ORIGINAL ARTICLE

Dense subgraphs in temporal social networks

Riccardo Dondi[1] · Pietro Hiram Guzzi[2,3] · Mohammad Mehdi Hosseinzadeh[1] · Marianna Milano[2,3]

Abstract

Interactions among entities are usually modeled using graphs. In many real scenarios, these relations may change over time, and different kinds exist among entities that need to be integrated. We introduce a new network model called temporal dual network, to deal with interactions which change over time and to integrate information coming from two different networks. In this new model, we consider a fundamental problem in graph mining, that is, finding the densest subgraphs. To deal with this problem, we propose an approach that, given two temporal graphs, (1) produces a dual temporal graph via alignment and (2) asks for identifying the densest subgraphs in this resulting graph. For this latter problem, we present a polynomial-time dynamic programming algorithm and a faster heuristic based on constraining the dynamic programming to consider only bounded temporal graphs and a local search procedure. We show that our method can output solutions not far from the optimal ones, even for temporal graphs having 10000 vertices and 10000 timestamps. Finally, we present a case study on a real dual temporal network.

Keywords Complex networks · Temporal graphs · Dual graphs · Graph algorithms · Dense subgraph

1 Introduction

Novel network models have been introduced, extending the classic graph model to represent properties of complex systems. For example, temporal information about interactions is represented in temporal graphs (Kempe et al. 2002; Holme and Saramäki 2012; Wu et al. 2014; Kostakis et al. 2017; Akrida et al. 2020; Dondi and Hosseinzadeh 2021; Rozenshtein and Gionis 2019; Galicia et al. 2020; Hosseinzadeh

✉ Riccardo Dondi
  riccardo.dondi@unibg.it

  Pietro Hiram Guzzi
  hguzzi@unicz.it

  Mohammad Mehdi Hosseinzadeh
  m.hosseinzadeh@unibg.it

  Marianna Milano
  m.milano@unicz.it

1   Università degli Studi di Bergamo, Bergamo, Italy

2   Magna Graecia University, Catanzaro, Italy

3   Data Analytics Research Centre, University of Catanzaro, Catanzaro, Italy

et al. 2023), while integration of different kinds of relationships is considered in dual graphs (Wu et al. 2016; Chen et al. 2022; Dondi et al. 2021) and network of networks (Gu et al. 2022).

In this paper, we introduce a new network model, called temporal dual network, to integrate interactions from two different networks (as in dual networks) that change over time (as in temporal graphs). The new model can be helpful to analyze the evolution of networks, in particular their cohesive parts. The integration of two static graphs via dual networks has been successfully applied in different domains (Wu et al. 2016). Examples of dual networks application range from genetics (protein–protein interaction - physical network - and interaction between two genetic variants - conceptual network) to social networks (co-author network - physical - and interest similarity - conceptual network) and recommender systems (social connectivity - physical network - and rating similarity - conceptual network). Our approach can be applied in these domains when it is interesting to consider also the temporal evolution of a network.

In this paper, we consider the case where we want to analyze a community an author belongs to. The idea is to consider two networks, a co-authorship network (conceptual network) and a network based on research interest (physical network). The community an author belongs to may

vary, as she/he may have new coauthors or may strengthen the relations with an existing author (by publishing more papers, for example) or again, a relation may be weakened over time. On the other hand, an author may change her/his research interests over time. For these reasons, considering only static graphs is not enough to represent these dynamics, but we have to consider how networks/communities change over time. Here, we consider two temporal networks (a conceptual and a physical temporal network) that represents different information, for example, a conceptual, temporal network represents a co-authorship temporal network; a physical network research interest.

We present a case study focusing on researchers working on algorithms for bioinformatics between 1993 and 2002, when this community started to establish (data are extracted from DBLP). A physical temporal graph is built based on the participation of two authors at the same conference in a specific year. This graph represents the relation between researchers who share interests (they both attend a same conference) but are not necessarily co-authors. The conceptual graph is a co-author temporal graph, which is built considering the mutual publications.

Another example of an application is the analysis of social networks to understand the preferences of users, as it may change some interests over time and this may be inferred from the context she/he considers on a platform and from new relations she/he establishes.

