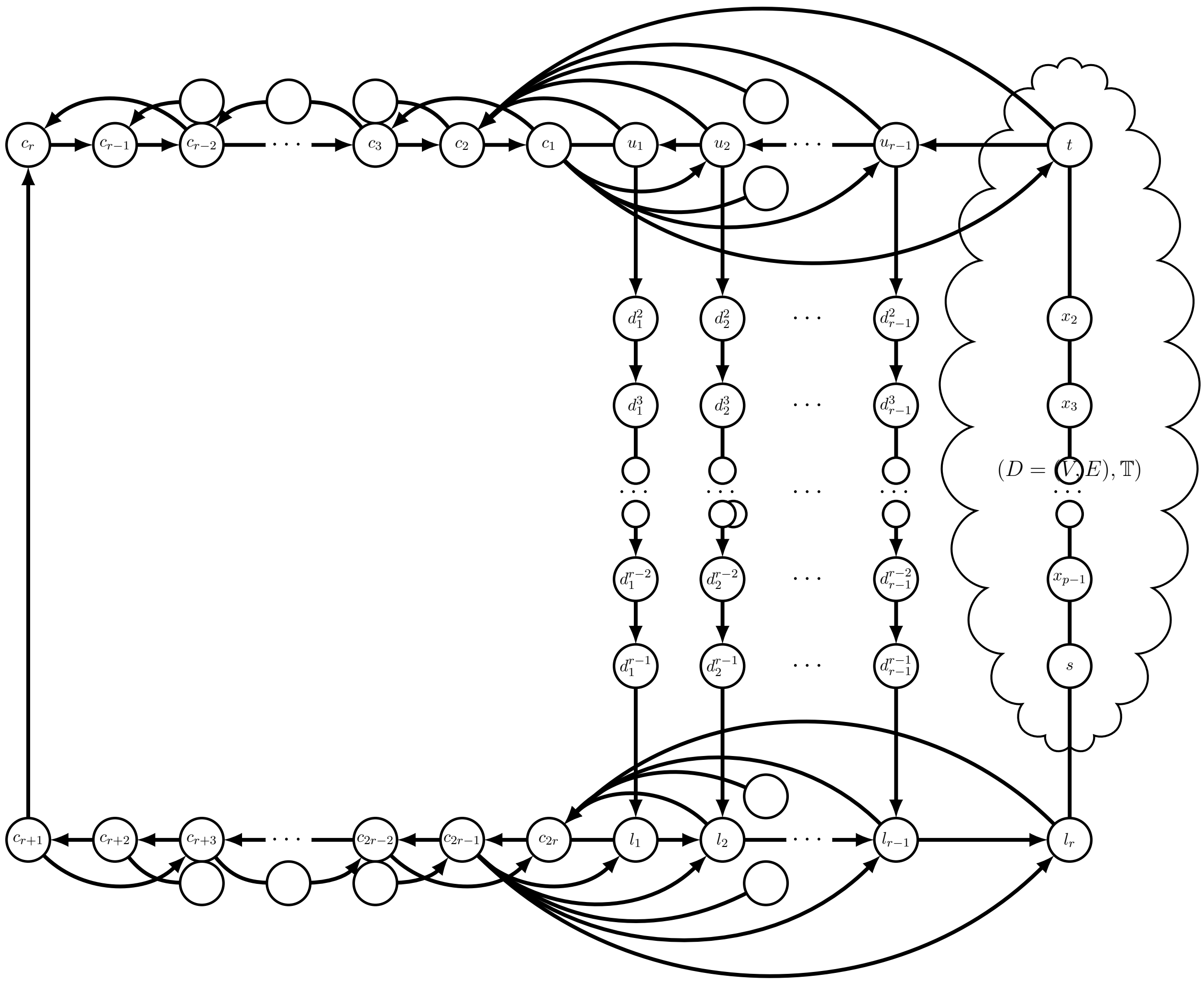

Figure 8: The reduction from O2O-RTT to MRTT used in the proof of Theorem 6 (the edges in D are not shown, unless they coincide with one of the shown edges). All the edges without arrows are present in both directions (for instance, both (t, x_2) and (x_2, t) are included in the set of edges, while only (t, u_{r-1}) is included in the set of edges).

The construction of the trip network $(D_r, \mathbb{T}_r)$ in the proof of the above theorem can be adapted in order to prove the following inapproximability results for both the MRTT and the SS-MRTT problem, in the case of strongly temporalisable trip networks.

Theorem 6 *Unless $P = NP$, the* MRTT *problem cannot be approximated within a factor less than $\frac{\sqrt{n}}{12}$ even if the input trip network is strongly temporalisable.*

Proof. As in the proof of Theorem 2, we use the gap technique by reducing in polynomial time the O2O-RTT decision problem to the MRTT problem. Consider an instance $\langle(D=(V,E),\mathbb{T}),s,t\rangle$ of the O2O-RTT problem, where $V=\{t=x_1,\ldots,x_p=s\}$ (without loss of generality, we assume that $p>22$). We then define a trip network $(D'=(V',E'),\mathbb{T}')$ as follows (see Figure 8). Let $(D_r=(V_r,E_r),\mathbb{T}_r)$, with $r=p+1$, be the trip network constructed in the proof of Theorem 5 (note that $r>23$): in the following, we identify each node $x_i\in V$ with the node d^i_r of the last descending gadget of D_r (that is, we consider $V=\{d^1_r=u_r=t,d^2_r=x_2,\ldots,d^{r-2}_r=x_{r-2}=x_{p-1},d^{r-1}_r=s\}$). Note that $l_r=d^r_r$ is not a node in V.. We then set $V'=V_r$ and

$$\begin{aligned}E' &= E_r\cup\{(d^i_r,d^j_r):(x_i,x_j)\in E\cup\{(d^{i+1}_r,d^i_r):i\in[r-1]\}\\ &\quad \cup\{(u_{i+1},u_i):i\in[r-1]\}\cup\{(l_i,l_{i+1}):i\in[r-1]\}\cup\{(u_1,c_1),(c_{2r},l_1)\}\end{aligned}$$

Note that, according to the definition of V' and E', each trip in $\mathbb{T}$ can be considered as a walk in D'. We then set

$$\mathbb{T}'=\mathbb{T}\cup\mathbb{T}_r\setminus\{T^U_r,T^L_r,T^{\downarrow\mathrm{l}}_r,T^{\downarrow\mathrm{r}}_r\}\cup\{T_U,T_L,T_{\uparrow\downarrow}\},$$

where $T_U,T_L,T_{\uparrow\downarrow}$ are the following three trips.

- $T_U=\langle t,u_{r-1},\ldots,u_1,c_1,t,c_2,c_1,c_3,c_2,\ldots,c_{r-1},c_{r-2},c_r,\,c_{r-1}\rangle$ (intuitively, T_U replaces T^U_r: it first visits $t=u_r,\ldots,u_1$, it then goes to c_1, and it finally continues exactly as T^U_r).
- $T_L=\langle c_{r+2},c_{r+1},c_{r+3},c_{r+2},\ldots,c_{2r-2},c_{2r},\,c_{2r-1},l_r,c_{2r},l_1,l_2,\ldots,l_r\rangle$ (intuitively, T_L replaces T^L_r: it first starts exactly as T^L_r, and it then visits $l_1,\ldots,l_r$).
- $T_{\uparrow\downarrow}=\langle s,d^{r-2}_r,\ldots,t,d^2_r,\ldots,d^{r-2}_r,s,l_r,s\rangle$ (intuitively, $T_{\uparrow\downarrow}$ replaces both $T^{\downarrow\mathrm{l}}_r$ and $T^{\downarrow\mathrm{r}}_r$).

$(D',\mathbb{T}')$ **is strongly temporalisable** (even if $(D,\mathbb{T})$ is not). The proof is similar to the one proving that $(D_r,\mathbb{T}_r)$ is strongly temporalisable (see the proof of Theorem 5). Indeed, whenever $h=r$ or $k=r$ in Table 3, we can replace T^U_h or T^U_k by T_U (respectively, T^L_h or T^L_k by T_L and $T^{\downarrow\mathrm{l}}_h$ or $T^{\downarrow\mathrm{l}}_k$ by $T_{\uparrow\downarrow}$) in the schedules included in the table. Since T^U_r (respectively, T^L_r and $T^{\downarrow\mathrm{l}}_r$) is included in T_U (respectively, T_L and $T_{\uparrow\downarrow}$), by doing so we have that all the reachability properties of the table are still satisfied apart from the last case of the last cell of the table itself, with $h=k=r$. However, in this case, for any temporalisation τ, we have that $d^{l_2}_r\in V^{\downarrow}$ is τ-reachable from $d^{l_1}_r\in V^{\downarrow}$ with $l_2<l_1$, since we can just use the trip $T_{\uparrow\downarrow}$ (which allows to go up from $d^{l_1}_r$ to $d^{l_2}_r$).

If $(D,\mathbb{T})$ is (s,t)-temporalisable, then there exists a temporalisation τ' of $(D',\mathbb{T}')$ with τ'-reachability at least $((r-1)r)^2$. To this aim, first note that, for any temporalisation τ' of $(D',\mathbb{T}')$, if $\tau'(T_{\uparrow\downarrow})=t'$, then the trip $T_{\uparrow\downarrow}$ arrives at l_r at time $t'+2r-3$, and terminates in s at time $t'+2r-2$. If $(D,\mathbb{T})$ is (s,t)-temporalisable, then there exists a temporalisation τ of $(D,\mathbb{T})$ such that t is τ-reachable from s. Let P be any temporal path in $G[D,\mathbb{T},\tau]$ from s to t, and let t_s (respectively, t_a) be the starting (respectively, arrival) time of P from s (respectively, in t). We then define a temporalisation τ' of $(D',\mathbb{T}')$ as follows. For any $T\in\mathbb{T}$, $\tau'(T)=\tau(T)$. Moreover, for any $h\in[r-1]$, $\tau'(T^{\downarrow\mathrm{l}}_h)=t_s-2r+1$, and $\tau'(T_L)=t_s-1-|T_L|=t_s-3r$. This allows the trips $T^{\downarrow\mathrm{l}}_h$ to "meet" the trip T_L in l_h at time $t_s-(r-h)-1$: note that T_L arrives in l_r at time t_s-1. Hence, we set $\tau'(T_{\uparrow\downarrow})=t_s-2r+2$ so that $T_{\uparrow\downarrow}$ arrives in l_r also at time t_s-1. By using the (last edge of the) trip $T_{\uparrow\downarrow}$, and then the path P, we can arrive in t at time t_a. By setting $\tau'(T_U)=t_a$ and, for any $h\in[r-1]$, $\tau'(T^{\downarrow\mathrm{r}}_h)=t_a+r-h$, we can arrive at any node d^k_h at time $t_a+r-h+k-1$. In other words, we have shown that all nodes of the first $r-1$ descending gadgets are τ'-reachable one from the other. That is, the τ'-reachability is at least $((r-1)r)^2$.

