Traveling Salesman Problems in Temporal Graphs⋆

Othon Michail[1] and Paul G. Spirakis[1,2]

[1] Computer Technology Institute & Press "Diophantus" (CTI), Patras, Greece
[2] Department of Computer Science, University of Liverpool, UK
Email: michailo@cti.gr, P.Spirakis@liverpool.ac.uk

Abstract. In this work, we introduce the notion of time to some well-known combinatorial optimization problems. In particular, we study problems defined on *temporal graphs*. A temporal graph $D = (V, A)$ may be viewed as a time-sequence $G_1, G_2 \ldots, G_l$ of static graphs over the same (static) set of nodes V. Each $G_t = D(t) = (V, A(t))$ is called the *instance of D at time t* and l is called the *lifetime of D*. Our main focus is on analogues of *traveling salesman problems* in temporal graphs. A sequence of time-labeled edges (e.g. a tour) is called *temporal* if its labels are strictly increasing. We begin by considering the problem of exploring the nodes of a temporal graph as soon as possible. In contrast to the positive results known for the static case, we prove that, it cannot be approximated within cn, for some constant $c > 0$, in general temporal graphs and within $(2 - \varepsilon)$, for every constant $\varepsilon > 0$, in the special case in which $D(t)$ is connected for all $1 \leq t \leq l$, both unless $\mathbf{P} = \mathbf{NP}$. We then study the temporal analogue of TSP(1,2), abbreviated TTSP(1,2), where, for all $1 \leq t \leq l$, $D(t)$ is a complete weighted graph with edge-costs from $\{1, 2\}$ and the cost of an edge may vary from instance to instance. The goal is to find a minimum cost temporal TSP tour. We give several *polynomial-time approximation algorithms* for TTSP(1,2). Our best approximation is $(1.7 + \varepsilon)$ for the generic TTSP(1,2) and $(13/8 + \varepsilon)$ for its interesting special case in which the lifetime of the temporal graph is restricted to n. In the way, we also introduce temporal versions of other fundamental combinatorial optimization problems, for which we obtain polynomial-time approximation algorithms and hardness results.

1 Introduction

A *temporal graph* is, informally speaking, a graph that changes with time. A great variety of both modern and traditional networks such as information and communication networks, social networks, transportation networks, and several physical systems can be naturally modeled as temporal graphs.

⋆ Supported in part by the (i) project FOCUS, "ARISTEIA" Action, OP EdLL, EU and Greek National Resources, (ii) FET EU IP project MULTIPLEX under contract no 317532, and (iii) School of EEE/CS of the Univ. of Liverpool. Full version: http://ru1.cti.gr/aigaion/?page=publication&kind=single&ID=1051

In this work, we restrict attention to *discrete time*. This is totally plausible when the dynamicity of the system is inherently discrete, which is for example the case in synchronous mobile distributed systems that operate in discrete rounds, but can also satisfactory approximate a wide range of continuous-time systems. Moreover, this choice gives to the resulting models and problems a purely combinatorial flavor. We also restrict attention to systems in which only the relationships between the participating entities may change and not the entities themselves. Therefore, in this paper, a temporal graph $D = (V, A)$ may always be viewed as a sequence $G_1, G_2 \ldots, G_l$ of static graphs over the same (static) set of nodes V. Each $G_t = D(t) = (V, A(t))$ is called the *instance of D at time t* and l is called the *lifetime of D*.

Though static graphs have been extensively studied, for their temporal generalization we are still far from having a concrete set of structural and algorithmic principles. Additionally, it is not yet clear how is the complexity of combinatorial optimization problems affected by introducing to them a notion of time. In an early but serious attempt to answer this question, Orlin [Orl81] observed that many dynamic languages derived from **NP**-complete languages can be shown to be **PSPACE**-complete. Among the other few things that we do know, is that the max-flow min-cut theorem holds with unit capacities for time-respecting paths [Ber96]. Additionally, Kempe *et al.* [KKK00] proved that, in temporal graphs, the classical formulation of Menger's theorem is violated and the computation of the number of node-disjoint s-t paths becomes **NP**-complete. In a very recent work [MMCS13], among other things, the authors achieved a reformulation of Menger's theorem which is valid for all temporal graphs and introduced several interesting cost minimization parameters for optimal temporal network design. Dutta *et al.* [DPR$^+$13], working on a distributed online dynamic network model, presented offline centralized algorithms for the k-gossip problem.