In this paper, we consider the identification of dense subgraphs in the context of temporal dual networks. The identification of cohesive subgraphs is a fundamental problem in graph mining since it is related to the identification of cohesive groups (Chen and Saad 2010; Galbrun et al. 2016; Dondi et al. 2021; Hosseinzadeh 2020; Cinaglia and Cannataro 2022). An analysis of the evolution of motifs in temporal networks has been proposed in Braha and Bar-Yam (2009) and the identification of dense subgraphs has been recently considered for temporal networks (Rozenshtein and Gionis 2019; Dondi and Hosseinzadeh 2021; Castelli et al. 2020).

Figure 1 depicts a toy example of a social network in the co-authorship domain, which may be represented as a temporal dual network. The figure shows three different timestamps of a temporal dual network. For each timestamp, both the physical and conceptual networks are depicted. Each node of a physical network represents an author, and the edges are the *co-authorship* relation. Weighted edges of the conceptual network model the shared research interests. The algorithm we present can detect the subgraph induced by Adam, Zhang, and Wang nodes.

We propose a problem for the identification of k densest subgraphs that are temporally disjoint in a temporal dual network, and we design a heuristic for it. This method is

based on (1) computing an alignment graph of the conceptual and physical graph and (2) finding k densest subgraphs in the alignment graph. For this second step, we design two algorithms: an exact dynamic programming algorithm, which is applicable only for small datasets, and a heuristic. This heuristic is based on solving a constrained version of the problem via dynamic programming and then applying a local search procedure. We present an experimental evaluation of these algorithms on synthetic datasets, generated varying the number of timestamps from 70 to 10000 and the number of nodes from 70 to 10000. Moreover, we present a case study on a real dual temporal network built by extracting data from DBLP.

The paper is organized as follows. First, in Sect. 2, we give the definitions that will be useful. We present the dual temporal graph model in the remaining part of the paper. Then, in Sect. 3, we present the algorithmic contributions of the paper, while in Sect. 4, we will present an experimental evaluation of our heuristic on synthetic datasets and a real dual temporal network. Finally, we conclude the paper with some future directions.

2 Definitions

In this section, we start by giving the definitions of temporal graphs and dual graphs, and we introduce the temporal dual graph model. Then, we present the formal definition of the problem we are interested in, that is finding k densest subgraphs that are active in disjoint intervals.

We start by introducing a discrete time domain over which is defined a temporal graph and a temporal dual graph.

Definition 1 A discrete time domain $\mathcal{T} = [0, 1, \ldots, t_{max}] \subseteq \mathbb{N}$, is a sequence of timestamps $t \in \mathcal{T}$. An interval $T = [t_i, t_j]$ over $\mathcal{T}$, with $t_i, t_j \in \mathcal{T}$ and $t_i < t_j$, consists of the timestamps between t_i and t_j.

Two intervals are *disjoint* if they do not share any timestamp. Next, we can present the definition of temporal graph. Notice that in the model we consider the set of nodes is not changing over time.

Definition 2 $G = (V, \mathcal{T}, E)$ is a temporal graph, where V is a set of nodes, and $E \subseteq V \times V \times \mathcal{T}$ is a set of temporal edges.

Given a temporal graph $G = (V, \mathcal{T}, E)$ and a temporal interval T, we define $G[T] = (V, E[T])$ as the *active graph* of G in interval T, where $E[T]$ is the set of active edges at interval T, defined as follows:

$$E[T] = \{(u, v, t) | (u, v, t) \in E \land t \in T\}.$$

Fig. 1 A toy example that illustrates the application of the proposed method
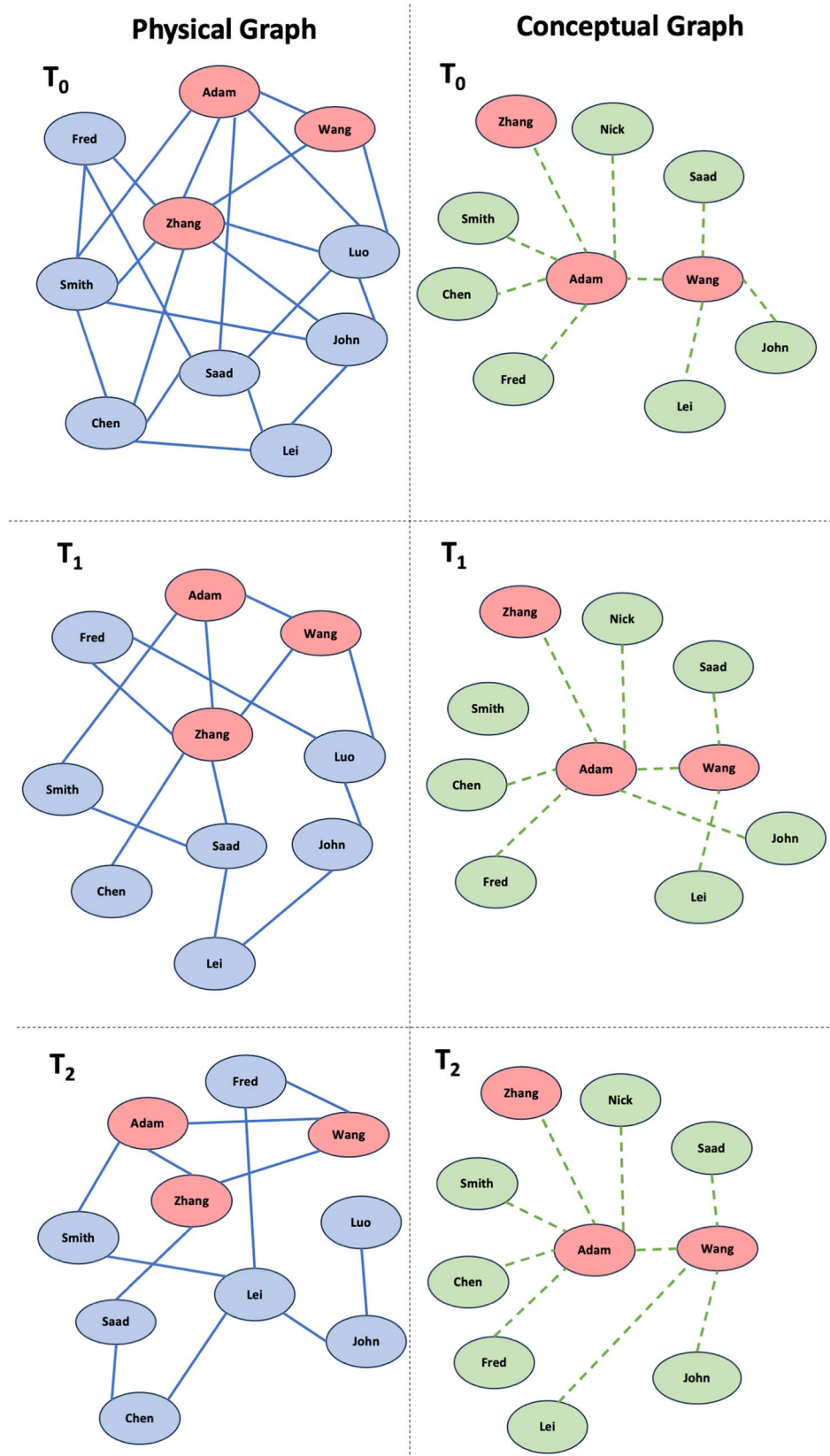
A similar definition of active edges can be given for active edges at timestamp $t \in \mathcal{T}$:

$$E[t] = \{(u, v, t)|(u, v, t') \in E \wedge t' = t\}.$$

We can now define the concept of episodes, which represent the temporal subgraphs we will look for.

Definition 3 Let $G = (V, \mathcal{T}, E)$ be a temporal graph, an episode, denoted by $G[W, T]$, where $W \subseteq V$ and T is an interval over $\mathcal{T}$, is a subgraph of $G[T]$ having node set W and temporal edge set E_W, where $E_W \subseteq E[T] \cap (W \times W)$.

Given a weighted temporal graph $G = (V, \mathcal{T}, E)$, an interval I over $\mathcal{T}$ and an edge $(u, v) \in E$, then the *average weight* of (u, v) in E, denoted by $w_I(u, v)$, is defined as follows:

$$w_I(u, v) = \frac{\sum_{t \in I} w(u, v, t)}{\sqrt{|I|}}$$

where $w(u, v, t)$ is the weight of edge (u, v) at time t ($w(u, v, t) = 0$ if (u, v, t) is not defined). In the definition of average weight, we divide by $\sqrt{|I|}$ and not by $|I|$, since in the latter case this may lead to dense subgraphs defined in a single timestamp.[1]