If $(D, \mathbb{T})$ is not (s,t)-temporalisable, then the τ'-reachability of any temporalisation τ' of $(D', \mathbb{T}')$ is at most $3rn + 7r^2(r-1)$. Let τ' be a temporalisation of $(D', \mathbb{T}')$. First note that τ' induces a temporalisation τ of $(D, \mathbb{T})$. Since $(D, \mathbb{T})$ is not (s,t)-temporalisable, we have that t is not τ-reachable from s. Moreover, since the edge (l_r, s) is the last edge in the trip $T_{\uparrow\downarrow}$, it cannot be used before the other edges in this trip. As a consequence, t is not τ''-reachable from l_r, where τ'' is the temporalisation induced by τ' on $(D', \mathbb{T} \cup \{T_{\uparrow\downarrow}\})$. The topology of D' thus implies that, for all $i, j \in [r]$, all temporal paths from l_i to u_j in $G[D', \mathbb{T}', \tau']$ must pass through the nodes $c_{2r}, c_{2r-1}, \dots, c_{r+1}$, the edge (c_{r+1}, c_r), and the nodes $c_r, c_{r-1}, \dots, c_1$.

Let $T^L_{i_{\min}}$ be one of the trips with minimum starting time according to τ' among all the trips in the lower gadget ($i_{\min} = r$ if T_L is the only trip with minimum starting time), and let $T^U_{i_{\max}}$ be one of the trips with maximum starting time according to τ' among all the trips in the upper gadget ($i_{\max} = r$ if T_U is the only trip with maximum starting time). Similarly to the proof of Theorem 5, we can then show that there is no temporal path in $G[D', \mathbb{T}', \tau']$ from l_h to c_{r+1} for $h \in [r] \setminus \{i_{\min}\}$ nor from c_r to u_k for $k \in [r] \setminus \{i_{\max}\}$. Note that the additional part $\langle c_{2r}, l_1, \dots, l_r \rangle$ of T_L (respectively, $\langle t, u_{r-1}, \dots, u_1, c_1 \rangle$ of T_U), compared to T^L_r (respectively, T^U_r), does not change the reasoning as it comes at the end (respectively, the beginning) of the trip and, in particular, after edge (l_r, c_{2r}) (respectively, before the edge (c_1, l_1)).

We can thus similarly conclude that $d^{l_2}_{h_2}$ is not τ'-reachable from $d^{l_1}_{h_1}$ for all $h_1, h_2 \in [r-1]$ with $h_1 \neq i_{\min}$ or $h_2 \neq i_{\max}$ and all l_1, l_2 with $1 < l_1, l_2 < r$. Note that the situation is different from the previous construction for $l_1 = l_2 = r$ or $l_1 = l_2 = 1$ as T_L makes $d^r_{h_2} = l_{h_2}$ τ'-reachable from $d^r_{h_1} = l_{h_1}$ for $h_1 < h_2$ and that T_U makes $d^1_{h_1} = u_{h_1}$ τ'-reachable from $d^1_{h_2} = u_{h_2}$ for $h_1 < h_2$. Overall, this means that only nodes in

$$\{d^1_{h_1}, \dots d^r_{h_1}, l_1, \dots, l_r, c_1, \dots, c_{2r}, u_1, \dots, u_r, d^1_{i_{\max}}, \dots, d^r_{i_{\max}}, d^1_r, \dots, d^r_r\}$$

can be τ'-reachable from $d^{l_1}_{h_1}$ for $h_1 \in [r-1]$ and $l_1 \in [r]$. Thus, the nodes in the first $r-1$ descending gadgets have τ'-reachability at most $7r$. The other $3r$ nodes have τ'-reachability at most n.

The MRTT problem cannot be approximated within a factor less than $\frac{\sqrt{n}}{12}$. We now prove that any polynomial-time algorithm $\mathcal{A}$ solving MRTT with approximation ratio $\rho < \frac{\sqrt{n}}{12}$ would allow us to decide in polynomial-time whether $(D, \mathbb{T})$ is (s,t)-temporalisable. As this latter problem is NP-complete, as stated by Theorem 1, this will conclude the proof of the theorem. If $(D, \mathbb{T})$ is (s,t)-temporalisable, then then there exists a temporalisation of $(D', \mathbb{T}')$ whose reachability is at least $((r-1)r)^2$. This implies that $\mathcal{A}$, with input the trip network $(D', \mathbb{T}')$, has to provide a temporalisation τ' with τ'-reachability at least $\frac{((r-1)r)^2}{\rho} > \frac{12r^2(r-1)^2}{r+1} > 11r^2(r-1)$ (note that $\sqrt{n} = \sqrt{r^2+2r} < r+1$ and $\frac{r-1}{r+1} = 1 - \frac{2}{r+1} > \frac{11}{12}$ for $r > 23$). On the other hand, if $(D, \mathbb{T})$ is not (s,t)-temporalisable, then the τ'-reachability of any temporalisation τ' of $(D', \mathbb{T}')$ is at most $3rn + 7r^2(r-1) = 10r^3 - r^2 < 11r^2(r-1)$ (note that $r^3 > 10r^2$ for $r > 10$). In summary, $(D, \mathbb{T})$ is (s,t)-temporalisable if and only if $\mathcal{A}$, with input the trip network $(D', \mathbb{T}')$, returns a temporalisation whose reachability is greater than $11r^2(r-1)$. □

Theorem 7 *Unless $P = NP$, the* SS-MRTT *problem cannot be approximated within a factor less than $\frac{\sqrt{n}}{12}$ even if the input trip network is strongly temporalisable.*

Proof. The reduction is exactly the same as the one used in the proof of the previous theorem. According to that proof, if $(D, \mathbb{T})$ is (s,t)-temporalisable, then there exists a temporalisation τ' of $(D', \mathbb{T}')$ such that any source in one of the first $r-1$ descending gadgets has τ'-reachability at least $r(r-1)$. On the other hand, if $(D, \mathbb{T})$ is not (s,t)-temporalisable, then, for any temporalisation τ' of $(D', \mathbb{T}')$, any source in one of the first $r-1$ descending gadgets has τ'-reachability at most $7r$. Any polynomial-time algorithm $\mathcal{A}$ solving SS-MRTT with approximation ratio $\rho < \frac{\sqrt{n}}{12}$ would then allow us to decide, in polynomial-time, whether $(D, \mathbb{T})$ is (s,t)-temporalisable, since $\frac{r(r-1)}{\rho} > \frac{12r(r-1)}{r+1} > 11r > 7r$ for $r > 23$. This concludes the proof of the theorem. □

4.1 Symmetric and strongly temporalisable trip networks

Because of the last theorem, we now focus on symmetric *and* strongly temporalisable trip networks.

Fact 2 *Let $(D, \mathbb{T})$ be a symmetric trip network. For any node u, there exists a schedule S of $(D, \mathbb{T})$ such that, for any node v reachable from u in D, v is S-reachable from u.*

Proof. We first note that, due to the symmetricity, any schedule S of $(D, \mathbb{T})$ is such that each node of a trip $T \in \mathbb{T}$ is S-reachable from any other node of T. Let $\mathcal{T}_u$ be a breadth-first search tree in the induced multidigraph M rooted in u, whose height is $h_{\mathcal{T}_u}$. By using $\mathcal{T}_u$, we will now define a schedule S of $(D, \mathbb{T})$ such that, for any node v in $\mathcal{T}_u$, v is S-reachable from u. This will prove the fact.

We will construct nested partial schedules $S_0, \dots, S_{h_{\mathcal{T}_u}} = S$ where a larger and larger subset of $\mathbb{T}$ is scheduled. Given a partial schedule S_ℓ, we say that an edge is "covered" by S_ℓ if it belongs to one of the trips scheduled in S_ℓ. Similarly, a node is "covered" by S_ℓ if it is the head or the tail of an edge covered by S_ℓ. At the beginning, we consider an empty schedule S_0. For each level ℓ of $\mathcal{T}_u$ with $\ell \in [h_{\mathcal{T}_u}]$, let $e_1, \dots, e_{k_\ell}$ be the edges connecting a node at level $\ell - 1$ to a node at level ℓ which are not yet covered by $S_{\ell-1}$. Let $T_{\ell,1}, \dots, T_{\ell,k_\ell}$ be k_ℓ (not necessarily distinct) trips in $\mathbb{T}$, which contain the edges $e_1, \dots, e_{k_\ell}$, respectively. Recall that $\reflectbox{T}_{\ell,1}, \dots, \reflectbox{T}_{\ell,k_\ell}$ denote their respective reverse trips. Let $\mathbb{T}_\ell$ denote the set of trips in $\{T_{\ell,1}, \reflectbox{T}_{\ell,1}, \dots, T_{\ell,k_\ell}, \reflectbox{T}_{\ell,k_\ell}\}$. As the edges $e_1, \dots, e_{k_\ell}$ were not covered by $S_{\ell-1}$, trips in $\mathbb{T}_\ell$ are not included in $S_{\ell-1}$. We can thus define S_ℓ as $S_{\ell-1}$ followed by the trips in $\mathbb{T}_\ell$ in an arbitrary order. Note that all edges from level $\ell - 1$ to level ℓ are now covered by S_ℓ. We continue similarly for the next levels and define $S = S_{h_{\mathcal{T}_u}}$ as the schedule obtained for the last layer.

To prove that any node v in $\mathcal{T}_u$ is S-reachable, we show that the following invariant is preserved: after processing each level ℓ, any node covered by S_ℓ is S_ℓ-reachable. Consider an edge e which is covered by S_ℓ but not by $S_{\ell-1}$ and let v' be its head or tail. Let $T_{\ell,i}, \reflectbox{T}_{\ell,i}$ denote the trip pair in $\mathbb{T}_\ell$ that contains e. Consider the edge e_i from level $\ell - 1$ to ℓ that belongs to $T_{\ell,i}$. The tail u' of e_i is either u or the head of an edge from level $\ell - 2$ to $\ell - 1$. As such an edge is covered by $S_{\ell-1}$, u' is thus $S_{\ell-1}$-reachable according to the invariant. As $T_{\ell,i}$ or $\reflectbox{T}_{\ell,i}$ contains a walk from u' to v', and both $T_{\ell,i}$ and $\reflectbox{T}_{\ell,i}$ are scheduled after $S_{\ell-1}$ in S_ℓ, v' is S_ℓ-reachable. The conclusion follows from the fact that all nodes in $\mathcal{T}_u$ are covered by $S = S_{h_{\mathcal{T}_u}}$. □

Fact 3 *Let $(D, \mathbb{T})$ be a symmetric trip network. For any node u, there exists a schedule S of $(D, \mathbb{T})$ such that, for any node v such that u is reachable from v in D, u is S-reachable from v.*

Proof. The proof is similar to the proof of Fact 2. □

Corollary 1 *Let $(D, \mathbb{T})$ be a symmetric trip network. Then, $(D, \mathbb{T})$ is strongly temporalisable if and only if D is strongly connected.*

Proof. If $(D, \mathbb{T})$ is strongly temporalisable, then, for any two nodes u and v in D, there exists a temporalisation $\tau_{u,v}$ of $(D, \mathbb{T})$, such that v is $\tau_{u,v}$-reachable from u. In other words, there exists a temporal path $P_{u \to v}$ from u to v in $G[D, \mathbb{T}, \tau_{u,v}]$. Hence, v is reachable from u in D. The converse implication is a direct consequence of Fact 2. □

We now prove that the MRTT problem remains NP-hard, even when we assume that the underlying multidigraph is strongly connected and that the trip network is symmetric (from the previous corollary, this implies that the trip networks is strongly temporalisable).