We make here one more step towards the direction of revealing the algorithmic principles of temporal graphs. In particular, we introduce the study of traveling salesman problems in temporal graphs, which, to the best of our knowledge, have not been considered before in the literature. Our main focus is on the Temporal Traveling Salesman Problem with Costs One and Two abbreviated TTSP(1,2) throughout the paper. In this problem, we are given a temporal graph $D = (V, A)$ every instance of which is a complete graph, i.e. $D(t) = (V, A(t))$ is complete for all $1 \leq t \leq l$. Moreover, the edges of every $D(t)$ are weighted according to a cost function $c : A \rightarrow \{1, 2\}$. Observe that A is a set of *time-edges* (e, t), where e is an edge and t is the time at which e appears. So, the cost function c is allowed to assign different cost values to different instances of the same edge, therefore, in this model, *costs are dynamic* in nature. We are asked to find a *Temporal TSP tour* (abbreviated *TTSP tour*) of minimum total cost. A TSP tour $(u_1, t_1, u_2, t_2, \ldots, t_{n-1}, u_n, t_n, u_1)$ is temporal if $t_i < t_{i+1}$ for all $1 \leq i \leq n-1$. The cost of such a TSP tour is $\sum_{1 \leq i \leq n} c((u_i, u_{i+1}), t_i)$, where $u_{n+1} = u_1$. We should remark that, in general, the lifetime of D is not restricted and therefore it can be much greater than n. Whenever we restrict attention to limited lifetime, this will be explicitly stated. We should also emphasize that, throughout this work, *the entire temporal graph is provided to the centralized algorithms in advance*. It is useful to observe that TTSP(1,2) seems to be naturally closer to the ASYMMETRIC TSP(1,2) and this seems to hold independent of whether the temporal graph has directed or undirected instances. In most places we assume directed edge-sets, however keep in mind that the undirected case is not expected to be any simpler. Finally, note that ATSP(1,2) is a special case of TTSP(1,2) which implies that TTSP(1,2) is also **APX**-hard [PY93] and cannot be approximated within any factor less than 207/206 [KS13].

1.1 Our Approach-Contribution

We now summarize our approach to approximate TTSP(1,2). Note that all the approximation algorithms that we present in this work are *polynomial-time algorithms* on a binary encoding of the temporal graph, i.e. on $|\langle D \rangle|$. In the static case, one easily obtains a (3/2)-factor approximation for ATSP(1,2) by computing a perfect matching maximizing the number of ones and then patching the edges together arbitrarily. This works well, because such a minimum cost perfect matching can be computed in polynomial time in the static case. This was one of the first algorithms known for ATSP(1,2). Other approaches have improved the factor to the best currently known 5/4 [Blä04]. Unfortunately, as we shall see, even the apparently simple task of computing a matching maximizing the number of ones is not that easy in temporal graphs. In particular, we prove that computing a matching maximizing the number of ones and additionally satisfying the temporal condition that all its edges appear at distinct times is **NP**-hard. The reason that we insist on distinct times is that we can form a temporal TSP tour by patching the edges of a matching only if the edges of the matching can be strictly ordered in time. In fact, an additional requirement is that the edges of the matching should have time differences of at least two, so that we can always fit a patching-edge between two time-consecutive edges of the matching. We call the corresponding problem Max-TEM(≥ 2).

We naturally then search for good approximations for Max-TEM(≥ 2). We follow two main approaches. One is to reduce the problem to MAXIMUM INDEPENDENT SET (MIS) in $(k+1)$-claw free graphs and the other is to reduce it to k'-SET PACKING for some k and k' to be determined. The first approach gives a $(7/4+\varepsilon)$-approximation $(= 1.75+\varepsilon)$ for the generic TTSP(1,2) and a $(12/7+\varepsilon)$-approximation $(\approx 1.71+\varepsilon)$ for the special case of TTSP(1,2) in which the lifetime is restricted to n. The latter is obtained by approximating a temporal path packing instead of a matching. The second approach improves these to $1.7+\varepsilon$ for the general case and to $13/8+\varepsilon = 1.625+\varepsilon$ when the lifetime is n. In all cases, $\varepsilon > 0$ is a small constant (not necessarily the same in all cases) adopted from the factors of the approximation algorithms for independent set and set packing. We leave as an interesting open problem whether a (3/2)-factor for TTSP(1,2) or its special case with lifetime restricted to n is within reach.

Apart from TTSP(1,2) we also consider the TEMPORAL (NODE) EXPLORATION (abbreviated TEXP) problem, in which we are given a temporal graph (unweighted and non-complete) and the goal is to visit all nodes of the temporal graph, by possibly revisiting nodes, minimizing the arrival time (in the static case, appears as Graphic TSP in the literature). Though, in the static case, the decision version of the problem, asking whether a given graph is explorable, can be solved in linear time, we show that in the temporal case it becomes **NP**-complete. Additionally, in the static case, there is a $(3/2 - \varepsilon)$-approximation for undirected graphs [GSS11] and a $O(\log n / \log\log n)$ for directed [AGM$^+$10]. In contrast to these, we prove that there exists some constant $c > 0$ such that TEXP cannot be approximated within cn unless $\mathbf{P} = \mathbf{NP}$. Additionally, we prove that even the special case in which every instance of the temporal graph is connected, cannot be approximated within $(2 - \varepsilon)$, for every constant $\varepsilon > 0$, unless $\mathbf{P} = \mathbf{NP}$. On the positive side, we show that TEXP can be approximated within the *dynamic diameter* (definition in Section 2) of the temporal graph.

Finally, in the way to approaching the above two main problems, we also obtain several interesting side-results, such as a $[3/(5+\varepsilon)]$-approximation for Max-TEM(≥ 2), a $[1/(7/2+\varepsilon)]$-approximation for Temporal Path Packing (TPP) when the lifetime is restricted to n, and in the full paper a $(1/5)$-approximation for Max-TTSP and an inapproximability result stating that for any polynomial time computable function $\alpha(n)$, Temporal Cycle Cover cannot be approximated within $\alpha(n)$, unless $\mathbf{P} = \mathbf{NP}$. To the best of our knowledge, all the aforementioned temporal problems are first studied in this work.