The weighted density of G in a interval I, denoted by $w\text{-dens}(G, I)$, is defined as follows:

$$w\text{-dens}(G, I) = \frac{\sum_{(u,v) \in E} w_I(u, v)}{|V|}.$$

Notice that the fact that the temporal graph is weighted changes some of the properties of episodes with respect to unweighted graphs. For example, while, as discussed in Rozenshtein and Gionis (2019), the density of episodes in unweighted graphs is a monotone non-decreasing function, this property does not hold in the weighted case, as it can be seen in the example of Fig. 2, where $w_{[1,2]}(v_1, v_2) = \frac{2}{2\sqrt{2}}$, which is approximately equal to 0.707, while $w_{[1,3]}(v_1, v_2) = \frac{2.1}{2\sqrt{3}}$, which is approximately equal to 0.404. Notice that $w_{[1,1]}(v_1, v_2) = w_{[2,2]}(v_1, v_2) = \frac{1}{2} = 0.5 \leq w_{[1,2]}(v_1, v_2)$.

Now, we introduce the definition of dual graph (an example is given in Fig. 3).

Definition 4 $G = (V, E_c, E_p, w_c)$ is a dual graph, where V is a set of nodes, and $G_c = (V, E_c, w_c)$, $G_p = (V, E_p)$ are two graphs defined over the same set of nodes V such that:

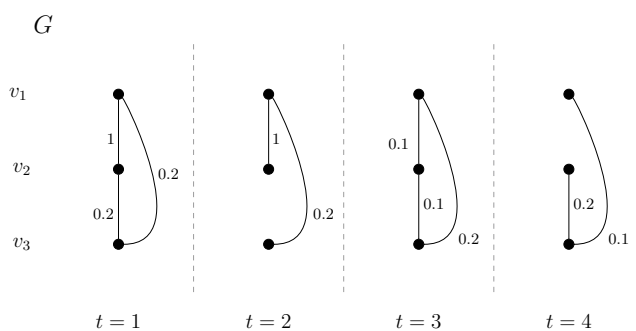– $G_c = (V, E_c, w_c)$ is a weighted graph, called conceptual graph

Fig. 2 A temporal weighted graph with three nodes and four timestamps [1, 2, 3, 4]. A densest subgraph is induced by nodes v_1 and v_2 in interval [1.2]; it has a density of $\frac{2}{2\sqrt{2}}$ which is approximately 0.707
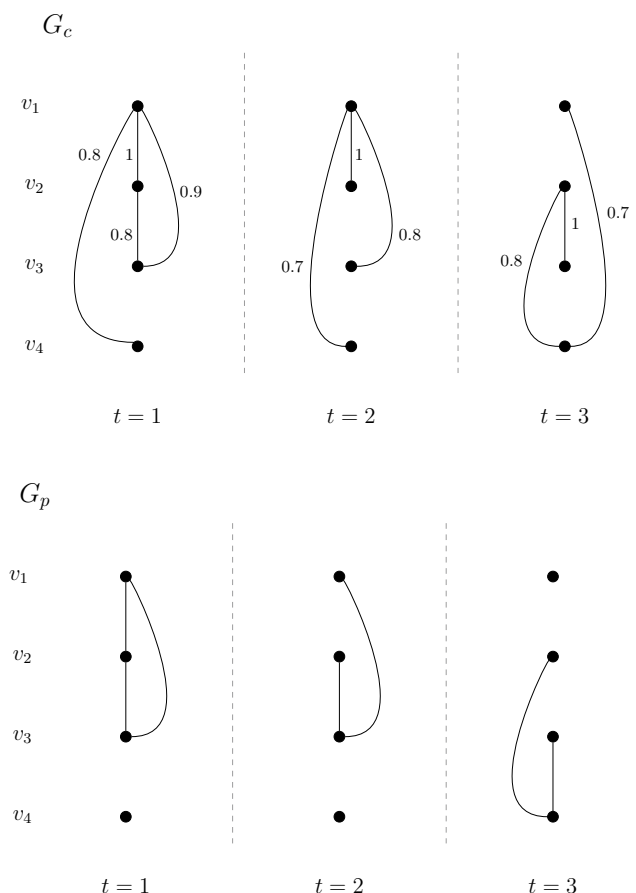

Fig. 3 An example of dual temporal graph: a conceptual temporal graph G_c (in the upper part) and a temporal physical graph G_p (in the lower part). The two graphs are defined over four vertices and three timestamps. Notice that G_c is a weighted graph (the label of each temporal edge denotes its weight), while G_p in unweighted

– $G_p = (V, E_p)$ is an unweighted graph, called physical graph.