Theorem 8 *The* MRTT *problem problem is NP-hard, even if $(D, \mathbb{T})$ is a symmetric trip network and D is strongly connected.*

Proof. We reduce 3-SAT to MRTT as follows. Let us consider a 3-SAT formula Φ, with n variables $x_1, \ldots, x_n$ and m clauses $c_1, \ldots, c_m$. Without loss of generality, we will assume that each variable appears positive in at least one clause and negative in at least one clause, that no literal appears in all clauses, and that there are at least two clauses. We set $l = \lceil (7n + m(m+3))^2/(m+2) \rceil + 1$ and $L = (7n + m(m+3))^2 + 1$, and we define the digraph $D = (V, E)$ as the union of the following gadgets (in the following, for each edge included in E, its reverse edge is implicitly also included in E).

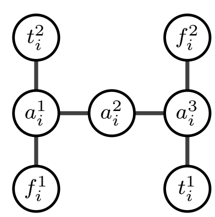

Figure 9: The variable gadget in the reduction of 3-SAT to symmetric MRTT (see the proof of Theorem 8).

Variable gadgets (see Figure 9). For each variable x_i of Φ with $i \in [n]$, V contains the seven *variable nodes* t_i^1, t_i^2, f_i^1, f_i^2, a_i^1, a_i^2, and a_i^3, and E contains the six edges (t_i^1, a_i^3), (a_i^3, f_i^2), (a_i^1, a_i^2), (a_i^2, a_i^3), (f_i^1, a_i^1), and (a_i^1, t_i^2).

Clause gadgets (see Figure 10). For each clause c_j of Φ with $j \in [m]$, V contains the two *clause nodes* c_j^1 and c_j^2 (we will call $c_1^1, \ldots, c_m^1$ the *bottom* clause nodes and $c_1^2, \ldots, c_m^2$ the *top* clause nodes), and the *middle nodes* d_j^k for $k \in [l]$. For each variable x_i which appears (positive or negative) in c_j, V contains the *head* node e_j^i. Finally, for each $h \in [m]$ with $h \neq j$, V contains the *head* node g_j^h. Concerning the edges, for each variable x_i which appears positive in c_j, E

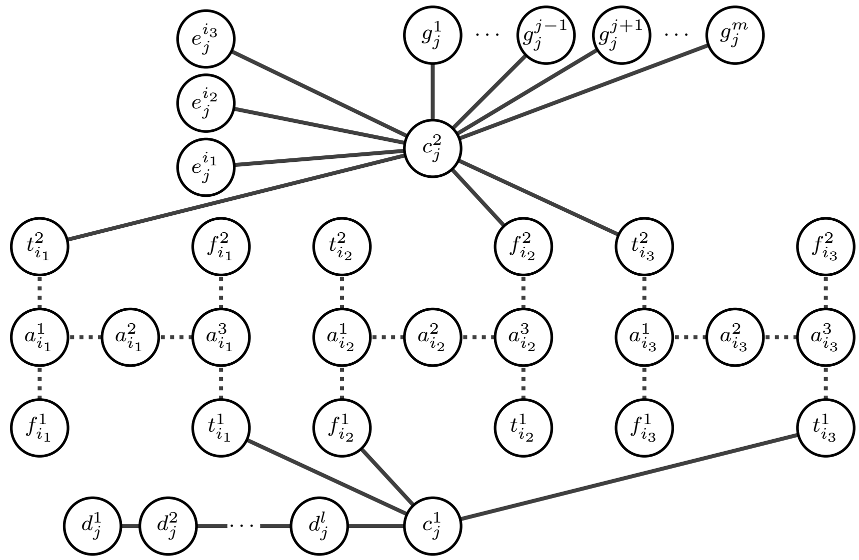

Figure 10: The clause gadget in the reduction of 3-SAT to symmetric MRTT (see the proof of Theorem 8), corresponding to the clause $c_j = x_{i_1} \vee \neg x_{i_2} \vee x_{i_3}$ (the dotted edges are included in the variable gadgets corresponding to the variables x_{i_1}, x_{i_2}, and x_{i_3}). Note that, for each $h \in [m]$ with $h \neq j$, E includes also the edge (c_j^1, c_h^2) (and its reverse edge).

contains the edges (c_j^1, t_i^1) and (t_i^2, c_j^2), while, for each variable x_i which appears negative in c_j, E contains the edges (c_j^1, f_i^1) and (f_i^2, c_j^2). In both cases, E contains the edge (c_j^2, e_j^i). For each $h \in [m]$ with $h \neq j$, E contains the edges (c_j^1, c_h^2) and (c_j^2, g_j^h). Finally, E contains the edges (d_j^k, d_j^{k+1}), for $k \in [l-1]$, and the edge (d_j^l, c_j^1).

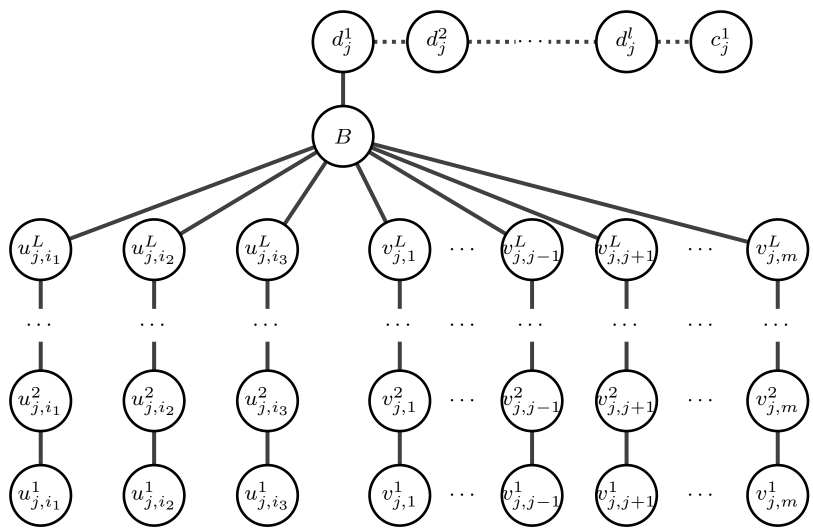

Figure 11: The part of the bottom hub gadget in the reduction of 3-SAT to symmetric MRTT (see the proof of Theorem 8), corresponding to the clause c_j which contains the variables x_{i_1}, x_{i_2}, and x_{i_3} (the dotted edges are included in the clause gadget corresponding to the clause c_j).

Bottom hub gadget (see Figure 11). V contains the *bottom hub node* B. For each clause c_j of Φ with $j \in [m]$, and for each variable x_i that appears (positive or negative) in c_j, V contains the set $A^{\mathrm{u}}_{j,i} = \{u^k_{j,i} : k \in [L]\}$. Moreover, for each $h \in [m]$, such that $h \neq j$, V contains the set $A^{\mathrm{v}}_{j,h} = \{v^k_{j,h} : k \in [L]\}$. We will refer to the nodes in $A^{\mathrm{u}}_{j,i}$ and in $A^{\mathrm{v}}_{j,h}$ as the *bottom tail nodes*. Concerning the edges, E contains the edge (B, d^1_j), and, for each variable x_i that appears (positive or negative) in c_j, the set $\{(u^k_{j,i}, u^{k+1}_{j,i}) : k \in [L-1]\}$ and the edge $(u^L_{j,i}, B)$. Finally, for each $h \in [m]$, such that $h \neq j$, E contains the set $\{(v^k_{j,h}, v^{k+1}_{j,h}) : k \in [L-1]\}$ and the edge $(v^L_{j,i}, B)$.

Top hub gadget (see Figure 12) V contains the *top hub node* U. For each clause c_j of Φ with $j \in [m]$, and for each variable x_i that appears (positive or negative) in c_j, V contains the set $A^{\mathrm{w}}_{j,i} = \{w^k_{j,i} : k \in [L]\}$. We will refer to the nodes in $A^{\mathrm{w}}_{j,i}$ as the *top tail nodes*. Concerning the edges, for each variable x_i that appears (positive or negative) in c_j, E contains the set $\{(w^k_{j,i}, w^{k+1}_{j,i}) : k \in [L-1]\}$ and the edge $(w^L_{j,i}, U)$. Finally, if x_i appears positive in c_j, E contains the edge (U, t^2_i), while if x_i appears negative in c_j, E contains the edge (U, f^2_i).

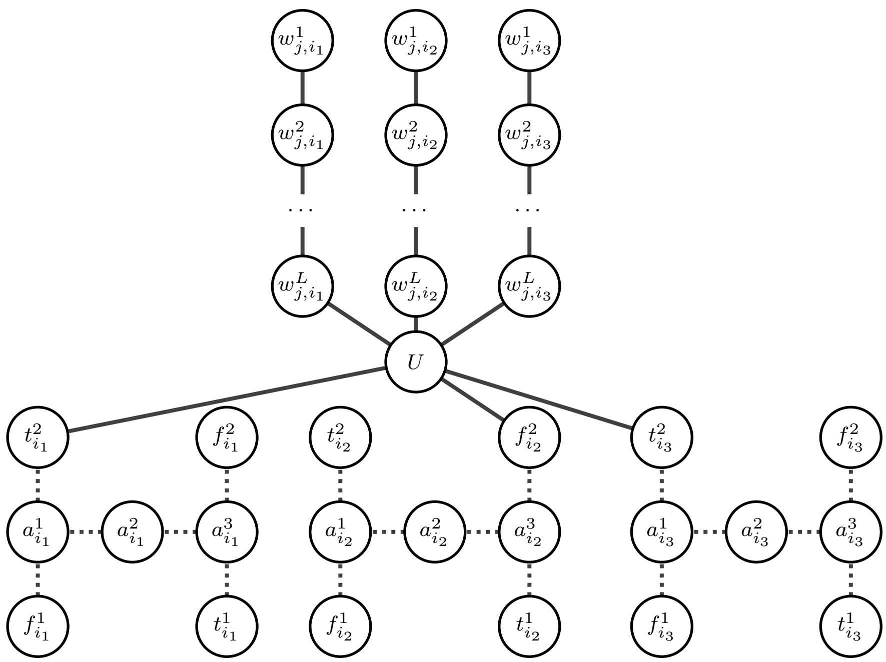

Figure 12: The part of the top hub gadget in the reduction of 3-SAT to symmetric MRTT (see the proof of Theorem 8), corresponding to the clause $c_j = x_{i_1} \vee \neg x_{i_2} \vee x_{i_3}$ (the dotted edges are included in the variable gadgets corresponding to the variables x_{i_1}, x_{i_2}, and x_{i_3}).