In Section 2, we formally define the model of temporal graphs under consideration and provide all further necessary definitions. Section 2.1 presents formal definitions of all temporal problems that we consider in this work. In Section 3, we consider the Temporal Exploration problem. Then, in Section 4 we introduce and study the TTSP(1,2) problem in weighted temporal graphs.

2 Preliminaries

Definition 1. *A* temporal graph *(or* dynamic graph*) D is an ordered pair of disjoint sets (V, A) such that $A \subseteq \binom{V}{2} \times \mathbb{N}$ ($V^2 \backslash \{(u,u) : u \in V\}$ in case of a digraph). The set V is the set of* nodes *and the set A is the set of* time-edges*.*

A temporal (di)graph $D = (V, A)$ can be also viewed as a static (underlying) graph $G_D = (V, E)$, where $E = \{e : (e,t) \in A \text{ for some } t \in \mathbb{N}\}$ contains all edges that appear at least once, together with a labeling $\lambda_D : E \to 2^{\mathbb{N}}$ defined as $\lambda_D(e) = \{t : (e,t) \in A\}$ (we omit the subscript D when no confusion can arise). We denote by $\lambda(E)$ the multiset of all labels assigned by λ to G_D and by $\lambda_{\min} = \min\{l \in \lambda(E)\}$ ($\lambda_{\max} = \max\{l \in \lambda(E)\}$) the minimum (maximum) label of D. We define the *lifetime* (or *age*) of a temporal graph D as $\alpha(D) = \lambda_{\max} - \lambda_{\min} + 1$. Note that in case $\lambda_{\min} = 1$ we have $\alpha(D) = \lambda_{\max}$.

For every time $t \in \mathbb{N}$, we define the *t-th instance of a temporal graph* $D = (V, A)$ as the static graph $D(t) = (V, A(t))$, where $A(t) = \{e : (e,t) \in A\}$ is the (possibly empty) set of all edges that appear in D at time t. A temporal graph $D = (V, A)$ may be also viewed as a *sequence of static graphs* $(G_1, G_2, \ldots, G_{\alpha(D)})$, where $G_i = D(\lambda_{\min} + i - 1)$ for all $1 \leq i \leq \alpha(D)$. Another, often convenient, representation of a temporal graph is the following. The *static expansion* of a temporal graph $D = (V, A)$ is a DAG $H = (S, E)$ defined as follows. If $V = \{u_1, u_2, \ldots, u_n\}$ then $S = \{u_{ij} : \lambda_{\min} - 1 \leq i \leq \lambda_{\max}, 1 \leq j \leq n\}$ and $E = \{(u_{(i-1)j}, u_{ij'}) : \text{if } (u_j, u'_j) \in A(i) \text{ for some } \lambda_{\min} \leq i \leq \lambda_{\max}\}$.

A *temporal* (or *time-respecting*) *walk* W of a temporal graph $D = (V, A)$ is an alternating sequence of nodes and times $(u_1, t_1, u_2, t_2, \ldots, u_{k-1}, t_{k-1}, u_k)$ where $(u_i u_{i+1}, t_i) \in A$, for all $1 \leq i \leq k-1$, and $t_i < t_{i+1}$, for all $1 \leq i \leq k-2$. We call $t_{k-1} - t_1 + 1$ the *duration* (or *temporal length*) of the walk W, t_1 its *departure time* and t_{k-1} its *arrival time*. A *journey* (or *temporal/time-respecting path*) J is a temporal walk with pairwise distinct nodes. A (u, v)-journey J is called *foremost from time* $t \in \mathbb{N}$ if it departs after time t and its arrival time is minimized. The *temporal distance* from a node u at time t (also called *time-node* (u, t)) to a node v is defined as the duration of a foremost (u, v)-journey from time t. We say that a temporal graph $D = (V, A)$ has *dynamic diameter* d, if d is the minimum integer for which it holds that the temporal distance from every time-node $(u, t) \in V \times \{0, 1, \ldots, \alpha(D) - d\}$ to every node $v \in V$ is at most d. A *temporal matching* of a temporal graph $D = (V, A)$ is a set of time-edges $M = \{(e_1, t_1), (e_2, t_2), \ldots, (e_k, t_k)\}$, such that $(e_i, t_i) \in A$, for all $1 \leq i \leq k$, $t_i \neq t_j$, for all $1 \leq i < j \leq k$, and $\{e_1, e_2, \ldots, e_k\}$ is a matching of G_D.

Similarly to weighted graphs we may define *weighted temporal graphs* by introducing a (temporal) cost function $c : A \to \mathcal{C}$, where $\mathcal{C}$ denotes the range of the costs, e.g. $\mathcal{C} = \mathbb{N}$. A temporal graph $D = (V, A)$ is called *complete* (*continuously connected*) if $D(t)$ is complete (connected, resp.) for all $1 \leq t \leq \alpha(D)$. In these cases, we may also say that D *has complete/connected instances*.