---

[1] Here we use $\sqrt{|I|}$ but other sublinear functions can be considered as well.

Now, we are able to introduce the definition of temporal dual graph.

Definition 5 $G = (V, \mathcal{T}, E_c, E_p, w_c)$ is a temporal dual graph (TDG), where

- V is a set of nodes
- $G_c = (V, \mathcal{T}, E_c, w_c)$ is a weighted temporal graph, called conceptual temporal graph
- $G_p = (V, \mathcal{T}, E_p)$ is an unweighted temporal graph, called physical temporal graph.

Now, we are able to define a temporal densest common subgraph of a temporal dual graph, which is fundamental for the problem we are interested in.

Definition 6 Temporal Common Subgraph.

Given a temporal dual graph $G = (V, \mathcal{T}, E_c, E_p, w_c)$ associated with a conceptual temporal graph G_c and a physical temporal graph G_p, a temporal common subgraph of G is a pair (W, T) where $T \in \mathcal{T}$ is a temporal interval and $W \subseteq V$ such that:

- $G_p[W, T]$ is connected
- The weighted density of (W, T), denoted by w-dens(W, T), is equal to $dens(G_c[W, T])$ (that is the density in the conceptual temporal graph).

We define the first problem we are interested in.

Problem 1 k-Densest-Episodes in a Temporal Dual Graph

Input: A temporal dual graph $G = (V, \mathcal{T}, E_c, E_p, w_c)$, a positive integer $k \in \mathbb{N}$.

Output: A set S of k temporal common subgraphs $S = \{(I_j, W_j) : 1 \leq j \leq k\}$, where $\{I_j : 1 \leq j \leq k\}$ is a set of disjoint intervals, such that $\sum_{j=1}^{k} w\text{-dens}(W_j, I_j)$ is maximized.

The k-Densest-Episodes problem is NP-hard, since, given a static dual graph (hence a temporal graph with a time domain consisting of a single timestamp), it is NP-hard to find a densest common subgraph in it Wu et al. (2016).

In order to solve the problem, we consider the following alignment approach:

1. We first align the conceptual temporal graph and the physical temporal graph and we obtain a temporal alignment graph
2. Then, we find a set of k episodes in the temporal alignment graph

2.1 Graph alignment approach

In this Section, we describe the approach we propose to solve the k-Densest-Episodes problem on temporal dual graphs by means of adapting a graph alignment approach. The use of graph alignment to deal with dual graphs has been considered previously in the literature (Guzzi et al. 2021, 2020; Milano et al. 2020; Guzzi and Milenković 2018). Here, we extend the definition to deal with temporal dual graphs and we define an alignment for each timestamp t.

Definition 7 Starting from two input graphs, a weighted graph $G_a = (V_a, E_a)$ (where E_a is a set of weighted edges) and an unweighted graph $G_b = (V_b, E_b)$, (where E_b is a set of unweighted edges), a graph alignment of G_a and G_b is a mapping A from V_a to V_b. In our scenario, we consider, without lack of generality, that graphs have the same node set and two different edge sets.

More in depth, we consider *local alignment* which is defined as a partial injective mapping A from V_a to V_b. In our case, the mapping (hence the alignment) of two graphs is implicitly defined by their identifiers, that is two corresponding vertices in the networks have the same identifier both in V_a and in V_b. The output of the alignment is a so-called *alignment graph*, which is a weighted graph $G_{al} = (V_{al}, E_{al})$, defined as follows.

Definition 8 Given a weighted graph $G_a = (V_a, E_a, w_a)$ and an unweighted graph $G_b = (V_b, E_b)$, an alignment graph $G_{al} = (V_{al}, E_{al}, w_{al})$, between G_a and G_b is defined as follows:

- The vertex set $V_{al} = \{c_i : (v_{ai}, v_{bi}) \in A\}$
- The edge set E_{al} is defined based on two possible cases, *match and mismatch*, and depends on a parameter δ. We set $\delta = 3$ in the experiments we made. For each set $\{c_i, c_j\}$ of two vertices $c_i, c_j \in V_{al}$ corresponding to pairs (v_{ai}, v_{bi}), (v_{aj}, v_{bj}), respectively, then:

  1. If both $(v_{ai}, v_{aj}) \in E_a$, and $(v_{bi}, v_{bj}) \in E_b$, then $(c_i, c_j) \in E_{al}$ with weight $w_{al}(c_i, c_j) = w_a(v_{ai}, v_{aj})$
  2. If $(v_{ai}, v_{aj}) \in E_a$, and $(v_{bi}, v_{bj}) \notin E_b$, where v_{bi}, v_{bj} are at distance lower than δ in G_b, then $(c_i, c_j) \in E_{al}$ with weight $w_{al}(c_i, w_j)$ defined as the average weight of the edges of the path connecting v_{bi}, v_{bj} in G_b (mismatch 1 case)
  3. If $(v_{ai}, v_{aj}) \in E_a$, and $(v_{bi}, v_{bj}) \notin E_b$, where v_{bi}, v_{bj} are at distance at least δ in G_b, then $(c_i, c_j) \notin E_{al}$ (mismatch 2 case)
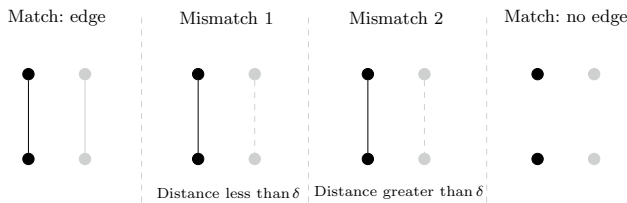 4. If $(v_{ai}, v_{aj}) \notin E_a$, then $(c_i, c_j) \notin E_{al}$.

Fig. 4 The possible cases of the graph alignment. The figure shows four pairs of edges. The two input graphs are highlighted with two different colors, black and gray. From the left we show a match and a mismatch case 1 (when the distance of the nodes in graph 2 is less than δ), a mismatch case 2 (when the distance of the nodes in graph 2 is greater than δ), and the absence of connection in the graphs

The output of the alignment is a new graph $G_{al} = (V_{al}, E_{al})$, called *alignment graph*. Figure 4 presents the possible cases, where we draw G_a with black vertices/edges and G_b with gray vertices/edges.

Definition 9 Timestamp Alignment Graph. Given a temporal dual graph $G = (V, \mathcal{T}, E_c, E_p, w_c)$, for each timestamp $t \in \mathcal{T}$, a Timestamp Alignment Graph $G_A[t]$ is an alignment graph of the conceptual graph $G_c[t]$ and the physical graph $G_p[t]$ of the same timestamp t. A temporal alignment graph $G_A = (V, \mathcal{T}, E_A, w_A)$ is a collection of timestamp alignment graphs, one for each timestamp, that is

$$G_A = \bigcup_{t=0}^{t_{max}} G_A[t]$$

2.2 Finding episodes in the alignment graphs

Once the temporal alignment graph is computed, we consider the problem of finding a set of (weighted) episodes in it, as defined in the following problem.

Problem 2 k-Densest-Alignment-Episodes

Input: A temporal alignment graph $G_A = (V, \mathcal{T}, E_A, w_A)$, a positive natural $k \in \mathbb{N}$.

Output: A set S of k temporal densest subgraphs $S = \{(I_j, W_j) : 1 \le j \le k\}$, where $\{I_j : 1 \le j \le k\}$ is a set of disjoint intervals, such that $\sum_{j=1}^{k} w\text{-dens}(W_j, I_j)$ is maximized.

We consider also a variant of the problem, called k-ℓ-Densest-Alignment-Episodes, that we introduce as an intermediate problem to design a heuristic for k-Densest-Alignment-Episodes. In k-ℓ-Densest-Alignment-Episodes the episodes are constrained to happen in a bounded length interval.

Problem 3 k-ℓ-Densest-Alignment-Episodes

Input: A temporal alignment graph $G_A = (V, \mathcal{T}, E_A, w_A)$, two positive naturals $\ell, k \in \mathbb{N}$, with $\ell \le t_{max}$.

Output: A set S of k temporal densest subgraphs $S = \{(I_j, W_j)\} : 1 \le j \le k\}$, where $\{I_j : 1 \le j \le k\}$ is a set of disjoint intervals each one of length at most ℓ, such that $\sum_{j=1}^{k} w\text{-dens}(W_j, I_j)$ is maximized.

We will show in Sect. 3 that, unlike the k-Densest-Episodes problem, k-Densest-Alignment-Episodes and k-ℓ-Densest-Alignment-Episodes can be solved in polynomial time.