Number of nodes. Let us first compute the cardinality of V. There are $7n$ variable nodes, $2m$ clause nodes, $3m + m(m-1)$ head nodes, ml middle nodes, 2 hub nodes, $3mL + m(m-1)L$ bottom tail nodes, and $3mL$ top tail nodes. Thus, $|V| = 2 + 7n + m(L+1)(m+4) + ml + mL$.

Trips. We now define the trip collection $\mathbb{T}$ in D (see Figure 13). In the following, for each trip T included in $\mathbb{T}$, we implicitly assume that the symmetric trip $\overline{T}$ is also included in $\mathbb{T}$.

Variable trips. For each $i \in [n]$, $\mathbb{T}$ contains the trips $T_i^{\mathrm{t}} = \langle t_i^1, a_i^3, f_i^2 \rangle$, $T_i^{\mathrm{f}} = \langle f_i^1, a_i^1, t_i^2 \rangle$, and $T_i^{\mathrm{a}} = \langle a_i^1, a_i^2, a_i^3 \rangle$ (see Figure 13(a)).

Bottom-variable trips. For each clause c_j of Φ with $j \in [m]$, if c_j contains the literal x_i, $\mathbb{T}$ contains the trip $T_{j,i}^{\mathrm{u}} = \langle u_{j,i}^1, \ldots, u_{j,i}^L, B, d_j^1, \ldots, d_j^l, c_j^1, t_i^1 \rangle$ (see Figure 13(b)), while if c_j contains the literal $\neg x_i$, $\mathbb{T}$ contains the trip $T_{j,i}^{\mathrm{u}} = \langle u_{j,i}^1, \ldots, u_{j,i}^L, B, d_j^1, \ldots, d_j^l, c_j^1, f_i^1 \rangle$ (see Figure 13(c)).

Bottom-clause trips. For each $(j,h) \in [m]^2$ such that $j \neq h$, $\mathbb{T}$ contains the trip $T_{j,h}^{\mathrm{v}} = \langle v_{j,h}^1, \ldots, v_{j,h}^L, B, d_j^1, \ldots, d_j^l, c_j^1, c_h^2, g_j^h \rangle$ (see Figure 13(d)).

Top trips. For each clause $c_j$of Φ with $j \in [m]$, if c_j contains the literal x_i, $\mathbb{T}$ contains the trip $T_{j,i}^{\mathrm{w}} = \langle w_{j,i}^1, \ldots, w_{j,i}^L, U, t_i^2, c_j^2, e_j^i \rangle$ (see Figure 13(e)), while if c_j contains the literal $\neg x_i$, $\mathbb{T}$ contains the trip $T_{j,i}^{\mathrm{w}} = \langle w_{j,i}^1, \ldots, w_{j,i}^L, U, f_i^2, c_j^2, e_j^i \rangle$ (see Figure 13(f)).

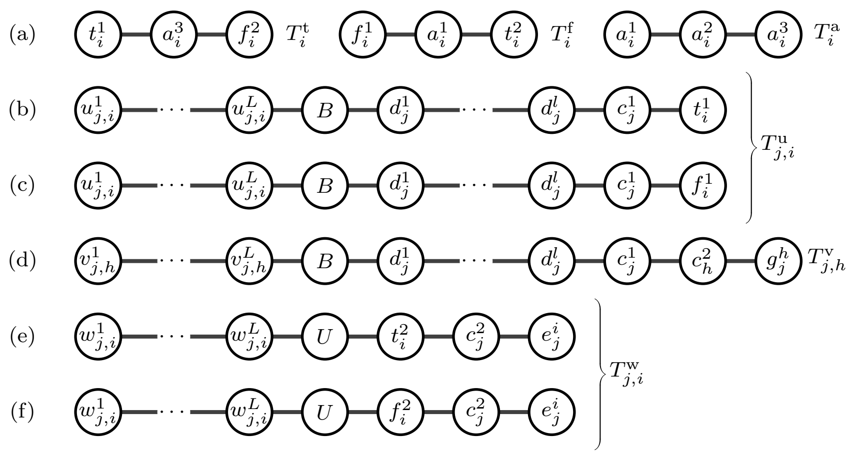

Figure 13: The trips in the reduction of 3-SAT to symmetric MRTT (see the proof of Theorem 8): (a) variable trips, (b) and (c) bottom-variable trips, (d) bottom-clause trips, and (e) and (f) top trips. Each kind of trip is included in both directions (that is, the trip collection is symmetric).

See Figure 14 for a global view of the trip network $(D, \mathbb{T})$.

Basic idea of the reduction. The temporal connections that are made possible by a temporalisation of the variable trips T_i^{t}, T_i^{f}, and T_i^{a} correspond to choosing whether the variable x_i is set to true or false. Enabling temporal connections from the middle nodes, which are between the bottom hub B and a bottom clause node c_j^1, to the head nodes connected to the top clause node c_j^2, corresponds to a "reward" in terms of reachability for satisfying the clause. The large size of the tails forces some constraints on the temporalisations with high reachability. This ensures that the "reward" is obtained only if the clause is satisfied. In particular, we will show that if there exists a satisfying assignment for Φ, then it is possible to produce a temporalisation τ such that the τ-reachability is at least Q, where $Q = |V|^2 - (7n+m(m+3))^2$. Otherwise, if Φ is not satisfiable, then any temporalisation has reachability less than Q. More precisely, since $Q > |V|^2 - L$ (recall that $L = (7n+m(m+3))^2+1$), we will show that any temporalisation that misses a connection from a bottom/top tail node to any node or from a node to any bottom/top tail node has reachability less than Q. Similarly, since $Q > |V^2| - (m+2)l$ (recall that $l = \lceil (7n+m(m+3))^2/(m+2) \rceil + 1$), we will

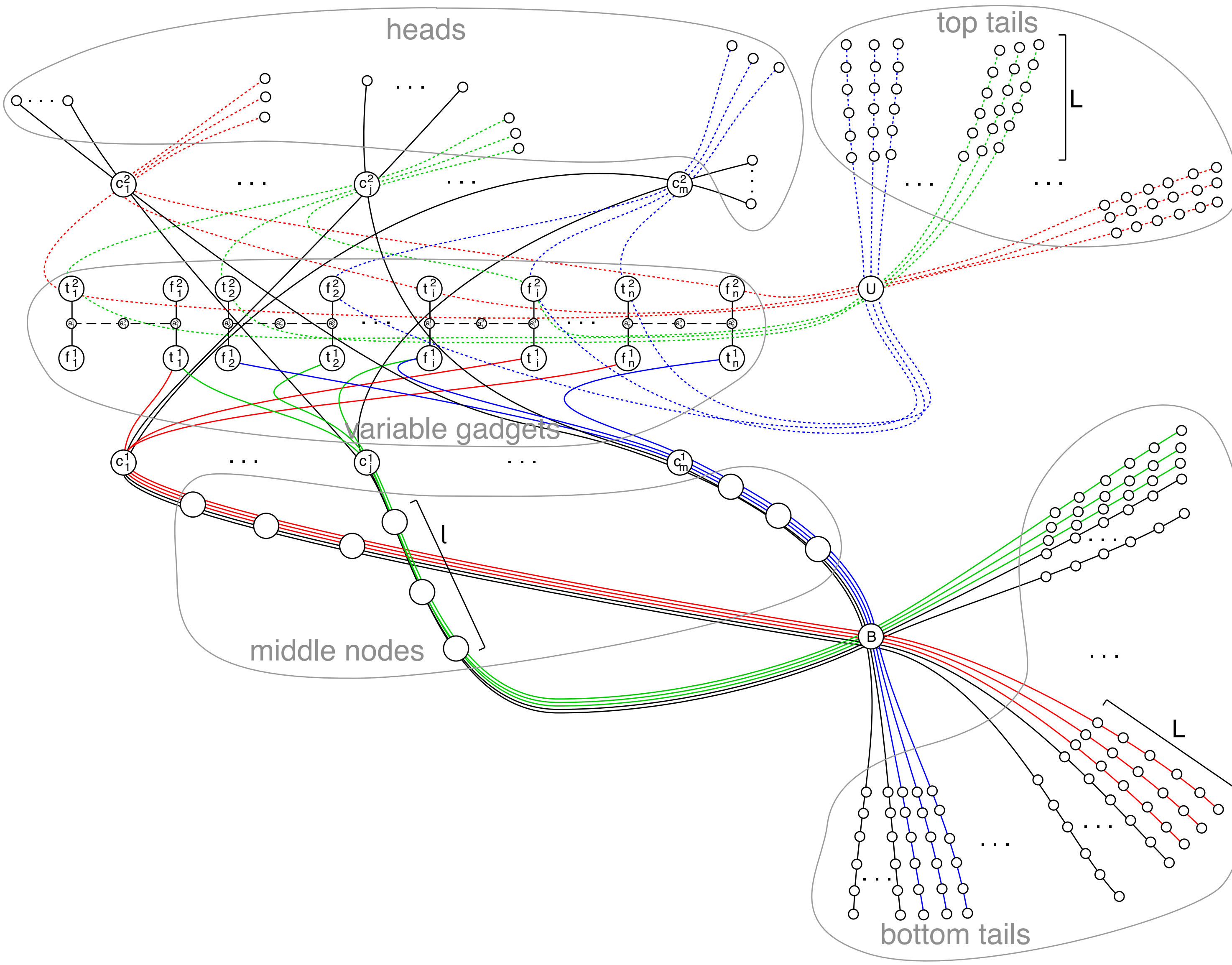

Figure 14: A global view of the reduction of 3-SAT to symmetric MRTT. Here the clause $c_j = (x_1 \wedge x_2 \wedge \overline{x_i})$ is associated to two nodes c_j^1 and c_j^2 that are connected to variable gadgets for x_1, x_2 and x_i through green lines (plain lines for bottom-variable trips and dotted lines for top trips). For a more detailed view of each gadget, see Figures 9, 10, 11, 12, and 13.

show that missing the connections from the middle nodes to the head nodes of the corresponding top clause node, also leads to a reachability lower than Q.

Activation of pairs of variable nodes. Consider a variable x_i and the associated variable gadget (see Figure 9). For a given temporalisation τ, let $t_1 = \tau(T_i^{\mathrm{f}})$, $t_2 = \tau(T_i^{\mathrm{a}})$, and $t_3 = \tau(T_i^{\mathrm{t}})$. Note that f_i^2 is τ-reachable from f_i^1 by using only the variable trips corresponding to the variable x_i if and only if $t_1 + 1 \leq t_2$ and $t_2 + 2 \leq t_3 + 1$, as these conditions enable transfers at a_i^1 and a_i^3. When this is the case, we say that τ *activates* (f_i^1, f_i^2). Note that if τ *activates* (f_i^1, f_i^2), then $t_1 \leq t_3 - 2$.