Throughout the text, unless otherwise stated, we denote by n the number of nodes of (temporal) (di)graphs. When no confusion may arise, we use the term *edge* for both undirected edges and arcs. Finally, a δ-factor (polynomial-time) approximation algorithm for a problem Π satisfies $\delta \geq 1$ if Π is a minimization problem and $\delta \leq 1$ if Π is a maximization problem.

2.1 Problem Definitions

TEMPORAL EXPLORATION - TEXP. Given a temporal graph $D = (V, A)$ and a source node $s \in V$, find a temporal walk that begins from s and visits all nodes minimizing the arrival time.

TTSP(1,2). Given a complete temporal graph $D = (V, A)$ and a cost function $c : A \to \{1, 2\}$ find a temporal TSP tour of minimum total cost.

Max-TEM($\geq k$). Given a temporal graph $D = (V, A)$ find a maximum cardinality temporal matching $M = \{(e_1, t_1), (e_2, t_2), \ldots, (e_h, t_h)\}$ satisfying that there is a permutation $t_{i_1}, t_{i_2}, \ldots, t_{i_h}$ of the t_js s.t. $t_{i_{(l+1)}} \geq t_{i_l} + k$ for all $1 \leq l \leq h-1$.

TEMPORAL PATH PACKING - TPP. We are given a temporal graph and we want to find time and node disjoint time-respecting paths maximizing the number of edges used. By time disjoint we require that they correspond to distinct intervals that differ by ≥ 2 in time.

3 Exploration of Temporal Graphs

In this section, we study the TEMPORAL EXPLORATION (TEXP) problem in (unweighted) temporal graphs. In contrast to several positive results known for the static case, we show that in temporal graphs the problem is quite hard. In particular, we show that the decision version of the problem is **NP**-complete and we give two hardness of approximation results for the optimization version, one for the generic case and another for the special case in which the temporal graph is continuously connected. On the positive side, we approximate the optimum of the generic instances within the dynamic diameter of the temporal graph.

3.1 Deciding Explorability is Hard in Temporal Graphs

Note that a walk in the (TEMPORAL) EXPLORATION is allowed to revisit nodes several times. Let us first focus on static graphs. Consider the decision version DEXP of EXPLORATION in which the goal is to decide whether a given graph is explorable. DEXP and finding an arbitrary solution can be solved in linear time for both undirected and directed static graphs. On the other hand, we prove that its temporal version, abbreviated DTEXP, is **NP**-complete.

3.2 Hardness of Approximate Temporal Exploration

Theorem 1. *There exists some constant $c > 0$ such that TEXP cannot be approximated within cn unless $\mathbf{P} = \mathbf{NP}$.*

The reason that we managed to obtain such a strong inapproximability result was that we were free to totally break at some point the connectivity of the temporal graph. This freedom is lost in continuously connected temporal graphs.

Theorem 2. *For every constant $\varepsilon > 0$, there is no $(2 - \varepsilon)$-approximation for TEXP in continuously (strongly) connected temporal graphs unless $\mathbf{P} = \mathbf{NP}$.*

Proof. The reduction is from HAMILTONIAN PATH (abbreviated HAMPATH). We prove that a $(2 - \varepsilon)$-factor approximation for TEXP in continuously connected temporal graphs could be used to decide HAMPATH. Let (G, s) be an instance of HAMPATH. We construct an instance of TEXP consisting of a continuously strongly connected temporal graph $D = (V, A)$ and a source node s'. D consists of three static graphs T_1, T_2, and T_3 as illustrated in Figure 1. The first graph T_1 (Figure 1(a)) consists of $G_1 = G$ and a set V_2 of additional nodes, i.e. $V = V_1 \cup V_2$. Denote by n, n_1, and n_2 the cardinalities of V, V_1, and V_2, respectively. For the time being it suffices to assume that $n_2 > n_1$. We set $s' = s$. We connect every node of V_1 to the leftmost node of V_2, then continue with a directed path spanning V_2 (i.e. a hamiltonian path on V_2), and finally we connect the rightmost node of V_2 to each node of V_1. T_1 persists until time $n_1 - 1$, that is $D(t) = T_1$ for all $1 \leq t \leq n_1 - 1$. Then, at time n_1, D changes to the second graph T_2 (Figure 1(b)) which is the same as T_1 without the internal edges of set V_1 (those are the edges of G that were present in T_1). T_2 persists until time $n_2 - 1$, that is $D(t) = T_2$ for all $n_1 \leq t \leq n_2 - 1$. Finally, at time n_2, D changes to the third graph T_3 (Figure 1(c)) in which each of V_1 and V_2 has its nodes connected by a line of 2-cycles and the left endpoints of the two sets are also connected by a 2-cycle. T_3 is preserved up to the lifetime of D, that is $D(t) = T_3$ for all $n_2 \leq t \leq \alpha(D)$. To ensure explorability of D, it suffices to set $\alpha(D) = 2n_2 + n_1$. Note that D is a continuously strongly connected temporal graph because T_1, T_2, and T_3 are strongly connected graphs.