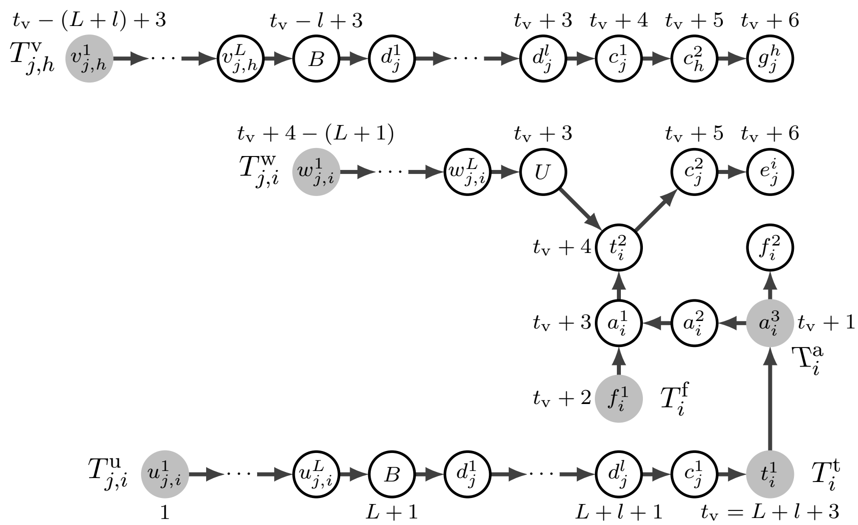

Figure 15: The temporalisation of the "forward" trips obtained from a truth assignment α satisfying the Boolean formula Φ (here, we assume that the variable x_i appears positive in the clause c_j, and that $\alpha(x_i) = \textsc{True}$). The gray nodes are the starting nodes of the trips. Note that $T^u_{j,i}$ arrives in B at time $L + 1 = t_v - l - 2$.

Similarly, t^2_i is τ-reachable from t^1_i by using only the variable trips corresponding to the variable x_i if and only if $t_3 + 1 \leq t_4$ and $t_4 + 2 \leq t_1 + 1$, where $t_4 = \tau(\Upsilon^{\mathrm{a}}_i)$. When this is the case, we say that τ activates pair (t^1_i, t^2_i). Note that if τ *activates* (t^1_i, t^2_i), then $t_3 \leq t_1 - 2$. The key observation for the sequel is that *no temporalisation can activate both* (f^1_i, f^2_i) *and* (t^1_i, t^2_i). We similarly say that τ activates pair (f^2_i, f^1_i) (respectively, (t^2_i, t^1_i)) when f^1_i (respectively, t^1_i) is τ-reachable from f^2_i (respectively, t^2_i), by using only the variable trips corresponding to the variable x_i. Once again, no temporalisation can activate both (f^2_i, f^1_i) and (t^2_i, t^1_i). However, one can easily see that it is possible to activate either both (f^1_i, f^2_i) and (f^2_i, f^1_i) or both (t^1_i, t^2_i) and (t^2_i, t^1_i).

Constructing a temporalisation from a satisfying assignment. We first show how to construct a temporalisation τ, when Φ is satisfiable, with reachability at least Q. Let α be a truth assignment to $x_1, \ldots, x_n$ that satisfies Φ (see Figures 15 and 16, where we assume that x_i appears positive in c_j and that $\alpha(x_i) = \textsc{True}$).

For any clause c_j and for any variable x_i appearing in c_j, we set $\tau(T^{\mathrm{u}}_{j,i}) = 1$. Note that $T^{\mathrm{u}}_{j,i}$ arrives in B at time $L + 1$, in c^1_j at time $L + l + 2$, and in the variable node connected to c^1_j and included in the variable gadget corresponding to x_i at time $t_{\mathrm{v}} = L + l + 3$. For each $i \in [n]$, if $\alpha(x_i) = \textsc{True}$, then we set $\tau(T^{\mathrm{t}}_i) = t_{\mathrm{v}}$, $\tau(\Upsilon^{\mathrm{a}}_i) = t_{\mathrm{v}} + 1$, and $\tau(T^{\mathrm{f}}_i) = t_{\mathrm{v}} + 2$, so that τ activates (t^1_i, t^2_i) (t^2_i being reachable at time $t_{\mathrm{v}} + 4$). Otherwise (that is, $\alpha(x_i) = \textsc{False}$), we set $\tau(T^{\mathrm{f}}_i) = t_{\mathrm{v}}$, $\tau(T^{\mathrm{a}}_i) = t_{\mathrm{v}} + 1$, and $\tau(T^{\mathrm{t}}_i) = t_{\mathrm{v}} + 2$ so that τ activates (f^1_i, f^2_i) (f^2_i being reachable at time $t_{\mathrm{v}} + 4$). For any clause c_j and for any variable x_i appearing in c_j, we set $\tau(T^{\mathrm{w}}_{j,i}) = t_{\mathrm{v}} + 4 - (L + 1)$, and, for any two clause c_j and c_h with $j \neq h$, we set $\tau(T^{\mathrm{v}}_{j,h}) = t_{\mathrm{v}} - (L + l) + 3$: this implies that all these trips reach their top clause node at the same time, that is, $t_{\mathrm{v}} + 5$.

For any clause c_j and for any variable x_i appearing in c_j, we set $\tau(\Upsilon^{\mathrm{w}}_{j,i}) = t_{\mathrm{v}} + 6$, and, for any two clause c_j and c_h with $j \neq h$, we set $\tau(\Upsilon^{\mathrm{v}}_{j,h}) = t_{\mathrm{v}} + 6$: this implies that all these trips reach their top clause node at the same time, that is, $t_{\mathrm{v}} + 7$. For each $i \in [n]$, if $\alpha(x_i) = \textsc{True}$, then we set

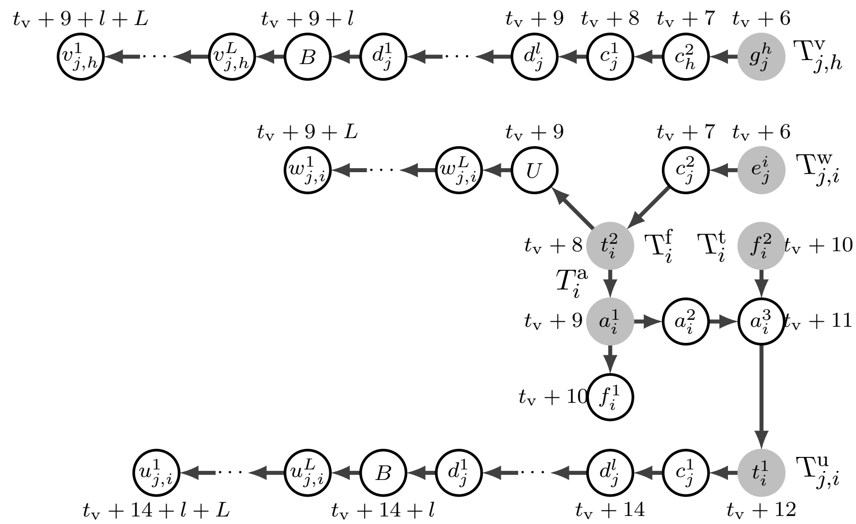

Figure 16: The temporalisation of the "backward" trips obtained from a truth assignment α satisfying the Boolean formula Φ (here, we assume that the variable x_i appears positive in the clause c_j, and that $\alpha(x_i) = \textsc{True}$). The gray nodes are the starting nodes of the trips and $t_v = L + l + 3$.

$\tau(\Upsilon_i^{f}) = t_v + 8$, $\tau(T_i^{a}) = t_v + 9$, and $\tau(\Upsilon_i^{t}) = t_v + 10$, so that τ activates (t_i^2, t_i^1) (t_i^1 being reachable at time $t_v + 12$). Otherwise (that is, $\alpha(x_i) = \textsc{False}$), we set $\tau(\Upsilon_i^{t}) = t_v + 8$, $\tau(\Upsilon_i^{a}) = t_v + 9$, and $\tau(\Upsilon_i^{f}) = t_v + 10$ so that τ activates (f_i^2, f_i^1) (f_i^1 being reachable at time $t_v + 12$). For any clause c_j and for any variable x_i appearing in c_j, we set $\tau(\Upsilon_{j,i}^{u}) = t_v + 12$. Note that $\Upsilon_{j,i}^{u}$arrives in B at time $t_v + 14 + l$.

We now show that the τ-reachability is at least Q. To this aim, let $G = G[D, \mathbb{T}, \tau]$ be the temporal graph induced by τ, and let X be set of the following nodes: the top and bottom tail nodes, the middle nodes, the two hub nodes, and the nodes c_j^1 for $j \in [m]$ (note that $V \setminus X$ contains all the head nodes, all the variable nodes, and the nodes c_j^2 for $j \in [m]$).

Claim 2 *For any node $x \in X$ and for any node $v \in V$, $v \in \mathcal{R}_G(x)$ and $x \in \mathcal{R}_G(v)$.*

Proof. Let us first show that, for any node $x \in X$ and for any node $v \in V \setminus X$, $x \in \mathcal{R}_G(v)$. First note that each top clause node c_j^2 can be reached at time at most $t_v + 7$ by the following set R_j of nodes: c_j^2 itself, each head e_j^i through the trip $\Upsilon_{j,i}^{w}$ and each head g_j^h with $h \neq j$ through the trip $\Upsilon_{j,h}^{v}$ (see Figure 16), and all variable nodes of any gadget associated to a variable x_i appearing in c_j through the trips T_i^{t}, Υ_i^{a}, and T_i^{f} or through the trips T_i^{f}, T_i^{a}, and T_i^{t} (see Figure 15). Notice that $\bigcup_{j \in [m]} R_j = V \setminus X$. Now we show that, by starting from c_j^2 at time $t_v + 7$, it is possible to reach each node in X, thus implying that, for any node $x \in X$ and for any node $v \in V \setminus X$, $x \in \mathcal{R}_G(v)$.

- The top hub U can be reached at time $t_v + 9$ through any trip $\Upsilon_{j,i}^{w}$ such that variable x_i appears in c_j (see Figure 16).
- All top tail nodes $w_{p,q}^r$ are reachable through the trips $\Upsilon_{p,q}^{w}$, which all "meet" in U at time $t_v + 9$ (see Figure 16).

- For any $h \in [m]$ with $h \neq j$, the bottom clause nodes c_h^1, the middle nodes d_h^r for $r \in [l]$, the bottom hub B, and the bottom tail nodes $v_{h,j}^s$ for $s \in [L]$ are reachable through the trips $\Upsilon_{h,j}^{\mathrm{v}}$ (see Figure 16). Note that all these trips arrive in B at time $t_{\mathrm{v}} + 9 + l$.

- Consider a variable x_i appearing in c_j and such that the associated literal has value TRUE according to the assignment α satisfying Φ. The bottom clause node c_j^1, the middle nodes d_j^r for $r \in [l]$, and all bottom tail nodes $u_{j,i}^r$ for $r \in [L]$ are reachable through the trips Υ_i^{f}, T_i^{a}, Υ_i^{t}, and $\Upsilon_{j,i}^{\mathrm{u}}$ or through the trips Υ_i^{t}, Υ_i^{a}, Υ_i^{f}, and $\Upsilon_{j,i}^{\mathrm{u}}$ (see Figure 16).

- All bottom tail nodes $u_{h,i}^r$ with $h \neq j$ and $r \in [L]$ are reachable through the trips $\Upsilon_{h,i}^{\mathrm{u}}$, which all "meet" in the bottom hub B at time $t_{\mathrm{v}} + 14 + l$, that is, later than the bottom-clause trips $\Upsilon_{h,j}^{\mathrm{v}}$ (see Figure 16).

Let us now prove that, for any node $x \in X$ and for any node $v \in V$, $v \in \mathcal{R}_G(x)$. First we prove that, for each $j \in [m]$, the top clause node c_j^2 is reachable at time $t_{\mathrm{v}} + 5$ from each node in X.

- The bottom hub B and all the bottom tail nodes $u_{p,q}^r$ and $v_{p,q}^r$ can reach B at time at most $t_{\mathrm{v}} - l + 3$ through the trips $T_{p,q}^{\mathrm{u}}$ and $T_{p,q}^{\mathrm{v}}$, and, hence, can reach the top clause node c_j^2 at time $t_{\mathrm{v}} + 5$ through the trips $T_{p,q}^{\mathrm{v}}$ (see Figure 15).

- The top hub U and all the top tail nodes $w_{p,q}^r$ can reach c_j^2 at time $t_{\mathrm{v}} + 5$ through the trips $T_{p,q}^{\mathrm{w}}$, which all "meet" in U at time $t_{\mathrm{v}} + 3$ (see Figure 15).

- The bottom clause nodes c_h^1, with $h \neq j$, and the middle nodes d_h^r for $r \in [l]$ can reach the top clause node c_j^2 at time $t_{\mathrm{v}} + 5$ through the trips $T_{h,j}^{\mathrm{v}}$ (see Figure 15).

- The bottom clause node c_j^1 and the middle nodes d_j^r for $r \in [l]$ can reach c_j^2 at time $t_{\mathrm{v}} + 5$ through the trips T_i^{t}, Υ_i^{a}, and T_i^{f} or through the trips T_i^{f}, T_i^{a}, and T_i^{t} (see Figure 15), where x_i is a variable whose truth assignment satisfies the clause c_j (note that the satisfiability of the formula Φ is also required here).

Now we show that, by starting from c_j^2 at time $t_{\mathrm{v}} + 5$, it is possible to reach the following set of nodes S_j: all nodes in X (since we already proved that these nodes are reachable from c_j^2, starting at time $t_{\mathrm{v}} + 7 > t_{\mathrm{v}} + 5$), the head nodes e_j^r and g_j^r through the trips $T_{j,r}^{\mathrm{w}}$ and $T_{j,r}^{\mathrm{v}}$, and all variable nodes of the gadget associated to a variable x_i appearing in c_j through the trips Υ_i^{f}, T_i^{a}, and Υ_i^{t} or through the trips Υ_i^{t}, Υ_i^{a}, and Υ_i^{f} (see Figure 16). Notice that $\bigcup_{j \in [m]} S_j = V$, and this concludes the proof of the claim. □

We now determine a lower bound on the τ-reachability by counting the number of nodes temporally reachable from different sources.

- From the nodes in X, that is the $3mL + m(m+2)L$ top and bottom tail nodes, the ml middle nodes, the 2 hub nodes, and the m bottom clause nodes, it is possible to reach each node in V. This adds $|X| \cdot |V| = |V|^2 - (|V| - |X|) \cdot |V|$ to the τ-reachability. Note that $|X| = Lm(m+5) + ml + 2 + m$.

- From the nodes in $V \setminus X$, that is the $7n$ variable nodes, the $m(m+2)$ head nodes, and the m top clause nodes it is possible to reach the nodes in X. This adds $(7n + m(m+3)) \cdot |X| = (|V| - |X|) \cdot |X|$ to the τ-reachability.

Hence, the τ-reachability is at least equal to $|V|^2-(|V|-|X|)^2 = |V|^2-(7n+m(m+3))^2 = Q$ (note that $|V| = 2+7n+m(L+1)(m+4)+ml+mL = (Lm(m+5)+ml+2+m)+(7n+m(m+3)) = |X|+(|V|-|X|)$).

Bounding reachability when Φ is not satisfiable. Let τ be any trip temporalisation of the trip network $(D,\mathbb{T})$ and let $G = G[D,\mathbb{T},\tau]$ be the temporal graph induced by τ.

Claim 3 *If the τ-reachability is at least equal to Q, then, for any $v \in V$ and for any bottom/top tail node x, we have $v \in \mathcal{R}_G(x)$ and $x \in \mathcal{R}_G(v)$.*

Proof. Without loss of generality, we prove the claim in the case in which $x \in A^{\mathrm{u}}_{j,i}$, for some clause c_j of Φ with $j \in [m]$ and for some variable x_i that appears (positive or negative) in c_j (the proofs of the other cases are similar). First of all observe that all nodes in $A^{\mathrm{u}}_{j,i}$ have the same reachability set and belong to the same reachability sets. Formally, for any two nodes $u^r_{j,i}$ and $u^s_{j,i}$ in $A^{\mathrm{u}}_{j,i}$ with $r,s \in [L]$, we have that $\mathcal{R}_G(u^r_{j,i}) = \mathcal{R}_G(u^s_{j,i})$, and that, for any node $v \in V$, $u^r_{j,i} \in \mathcal{R}_G(v)$ if and only if $u^s_{j,i} \in \mathcal{R}_G(v)$. This is due to the fact the bottom-variable trips $T^{\mathrm{u}}_{j,i}$ and $\reflectbox{$T$}^{\mathrm{u}}_{j,i}$ are the only trips passing through the nodes in $A^{\mathrm{u}}_{j,i}$. This observation implies that if there exists $v \in V$ such that either $v \notin \mathcal{R}_G(u^r_{j,i})$ or $u^r_{j,i} \notin \mathcal{R}_G(v)$ for some bottom tail node $u^r_{j,i}$, then the τ-reachability is at most $|V|^2 - L < Q$. Hence, the claim follows. $\square$

Let us now consider the following time constraints that, as a consequence of the above claim, need to be satisfied by τ, if the τ-reachability is at least equal to Q. For the sake of brevity, we will give a detailed proof of the first constraint only, since the proofs of the other ones are similar: intuitively, these proofs are based on the fact that the connections provided by some trips between two nodes use the minimum number of edges (that is, they are shortest paths with respect to the number of hops).

C1 For all clauses c_j of Φ with $j \in [m]$ and for all variables x_i that appear (positive or negative) in c_j, all trips $T^{\mathrm{w}}_{j,i}$ are assigned the same starting time t^{w}, that is, $\tau(T^{\mathrm{w}}_{j,i}) = t^{\mathrm{w}}$ (this implies that all these trips reach node U at time $t^{\mathrm{U}} = t^{\mathrm{w}}+L$ and the node c^2_j at time $t^{\mathrm{U}}+2$). This constraint is needed in order to have any top tail node able to reach any head node at the end of a top trip. Indeed, if there exists two trips $T^{\mathrm{w}}_{j_1,i_1}$ and $T^{\mathrm{w}}_{j_2,i_2}$, with $j_1, j_2 \in [m]$, $i_1, i_2 \in [n]$, and the variable x_{i_1} (respectively, x_{i_2}) appearing (positive or negative) in the clause c_{j_1} (respectively, c_{j_2}), such that $\tau(T^{\mathrm{w}}_{j_1,i_1}) > \tau(T^{\mathrm{w}}_{j_2,i_2})$, since there is no trip connecting U to $c^2_{j_2}$ by using less than two edges, any top tail node in $A^{\mathrm{w}}_{j_1,i_1}$ reaches $c^2_{j_2}$ at time $\tau(T^{\mathrm{w}}_{j_1,i_1})+L+2 > \tau(T^{\mathrm{w}}_{j_2,i_2})+L+2$, thus implying that it cannot reach the head node $e^{i_2}_{j_2}$ (since the edge $(c^2_{j_2}, e^{i_2}_{j_2})$ appears at time $\tau(T^{\mathrm{w}}_{j_2,i_2})+L+2$). Because of the previous claim, this contradicts the assumption that the τ-reachability is at least equal to Q.

C2 For all clauses c_j of Φ with $j \in [m]$ and for all variables x_i that appear (positive or negative) in c_j, all trips $\reflectbox{$T$}^{\mathrm{w}}_{j,i}$ are assigned the same starting time $t^{\mathrm{w,s}}$, that is, $\tau(\reflectbox{$T$}^{\mathrm{w}}_{j,i}) = t^{\mathrm{w,s}}$ (this implies that all these trips reach node U at time $t^{\mathrm{U,s}} = t^{\mathrm{w,s}}+3$). This constraint is needed in order to have any head node at the end of a top trip able to reach any top tail node.

C3 $t^{\mathrm{U}} \le t^{\mathrm{U,s}}$. This constraint is needed in order to have any top tail node in $A^{\mathrm{w}}_{j_1,i_1}$ able to reach any top tail node in $A^{\mathrm{w}}_{j_2,i_2}$, by first using the trip $T^{\mathrm{w}}_{j_1,i_1}$ (in order to reach U at time t^{U}, as stated in C1) and then using the trip $\reflectbox{$T$}^{\mathrm{w}}_{j_2,i_2}$ (which passes through U at time $t^{\mathrm{U,s}}$, as stated in C2).

C4 For each clause c_j of Φ with $j \in [m]$ and for any $h \in [m]$ with $h \neq j$, $\tau(T^{\mathrm{v}}_{j,h}) = t^{\mathrm{U}} - L - l$. This constraint is needed in order to have any top tail node able to reach any head node at the end of a bottom-clause trip and any bottom tail node able to reach any head node at the end of a top trip.

C5 For each clause c_j of Φ with $j \in [m]$ and for any $h \in [m]$ with $h \neq j$, $\tau(\Upsilon^{\mathrm{v}}_{j,h}) = t^{\mathrm{U,s}} - 3$. This constraint is needed in order to have any head node able to reach both any top tail node and any bottom tail node.

C6 For each clause c_j of Φ with $j \in [m]$ and for each variable x_i that appears (positive or negative) in c_j, $\tau(T^{\mathrm{u}}_{j,i}) \leq t^{\mathrm{U}} - L - l$. This constraint is needed in order to have any bottom tail node able to reach any head node at the end of a bottom-clause trip.

C7 For each clause c_j of Φ with $j \in [m]$ and for each variable x_i that appears (positive or negative) in c_j, $\tau(\Upsilon^{\mathrm{u}}_{j,i}) \geq t^{\mathrm{U,s}} - 2$. This constraint is needed in order to have any head node at the end of a bottom-clause trip able to reach any bottom tail node.