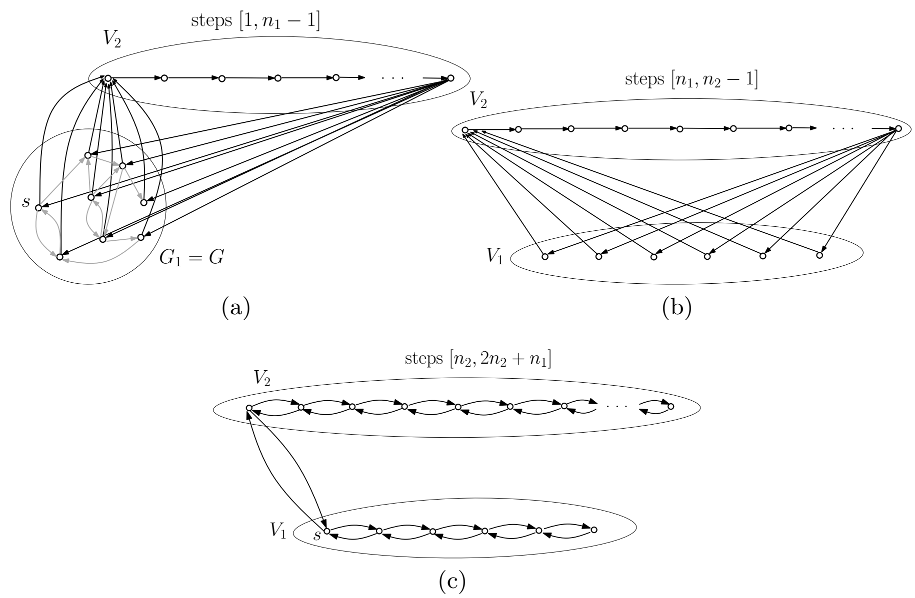

Fig. 1. The temporal graph constructed by the reduction. (a) T_1 (b) T_2 (c) T_3

($\Rightarrow$) If G is hamiltonian, then the hamiltonian path of G_1, followed by an edge leading from V_1 to V_2, and finally followed by the hamiltonian path on V_2 gives a hamiltonian journey of D and thus V can be explored optimally in $n_1 + n_2 - 1$ steps.

($\Leftarrow$) If G is not hamiltonian, then we prove that in this case the optimum exploration needs at least $2n_2 + 1$ steps. Observe that by time $n_1 - 1$ the exploration cannot have visited all nodes of V_1 because G_1 is not hamiltonian from s (Figure 1(a)). This remains true until time $n_2 - 1$, because in the interval $[n_1, n_2 - 1]$ the only edges that lead to nodes in V_1 cannot have been reached before time n_2 (Figure 1(b)). So, by time $n_2 - 1$ there is an unvisited node in V_1. Moreover, by the same time the rightmost node of V_2 is also unvisited because the temporal distance from $(s, 0)$ to it is n_2. Then, even if at time n_2 the exploration hits one of them, the other is at distance $\geq n_2 + 1$ because the leftmost node of V_1 in Figure 1(c) is s. So, in total, at least $2n_2 + 1$ steps are needed to explore V.

It remains to prove that the above reduction can be adjusted to introduce the claimed gap. As ε is a constant, we can restrict attention to instances of HAMPATH of order at least $2/\varepsilon$ and provide a gap introducing reduction from those instances (which obviously still remain hard to decide), that is $n_1 \geq 2/\varepsilon \Rightarrow \varepsilon \geq 2/n_1 \Rightarrow 2-\varepsilon \leq 2-(2/n_1)$. Moreover, in the above reduction set $n_2 = n_1^2+n_1$ (observe that we can set n_2 equal to any polynomial-time computable function of n_1). So, by what has been proved so far, we have that:

- If G is hamiltonian, then $\text{OPT} = n_1 + n_2 - 1 = n_1^2 + 2n_1 - 1$.
- If G is not hamiltonian, then $\text{OPT} \geq 2n_2 + 1 = 2(n_1^2 + n_1) + 1 > 2(n_1^2 + n_1)$.

Consider the hamiltonian case. As $2 - \varepsilon \leq 2 - (2/n_1)$ we have

$$(2-\varepsilon)(n_1^2 + 2n_1 - 1) \leq (2 - \frac{2}{n_1})(n_1^2 + 2n_1 - 1) = 2n_1^2 + 4n_1 - 2 - 2n_1 - 4 + \frac{2}{n_1}$$
$$= 2(n_1^2 + n_1) + (\frac{2}{n_1} - 6) \leq 2(n_1^2 + n_1).$$

Thus, whenever G is hamiltonian, the $(2-\varepsilon)$-approximation algorithm returns a solution of cost $\leq (2-\varepsilon)\text{OPT} = (2-\varepsilon)(n_1^2 + 2n_1 - 1) \leq 2(n_1^2 + n_1)$. On the other hand, whenever G is not hamiltonian, $\text{OPT} > 2(n_1^2 + n_1)$ and thus also the solution returned by the algorithm must have cost $> 2(n_1^2 + n_1)$. Thus a $(2-\varepsilon)$-approximation algorithm would decide instances of HAMPATH of order at least $2/\varepsilon$ in polynomial time, by comparing the solution to the polynomial-time computable $2(n_1^2 + n_1)$ threshold. This cannot be the case unless $\mathbf{P} = \mathbf{NP}$. □

On the positive side:

Theorem 3. *We provide a d-approximation algorithm for TEXP restricted to temporal graphs with dynamic diameter* $\leq d$ *and lifetime* $\geq (n-1)d$.