C8 For each clause c_j of Φ with $j \in [m]$, for each variable x_i that appears (positive or negative) in c_j, and for any $h \in [m]$ with $h \neq j$, $\tau(T^{\mathrm{v}}_{j,h}), \tau(T^{\mathrm{u}}_{j,i}) \leq t^{\mathrm{U,s}} - L - l - 4$. These constraints are needed in order to have any bottom tail node able to reach any top tail node.

The above constraints have been derived by using the fact that $Q > |V|^2 - L$. We now take advantage of the fact that $Q > |V|^2 - (m+2)l$. Note that, since Φ is not satisfiable, for any truth-assignment to the variables of Φ, there must exist a clause which is not satisfied by the assignment. Let us then consider the following truth-assignment α, which is derived from τ. For each variable gadget corresponding to a variable x_i, $\alpha(x_i) = \textsc{True}$ if $\tau(T^{\mathrm{t}}_i) \leq \tau(T^{\mathrm{f}}_i)$, otherwise $\alpha(x_i) = \textsc{False}$. Note that from the paragraph about the activation of pairs of variable nodes, it follows that if $\alpha(x_i) = \textsc{True}$ (respectively, $\alpha(x_i) = \textsc{False}$), then τ does not activate (f^1_i, f^2_i) (respectively, (t^1_i, t^2_i)). Let c_{j_α} be a clause which is not satisfied by α. We now show that the middle nodes $d^k_{j_\alpha}$, for $k \in [l]$, cannot reach the head nodes connected to $c^2_{j_\alpha}$. Intuitively, the main reason is that $c^2_{j_\alpha}$ cannot be reached from the middle nodes through any of the variable gadgets associated to the variables appearing in c_{j_α}, as τ does not activate, in these gadgets, the pair connected to $c^1_{j_\alpha}$ and $c^2_{j_\alpha}$, and no other temporal path is possible. More formally, let us analyse all possible temporal paths from a middle node x (with $x = d^k_{j_\alpha}$, for some $k \in [l]$) to a head node h connected to $c^2_{j_\alpha}$.

- Going through a node c^2_k with $k \neq j_\alpha$ is not allowed by the above time constraints (in particular, by the synchronization of forward bottom-clause trips and forward top trips at $c^2_{j_\alpha}$ and c^2_k). Indeed, each temporal path from x to c^2_k reaches c^2_k using either a top forward trip or a bottom-clause forward trip, which both arrive in c^2_k at time $t^{\mathrm{U}} + 2$ according to time constraints C1 and C4. Although $c^2_{j_\alpha}$ might be τ-reachable from c^2_k, the arrival time will be greater than $t^{\mathrm{U}} + 2$ while the trip to h depart from $c^2_{j_\alpha}$ at time $t^{\mathrm{U}} + 2$.

- Using any trip to go to B and then reach h through a node c^1_k with $k \neq j_\alpha$ is also not possible. Let us suppose we use a backward bottom-variable or bottom-clause trip T_1 to go from x to B, arrive in B at time t, and then use a forward bottom-variable or bottom-clause trip T_2 to reach c^1_k, and let t' be the time T_2 goes through B. Clearly, $t' \geq t$. Because of the temporal constraints C5 and C7, we have that $t \geq t^{\mathrm{U,s}} + l$, which implies $t \geq t^{\mathrm{U}} + l$ because of constraint C3. However, because of temporal constraints C4 and C6 we have that $t' \leq t^{\mathrm{U}} - l$ in contradiction with $t \leq t'$.

- Going through a variable node t_i^1, if we assume that x_i appears positive in c_{j_α}, is not possible either. As mentioned before the choice of c_{j_α} implies that τ does not activate pair (t_i^1, t_i^2) and the path of length four through the variable gadget for x_i from t_i^1 to t_i^2 is not τ-compatible. We could consider reaching f_i^2 and then a clause node c_k^2 but the situation would be similar as in the first case. We could finally consider to go from t_i^1 to another clause node c_k^1 such that x_i also appears positive in c_k. Let Υ_k denote the backward bottom-variable trip allowing to go from t_i^1 to c_k^1 and let t denote the time when it arrives in c_k^1. The time constraint C7 then imply that $t = \tau(\Upsilon_k) + 1 \geq t^{\mathrm{U,s}} - 1$. Let T be a bottom-variable or bottom-clause trip that we use to leave c_k^1 and later reach h. T has to arrive in c_k^1 at $t' \geq t$. Since $t' = \tau(T) + L + l + 1$, from the time constraint C8 it follows that $t' \leq t^{\mathrm{U,s}} - 3$ in contradiction with $t' \geq t$.

- Going through a variable node f_i^1, if we assume that x_i appears negative in c_{j_α}, is not possible either for similar reasons.

We have thus proved that no head connected to $c_{j_\alpha}^2$ is τ-reachable from any middle node between B and $c_{j_\alpha}^1$: this implies that the reachability of τ is at most $|V|^2 - l(m+2) < Q$. This concludes the proof of the theorem. □

Our last result shows that the maximum temporal reachability obtainable in symmetric strongly temporalisable trip networks is quadratic with respect to the number of nodes. The general idea to prove this result is to find a somewhat central trip, and then schedule trips so that a constant fraction of nodes can reach the central trip and a constant fraction of nodes are reached from the central trip relying on Facts 2 and 3. We will find this central trip as a centroid in a weighted tree.

Let us, then, first recall the definition of centroid. Given a node-weighted tree R, the weight of R is defined as the sum of the weights of its nodes. We then define a *weighted centroid* of a tree R of weight K as a node c such that the removal of c disconnects R into subtrees of weight $2K/3$ at most. Such a centroid can be found efficiently as stated below.

Lemma 1 (Folklore) *Given a node-weighted tree R, a centroid node c can be found in linear time. Moreover, if the weight of R is K and the centroid c has weight $2K/3$ at most, then there exists a partition P_1, P_2 of its pending subtrees such that both $P_1 \cup \{c\}$ and P_2 have total weight $2K/3$ at most. Such a partition can be computed at the cost of sorting the subtrees by non-decreasing weight.*

Proof. Note that the classical algorithm for finding a centroid in an unweighted tree can easily be adapted to the weighted case. Recall that it consists in starting from any node v. If it is not a centroid, then move to a neighbor whose subtree has weight greater than $K/2$ and repeat the test until finding a centroid. The partition P_1, P_2 is obtained by trying to add subtrees in P_1 one after another by non-decreasing weight and stopping as soon as $P_1 \cup \{c\}$ has weight $K/3$ or more. □

Theorem 9 *Let $(D, \mathbb{T})$ be a symmetric and strongly temporalisable trip network. Then there exists a schedule S such that the S-reachability of $(D, \mathbb{T})$ is at least a fraction $2/9$ of all node pairs. Such a schedule can be computed in polynomial time.*

Proof. Without loss of generality, we can assume that all trips in $\mathbb{T}$ are distinct. If this is not the case, we can keep only one of multiple copies of the same trip: this can only reduces the reachability of the modified trip network. We can then consider trips in pairs $(T, \reflectbox{$T$})$, where $\reflectbox{$T$}$ is the reverse trip of T, and we denote by $\mathbb{TP}$ the set of such pairs. For any trip T, we also denote by $V(T) \subseteq V$ the set of nodes which T (and $\reflectbox{$T$}$) passes through. Finally, we assign arbitrarily each node $v \in V$ to a single trip pair $(T, \reflectbox{$T$})$ such that $v \in V(T)$, and let n_T denote the number of nodes assigned to the pair $(T, \reflectbox{$T$})$.

We now define the *transfer* undirected graph $\mathbb{P} = (\mathbb{TP}, \mathbb{EP})$, where two trip pairs are connected when they share a node, that is, $\{(T_1, \reflectbox{$T$}_1), (T_2, \reflectbox{$T$}_2)\} \in \mathbb{EP}$ if and only if $V(T_1) \cap V(T_2) \neq \emptyset$. Let M denote the multidigraph induced by $(D, \mathbb{T})$. According to Corollary 1, M is strongly connected and, hence, $\mathbb{P}$ is connected. We then compute in linear time a weighted spanning tree R of $\mathbb{P}$, where each trip pair $(T, \reflectbox{$T$})$ is weighted by the number n_T. Note that the weight of R is the number $n = |V|$ of nodes in D. We then find a centroid $(C, \reflectbox{$C$})$ of R according to Lemma 1.

First suppose that $(C, \reflectbox{$C$})$ has weight greater than $2n/3$. Let S be any schedule of $(D, \mathbb{T})$ that starts with C followed by $\reflectbox{$C$}$. Then, we have that, for any u and v in $V(C)$, v is S-reachable from u, that is, the S-reachability is greater than $4n^2/9$. The theorem follows.

Conversely, let us suppose that $(C, \reflectbox{$C$})$ has weight $2n/3$ at most. According to Lemma 1, we then consider a partition P_1, P_2 of $R \setminus \{(C, \reflectbox{C})\}$ such that both $P_1 \cup \{(C, \reflectbox{C})\}$ and P_2 have weight $2n/3$ at most. For $i = 1, 2$, let $V(P_i)$ denote the set of nodes assigned to T, for some trip T such that $(T, \reflectbox{$T$}) \in P_i$. Let $B_1, \ldots, B_{|P_1|}$ be the subtrees in P_1, sorted in an arbitrary way. For each i with $i \in [|P_1|]$, the subtree B_i corresponds to a strongly connected set V_i of D. Moreover, V_i must contain a node $u_i \in V(C)$ as some trip pair of B_i is connected to $(C, \reflectbox{$C$})$ in $\mathbb{P}$. We can then define a schedule S_i of the trip pairs in B_i according to Fact 3 so that, for any node v in B_i, u_i is S_i-reachable from v. Hence, the schedule $S = S_1, \ldots, S_{|P_1|}, C, \reflectbox{C}$ is such that, for any $u \in V(P_1) \cup V(C)$ and $c \in V(C)$, c is S-reachable from u. Similarly, by reasoning on the subtrees in P_2, we can extend S, so that, for any $u \in V(P_2)$ and $c \in V(C)$, u is S-reachable from c. In other words, the final schedule S is such that, for any $u \in V(P_1) \cup V(C)$ and $v \in V(P_2)$, v is S-reachable from u.

Let n_2 denote the weight of P_2, that is, $n_2 = |V(P_2)|$. As both $P_1 \cup \{(C, \reflectbox{C})\}$ and P_2 have weight $2n/3$ at most, we have that $n/3 \leq n_2 \leq 2n/3$. Hence, the number of pairs of nodes u and v such that $u \in V(P_1) \cup V(C)$ and $v \in V(P_2)$ is at least $(n - n_2)n_2 \geq \frac{2}{9}n^2$ for $n/3 \leq n_2 \leq 2n/3$. The theorem thus follows. □

5 Open problems

From a theoretical point of view, it would be interesting to close the gap between Theorems 6 and 2 with regard to the inapproximability of MRTT in a strongly temporalisable network. From a more applicative point of view, it would be worth exploring other restrictions of the problem where constant approximation is possible. For example, we leave as a future work the study of trip networks with single edge trips as they can be seen as a variation in directed graphs of label-connectivity as defined in [23].

Finally, an interesting generalisation consists in allowing variable waiting times in-between two consecutive edges of a trip. In other words, a temporalisation would then assign an appearing time to each edge of a trip so that the trip becomes a valid temporal walk: each edge appears after the arrival time of the previous edge.

References

[1] N. Alon, A. Gyárfás, and M. Ruszinkó. Decreasing the diameter of bounded degree graphs. *Journal of Graph Theory*, 35(3):161–172, 2000.

[2] N. Alon, R. Yuster, and U. Zwick. Color coding. In *Encyclopedia of Algorithms*, pages 335–338. Springer, 2016.

[3] G. Ausiello, A. Marchetti-Spaccamela, P. Crescenzi, G. Gambosi, M. Protasi, and V. Kann. *Complexity and approximation: combinatorial optimization problems and their approximability properties*. Springer, 1999.

[4] Richard Bellman. The theory of dynamic programming. *Bulletin of the American Mathematical Society*, 60(6):503–516, 1954.

[5] Elisabetta Bergamini, Pierluigi Crescenzi, Gianlorenzo D'Angelo, Henning Meyerhenke, Lorenzo Severini, and Yllka Velaj. Improving the betweenness centrality of a node by adding links. *ACM Journal of Experimental Algorithmics*, 23, 2018.

[6] Kenneth A. Berman. Vulnerability of scheduled networks and a generalization of menger's theorem. *Networks*, 28(3):125–134, 1996.

[7] S. Bhadra and A. Ferreira. Complexity of connected components in evolving graphs and the computation of multicast trees in dynamic networks. In *ADHOC-NOW*, pages 259–270, 2003.

[8] F. Brunelli, P. Crescenzi, and L. Viennot. On computing pareto optimal paths in weighted time-dependent networks. *Information Processing Letters*, 168:106086, 2021.

[9] A. Casteigts, P. Flocchini, W. Quattrociocchi, and N. Santoro. Time-varying graphs and dynamic networks. *IJPEDS*, 27(5):387–408, 2012.

[10] A. Casteigts, A.-S. Himmel, H. Molter, and P. Zschoche. Finding temporal paths under waiting time constraints. In *ISAAC*, pages 30:1–30:18, 2020.

[11] Eddie Cheng, Jerrold W Grossman, and Marc J Lipman. Time-stamped graphs and their associated influence digraphs. *Discrete Applied Mathematics*, 128(2):317–335, 2003.

[12] F. Corò, G. D'Angelo, and C.M. Pinotti. Adding edges for maximizing weighted reachability. *Algorithms*, 13(3):68, 2020.

[13] F. Corò, G. D'Angelo, and Y. Velaj. Recommending links to maximize the influence in social networks. In *IJCAI*, pages 2195–2201, 2019.

[14] Pierluigi Crescenzi, Gianlorenzo D'Angelo, Lorenzo Severini, and Yllka Velaj. Greedily improving our own closeness centrality in a network. *ACM Trans. Knowl. Discov. Data*, 11(1), 2016.

[15] A. Deligkas and I. Potapov. Optimizing reachability sets in temporal graphs by delaying. In *AAAI*, pages 9810–9817, 2020.

[16] E.D. Demaine and M. Zadimoghaddam. Minimizing the diameter of a network using shortcut edges. In *SWAT*, pages 420–431, 2010.

[17] J. Enright, K. Meeks, and F. Skerman. Assigning times to minimise reachability in temporal graphs. *Journal of Computer and System Sciences*, 115:169–186, 2021.

[18] J.A. Enright, K. Meeks, G.B. Mertzios, and V. Zamaraev. Deleting edges to restrict the size of an epidemic in temporal networks. In *MFCS*, pages 57:1–57:15, 2019.

[19] F. Frati, S. Gaspers, J. Gudmundsson, and L. Mathieson. Augmenting graphs to minimize the diameter. *Algorithmica*, 72(4):995–1010, 2015.

[20] Michael L. Fredman, János Komlós, and Endre Szemerédi. Storing a sparse table with 0(1) worst case access time. *J. ACM*, 31(3):538–544, 1984.

[21] Esther Galby, Paloma T. Lima, and Bernard Ries. Reducing the domination number of graphs via edge contractions and vertex deletions. *Discrete Mathematics*, 344(1):112169, 2021.

[22] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

[23] F. Göbel, J. Orestes Cerdeira, and H.J. Veldman. Label-connected graphs and the gossip problem. *Discrete Mathematics*, 87(1):29–40, 1991.

[24] F. Harary and G. Gupta. Dynamic graph models. *Mathematical and Computer Modelling*, 25(7):79–87, 1997.

[25] S. Mitchell Hedetniemi, S.T. Hedetniemi, and A.L. Liestman. A survey of gossiping and broadcasting in communication networks. *Networks*, 18(4):319–349, 1988.

[26] P. Holme and J. Saramäki. Temporal networks. *Physics Reports*, 519(3):97–125, 2012.

[27] Petter Holme and Jari Sarämaki. *Temporal Networks*. Springer, 2013.

[28] D. Kempe, J. Kleinberg, and A. Kumar. Connectivity and inference problems for temporal networks. *Journal of Computer and System Sciences*, 64(4):820 – 842, 2002.

[29] Eun Jung Kim, Martin Milanič, Jérôme Monnot, and Christophe Picouleau. Complexity and algorithms for constant diameter augmentation problems. *Theoretical Computer Science*, 2021.

[30] M. Latapy, T. Viard, and C. Magnien. Stream graphs and link streams for the modeling of interactions over time. *Social Netw. Analys. Mining*, 8(1):61:1–61:29, 2018.

[31] John M. Lewis and Mihalis Yannakakis. The node-deletion problem for hereditary properties is np-complete. *Journal of Computer and System Sciences*, 20(2):219–230, 1980.

[32] Naoki Masuda and Renaud Lambiotte. *A Guide to Temporal Networks*. World Scientific, 2016.

[33] G.B. Mertzios, O. Michail, I. Chatzigiannakis, and P.G. Spirakis. Temporal network optimization subject to connectivity constraints. In *ICALP*, pages 657–668, 2013.

[34] G.B. Mertzios, O. Michail, and P.G. Spirakis. Temporal network optimization subject to connectivity constraints. *Algorithmica*, 81(4):1416–1449, 2019.

[35] George B. Mertzios, Hendrik Molter, Malte Renken, Paul G. Spirakis, and Philipp Zschoche. The complexity of transitively orienting temporal graphs. In *MFCS*, volume 202 of *LIPIcs*, pages 75:1–75:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.

[36] O. Michail. An introduction to temporal graphs: An algorithmic perspective. *Internet Mathematics*, 12(4):239–280, 2016.

[37] Hendrik Molter, Malte Renken, and Philipp Zschoche. Temporal reachability minimization: Delaying vs. deleting. In *MFCS*, volume 202 of *LIPIcs*, pages 76:1–76:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.

[38] Jeanette P. Schmidt and Alan Siegel. The spatial complexity of oblivious k-probe hash functions. *SIAM J. Comput.*, 19(5):775–786, 1990.

[39] H. Wu, J. Cheng, Y. Ke, S. Huang, Y. Huang, and H. Wu. Efficient Algorithms for Temporal Path Computation. *IEEE Transactions on Knowledge and Data Engineering*, 28:2927–2942, 2016.

A Proof of Theorem 3

Similarly to the proof of Theorem 2, we will make use of the gap technique. Suppose by contradiction that there exists a $r(\cdot)$-approximation algorithm $\mathcal{A}$ for the SS-MRTT problem, for some $\epsilon \in (0,1)$ and for some function $r(n)$ of the number n of nodes that satisfies $r(n) \leq cn^{1-\epsilon}$ for some constant c. We will now show that it is possible to exploit such an algorithm in order to solve in polynomial time the O2O-RTT problem, which would imply that P = NP (because of Theorem 1). Let us consider an instance $\langle (D = (V,E), \mathbb{T}), s, t \rangle$ of the O2O-RTT problem, where $V = \{s = v_1, \ldots, v_n = t\}$. Without loss of generality, we assume that $n > c+1$. We define an instance $(D' = (V', E'), \mathbb{T}')$ of SS-MRTT as follows.

- $V' = V \cup \{v_{n+i} : i \in [K]\}$ with $K = \lceil cn^{2/\epsilon} \rceil$.
- $E' = E \cup \{(t, v_{n+i}) : i \in [K]\}$.

The trip collection $\mathbb{T}'$ is then defined as the union of $\mathbb{T}$ with all the one-edge trips corresponding to the edges in $E' \setminus E$.

Consider an optimal temporalisation τ^* of $(D', \mathbb{T}')$: the maximum reachability of s is thus $\texttt{opt} = |\mathcal{R}_{G[D',\mathbb{T}',\tau^*]}(s)|$. Moreover, let x be the value of the temporalisation computed by the approximation algorithm $\mathcal{A}$: hence, $\frac{\texttt{opt}}{r(n')} \leq x \leq \texttt{opt}$ where $n' = n + K$.

If $(D, \mathbb{T})$ is (s,t)-temporalisable, then $\texttt{opt} \geq K+2$. Indeed, the very same temporalisation can be extended to $(D', \mathbb{T}')$, by assigning to all the new one-edge trips a starting time greater than the arrival time in t, so that all the K new out-neighbors of t are reachable. Note that $\texttt{opt} \geq K+2$ implies that $x \geq \frac{\texttt{opt}}{r(n')} \geq \frac{K+2}{r(n')} \geq \frac{cn^{2/\epsilon}}{r(n')}$. Since $n' = n + K$, we have that $n' < n + cn^{2/\epsilon} + 1 < n^{1+2/\epsilon}$ (since $n > c+1$ and $\epsilon \in (0,1)$). Since r is non-decreasing, we get $x \geq \frac{cn^{2/\epsilon}}{r(n')} > \frac{cn^{2/\epsilon}}{r(n^{1+2/\epsilon})} \geq \frac{cn^{2/\epsilon}}{c(n^{1+2/\epsilon})^{1-\epsilon}} = n^{1+\epsilon} > n$. Hence, if $(D, \mathbb{T})$ is (s,t)-temporalisable, then $x > n$. On the other hand, if $(D, \mathbb{T})$ is not (s,t)-temporalisable, then $\texttt{opt} < n$, since none of the new out-neighbors of t are

τ'-reachable from s (without passing through t), for any temporalisation τ' of $(D', \mathbb{T}')$. Hence, $x \leq \texttt{opt} < n$.

We can conclude that $(D, \mathbb{T})$ is (s,t)-temporalisable if and only if the value x of the solution computed by the approximation algorithm $\mathcal{A}$ is greater than n. This implies that the O2O-RTT problem is solvable in polynomial time, and the theorem is proved.