4 Temporal Traveling Salesman with Costs One and Two

In this section, we deal with TTSP(1,2) which is a generalization of the well known ATSP(1,2) to weighted temporal graphs. Recall that in TTSP(1,2) we are given a complete temporal graph $D = (V, A)$, with its time-edges weighted according to a cost function $c : A \to \{1, 2\}$, and we are asked to find a temporal TSP tour of minimum total cost. Our approach is to compute a temporal matching using many 1s and then extend it to a TTSP tour. Unfortunately:

Theorem 4. *Max-TEM*($\geq k$) *is* **NP***-hard for every independent of the lifetime polynomial-time computable* $k \geq 1$.

4.1 Approximating TTSP(1,2) via Maximum Independent Sets

Clearly, by taking an arbitrary temporal TSP tour, one obtains a trivial 2-factor approximation for TTSP(1,2). In the worst case, its cost is $2n$ (paying always

2s) while the cost of the optimum TSP tour is at least n (paying always 1s). Can we do better? Recall that, in ATSP(1,2) it is known that we can do much better as there is a (5/4)-factor approximation [Blä04]. In this section, we provide our first approximation algorithms for both the generic TTSP(1,2) and its special case with lifetime restricted to n. To do this, we first show that, though by Theorem 4 Max-TEM(≥ 2) is **NP**-hard, it can still be approximated within some constant via a reduction to Maximum Independent Set (MIS) in 5-claw free graphs. Recall that a graph is k-claw free if there is no k-independent set in the neighborhood of any node. We then translate this to an approximation for TTSP(1,2). For the restricted lifetime case we follow in Section 4.1.1 a similar approach by approximating a temporal path packing this time.

We begin by showing that a constant factor approximation algorithm for Max-TEM(≥ 2) translates to a constant approximation algorithm for TTSP(1,2) with factor strictly smaller than 2. This then naturally motivates us to search for constant approximations for temporal matchings.

Lemma 1. *An $(1/c)$-factor approximation for Max-TEM(≥ 2) implies a $(2 - \frac{1}{2c})$-factor approximation for TTSP(1,2).*

We now present a constant factor approximation for Max-TEM(≥ 1).

Theorem 5. *There is a $(3/5)$-approximation algorithm for Max-TEM(≥ 1).*

Proof. We are given a temporal graph $D = (V, A)$ and our goal is to return a temporal matching M of maximum cardinality. To simplify the description let us consider the static expansion $H = (S, E)$ of D. Now given an edge $e = (u_{(i-1)j}, u_{ij'})$ of the static expansion we may think of it as having the following positions for conflicts with other edges, i.e. edges that cannot be taken together with e in a temporal matching: (1) Edges of the same row as e, i.e. all edges of the form $(u_{(i-1)l}, u_{il'})$, (2) edges of the same column as $u_{(i-1)j}$, i.e. all edges that have one endpoint of the form u_{kj}, and (3) edges of the same column as $u_{ij'}$, i.e. all edges that have one endpoint of the form $u_{kj'}$. Consider now the graph $G = (E, K)$ where $(e_1, e_2) \in K$ iff e_1 and e_2 satisfy some of the above three constraints. Observe that the set of nodes E of G is the set of edges of the static expansion H. It is straightforward to observe that temporal matchings of D are now equivalent to independent sets of G. Observe now that G is 4-claw free which means that there is no 4-independent set in the neighborhood of any node. To see this take any $e \in E$ and any set $\{e_1, e_2, e_3, e_4\}$ of four neighbors of e in G. As there are only three constraints at least two of the neighbors, say e_i and e_j, must be connected to e by the same constraint. Finally, observe that if e_i and e_j both satisfy the same constraint with e (e.g. belong to the same row as e) then they must satisfy the same constraint with each other (e.g. if e_i and e_j belong to the same row as e then e_i belongs to the same row as e_j) implying that e_i and e_j are also connected by an edge in G. From [Hal95] we have a factor of 3/5 for MIS in 4-claw free graphs. □

The following lemma makes a slight modification to the proof of Theorem 5 to obtain a constant approximation for Max-TEM(≥ 2).

Lemma 2. *There is a* $\frac{1}{2+\varepsilon}$*-approximation algorithm for Max-TEM(*≥ 2*).*

Theorem 6. *There is a* $(7/4+\varepsilon)$*-approximation algorithm for TTSP*$(1,2)$*.*

4.1.1 Lifetime Restricted to n We now restrict our attention to temporal graphs with lifetime $\alpha(D)$ restricted to n. In this case, we show that an extension of the above ideas provides us with an improved $12/7 \approx 1.71$-factor approximation for TTSP$(1,2)$. A difference now is that instead of approximating a temporal matching we approximate a temporal path packing.

Lemma 3. *An* $(1/c)$*-factor approximation for TPP implies a* $(2-\frac{1}{c})$*-factor approximation for TTSP*$(1,2)$*.*

Lemma 4. *There is a* $\frac{1}{(7/2)+\varepsilon}$*-factor approximation for TPP when* $\alpha(D)=n$*.*

Proof. We directly express a TPP as an independent set of time-edges in the static expansion $H=(S,E)$. Given an edge $e=(u_{ij},u_{(i+1)j'}) \in E$ we add the following constraints. (1) All edges with tail u_{ik} (i.e. for all $1 \leq k \leq n$), (2) all edges $(u_{(i-1)k},u_{il})$ such that $l \neq j$ or $k=j'$, (3) all edges $(u_{(i+1)k},u_{(i+2)l})$ such that $k \neq j'$ or $l=j$, (4) all edges (with tails) in $[1,i-2] \cup [i+2,n]$ that have an endpoint in the same column as the tail of e, and (5) all edges (with tails) in $[1,i-2] \cup [i+2,n]$ that have an endpoint in the same column as the head of e. Note now that the resulting graph of constraints is $(7+1)$-claw free. From [Hal95], in $(h+1)$-claw free graphs, for all $h \geq 4$, MIS can be approximated within $1/(h/2+\varepsilon)$. As in our case $h=7$ we have a $[1/(7/2+\varepsilon)]$-factor approximation for MIS and thus for TPP. ☐

Theorem 7. *There is a* $(12/7+\varepsilon)$*-factor approximation algorithm for TTSP*$(1,2)$ *when* $\alpha(D)=n$*.*

4.2 Improved Approximations for TTSP(1,2) via Set Packing

We now present a different reduction idea, from Max-TEM(≥ 2) to k-Set Packing this time, that gives improved approximations for TTSP(1,2).

Lemma 5. *There is a* $\frac{3}{5+\varepsilon}$*-approximation algorithm for Max-TEM(*≥ 2*).*

Proof. We express the temporal matching problem as a 4-Set Packing. Then, from [Cyg13], we have that k-Set Packing can be approximated within $3/(k+1+\varepsilon)$ yielding $3/(5+\varepsilon)$ for $k=4$. In k-Set Packing we are given a family $F \subseteq 2^U$ of sets of size at most k, where U is some universe of elements, and we are asked to find a maximum size subfamily of F of pairwise disjoint sets. Given $D=(V,A)$, we set $U=V \cup \{1,2,\ldots,\alpha(D)\}$. Let $H=(S,E)$ be the static expansion of D. Construct now F as follows. For every $(u_{ij},u_{(i+1)j'}) \in E$ set $F \leftarrow F \cup \{\{u_j,u_{j'},i-1,i\}\}$. Clearly, $\{u_j,u_{j'},(i-1),i\} \in 2^U$ because u_j, $u_{j'}$, $i-1$, and i are pairwise distinct elements, thus indeed $F \subseteq 2^U$. Note that every set contains 4 elements, thus we have created an instance of 4-Set Packing. The claim follows by observing that there is a temporal matching of size h in D iff there is a packing of F of size h. ☐

Theorem 8. *There is a* $(1.7+\varepsilon)$*-approximation algorithm for TTSP*$(1,2)$*.*

4.2.1 Lifetime Restricted to n Now assume again that the lifetime $\alpha(D)$ of the temporal graph is restricted to n. In this case, we devise via a reduction to 3-Set Packing an improved $13/8 = 1.625$-factor approximation for TTSP$(1,2)$.

Theorem 9. *There is a $(13/8 + \varepsilon)$-factor approximation algorithm for TTSP$(1,2)$ when $\alpha(D) = n$.*

Proof. Every TTSP tour, including the optimum tour, must necessarily use precisely the time-labels $1, 2, \ldots, n$ because otherwise it cannot cover all nodes in n steps. So, the optimum TTSP tour can be partitioned into two temporal matchings, M_O and M_E, both with time differences ≥ 2 between consecutive labels. M_O is the *odd matching* using labels $1, 3, 5, \ldots$ and M_E is the *even matching* using labels $2, 4, 6, \ldots$. So, if we denote by OPT_{TTSP} the cost of the optimum TTSP tour and by $o(D')$ the number of edges of cost one of a single-label subgraph D' of the temporal graph D, we have $o(M_O) + o(M_E) = 2n - \text{OPT}_{TTSP}$.

We now approximate the maximum odd and maximum even matchings of the temporal graph D (counting the number of edges of cost one). Assume, for example, that we want to approximate the maximum matching that uses only odd labels (the even labels case is symmetric). We express it as a 3-Set Packing as follows. Recall that in 3-Set Packing we are given a family $F \subseteq 2^U$ of sets of size at most 3, where U is some universe of elements, and we are asked to find a maximum size subfamily of F of pairwise disjoint sets. We set $U = V \cup L_O$, where $L_O = \{1, 3, 5, \ldots\} \subset \{1, 2, \ldots, n\}$ is the set of all odd labels. Now consider the subgraph $H = (S, E)$ of the static expansion of D consisting only of the edges of cost one that appear at odd times and construct F as follows. For every $(u_{ij}, u_{(i+1)j'}) \in E$ set $F \leftarrow F \cup \{\{u_j, u_{j'}, i\}\}$. Clearly, $\{u_j, u_{j'}, i\} \in 2^U$ because u_j, $u_{j'}$, and i are pairwise distinct elements, thus indeed the constructed $F \subseteq 2^U$. Note that every set contains 3 elements, thus we have created an instance of 3-Set Packing. It is not hard to show that there is an odd temporal matching of size h iff there is a packing of size h. The reason is that two sets $\{u, v, t\}$ and $\{u', v', t'\}$ do not conflict and can be picked together in the packing iff the corresponding edges can be picked at the same time in an odd temporal matching. Now, from [Cyg13], we have that k-Set Packing can be approximated within $3/(k + 1 + \varepsilon)$ yielding $3/(4 + \varepsilon')$ for $k = 3$. We omit ε' in the sequel and add it in the end. So, if we denote by OPT_O and ALG_O (OPT_E and ALG_E) the size of the optimum odd (even) matching and of the odd (even) matching produced by the above algorithm, respectively, we have $\text{ALG}_O \geq \frac{3}{4}\text{OPT}_O$ and $\text{ALG}_E \geq \frac{3}{4}\text{OPT}_E$. Now from the two computed matchings we keep the one with maximum cardinality. Denote its cardinality by ALG_M. Clearly, $2\text{ALG}_M \geq \text{ALG}_O + \text{ALG}_E$, so we have

$$\begin{aligned}\text{ALG}_M &\geq \frac{1}{2}(\text{ALG}_O + \text{ALG}_E) \geq \frac{1}{2} \cdot \frac{3}{4}(\text{OPT}_O + \text{OPT}_E) = \frac{3}{8}(\text{OPT}_O + \text{OPT}_E)\\ &\geq \frac{3}{8}[o(M_O) + o(M_E)] = \frac{3}{8}(2n - \text{OPT}_{TTSP}) = \frac{6}{8}n - \frac{3}{8}\text{OPT}_{TTSP}\end{aligned}$$

Now, we complete the produced matching arbitrarily with the missing edges to obtain a TTSP tour. This is feasible because the matching has time differences

≥ 2 between its edges. Denote by ALG_{TTSP} the cost of the produced TTSP tour. As in the worst case, every added edge has cost 2, we have

$$\text{ALG}_{TTSP} \leq 2n - \text{ALG}_M \leq 2n - \frac{6}{8}n + \frac{3}{8}\text{OPT}_{TTSP} = \frac{10}{8}n + \frac{3}{8}\text{OPT}_{TTSP}$$
$$\leq \frac{10}{8}\text{OPT}_{TTSP} + \frac{3}{8}\text{OPT}_{TTSP} = \frac{13}{8}\text{OPT}_{TTSP}. \qquad \square$$

References

[AGM+10] A. Asadpour, M. X. Goemans, A. Madry, S. O. Gharan, and A. Saberi. An $O(\log n/\log\log n)$-approximation algorithm for the asymmetric traveling salesman problem. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 379–389, 2010.

[Ber96] K. A. Berman. Vulnerability of scheduled networks and a generalization of Menger's theorem. *Networks*, 28[3]:125–134, 1996.

[Blä04] M. Bläser. A 3/4-approximation algorithm for maximum atsp with weights zero and one. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 61–71. Springer, 2004.

[Cyg13] M. Cygan. Improved approximation for 3-dimensional matching via bounded pathwidth local search. In *Proceedings of the IEEE 54th Annual Symposium on Foundations of Computer Science (FOCS)*, 2013.

[DPR+13] C. Dutta, G. Pandurangan, R. Rajaraman, Z. Sun, and E. Viola. On the complexity of information spreading in dynamic networks. In *Proc. of the 24th Annual ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 717–736, 2013.

[GSS11] S. O. Gharan, A. Saberi, and M. Singh. A randomized rounding approach to the traveling salesman problem. In *Proceedings of the IEEE 52nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 550–559, Washington, DC, USA, 2011. IEEE Computer Society.

[Hal95] M. M. Halldórsson. Approximating discrete collections via local improvements. In *Proceedings of the sixth annual ACM-SIAM symposium on Discrete algorithms (SODA)*, pages 160–169, 1995.

[KKK00] D. Kempe, J. Kleinberg, and A. Kumar. Connectivity and inference problems for temporal networks. In *Proceedings of the 32nd annual ACM symposium on Theory of computing (STOC)*, pages 504–513, 2000.

[KS13] M. Karpinski and R. Schmied. On improved inapproximability results for the shortest superstring and related problems. *Proc. 19th CATS*, pages 27–36, 2013.

[MMCS13] G. B. Mertzios, O. Michail, I. Chatzigiannakis, and P. G. Spirakis. Temporal network optimization subject to connectivity constraints. In *40th International Colloquium on Automata, Languages and Programming (ICALP)*, volume 7966 of *LNCS*, pages 663–674. Springer, 2013.

[Orl81] J. B. Orlin. The complexity of dynamic languages and dynamic optimization problems. In *Proceedings of the 13th annual ACM symposium on Theory of computing (STOC)*, pages 218–227. ACM, 1981.

[PY93] C. H. Papadimitriou and M. Yannakakis. The traveling salesman problem with distances one and two. *Mathematics of Operations Research*, 18[1]:1–11, 1993.