2.3 The densest subgraph problem

The approach we propose for solving the k-Densest-Alignment-Episodes and the k-ℓ-Densest-Alignment-Episodes problem is based on the computation of a solution of the Densest Subgraph problem on static (weighted) graphs. Given a graph the Densest Subgraph problem asks for a subgraph of maximum weighted density. The problem can be solved in polynomial-time (Goldberg 1984) with Goldberg's algorithm, which is based on a reduction to a series of min-cut computation. The time complexity of the Goldberg's algorithm is $O(mn \log n)$ (also in $O(n^3)$ time for unweighted graphs (Kawase and Miyauchi 2018)). Furthermore, the Densest Subgraph problem can be approximated within factor $\frac{1}{2}$ by a greedy algorithm of time complexity $O(n + m)$ for unweighted graphs and $O(m + n \log n)$ for weighted graphs (Charikar 2000). In what follows, we denote by $t_{densest}$ the time required to compute a densest subgraph in a static graphs.

3 Algorithms for k-Densest-Alignment-Episodes and k-ℓ-Densest-Alignment-Episodes

In this section, we present our algorithmic methods. We start by presenting the dynamic programming polynomial-time algorithms for k-Densest-Alignment-Episodes (an algorithm called DP) and k-ℓ-Densest-Alignment-Episodes (an algorithm called L-DP), then we present a heuristic approach applied on an optimal solution of k-ℓ-Densest-Alignment-Episodes in order to compute a (possibly suboptimal) solution of k-Densest-Alignment-Episodes.

3.1 Polynomial-time algorithms for k-Densest-Alignment-Episodes and k-ℓ -Densest-Alignment-Episodes

First, we present the DP algorithm for k-Densest-Alignment-Episodes. Given an alignment graph G_A over the time interval $[1, j]$, with $j \leq t_{\max}$, we consider the function $D[j, h]$, with $h \leq k$ and $0 \leq j \leq t_{\max}$, that is equal to the density of h densest episodes in $G_A[1, j]$.

Given two timestamps i and j, with $1 \leq i \leq j \leq t_{\max}$, we denote by $\mathrm{Dens}(G_A[i, j])$ the density of a densest subgraph in $G_A[i, j]$, where the subgraph must be defined in timestamp j (not necessarily in i). Assume that $\mathrm{Dens}(G_A[i, j])$ has already been computed for all values $1 \leq i \leq j \leq t_{\max}$, function $D(j, h)$, $1 \leq i \leq j \leq t_{\max}$, $1 \leq h \leq k$, can be computed as follows:

For $h \geq 2$ and $j \geq 2$:

$$D(j, h) = \max \begin{cases} \max_{2 \leq i \leq j} D(i-1, h-1) + \mathrm{Dens}(G_A[i, j]) & \text{with } j \geq 2 \\ D(j-1, h) \end{cases}$$ (1)

For $h = 1$ and $j \geq 2$:

$$D(j, 1) = \max_{1 \leq i \leq j} \begin{cases} \mathrm{Dens}(G_A[i, j]) \\ D(j-1, 1) \end{cases}$$ (2)

For $j = 1$:

$$D(1, h) = \begin{cases} -\infty & \text{if } h \geq 2 \\ \mathrm{Dens}(G_A[1, 1]) & \text{if } h = 1 \end{cases}$$ (3)

Next, we prove the correctness of Eqs. 1, 2, and 3.

Lemma 1 $D(j, h) = q$ *if and only if there exist h episodes in $G_A[1, j]$ of overall density q.*

Proof We prove the lemma by induction on h and on j.

If $h = 1$, we prove the correctness of Eqs. 2 and 3 by induction on $j \geq 1$. In the base case, when $j = 1$, then $D(1, 1) = q$ if and only if there exists a densest subgraph in timestamp 1 having density q, thus proving the correctness of Eq. 3.

Now, we show the correctness for $j \geq 2$, assuming the correctness for $j - 1$. Consider one episode of maximum density contained in interval $[1, j]$, then either it is defined in timestamp j, hence it has density equal to an episode in $G_A[i, j]$, for some i with $1 \leq i \leq j$, or it is not defined in position j and by induction hypothesis it has density equal to $D(j - 1, 1)$.

Assume now that $D(j, 1) = q$. If $D(j, 1) = \max_{1 \leq i \leq j} \mathrm{Dens}(G_A[i, j])$, then there exists one episode defined in $[i, j]$ of density q. Assume that

$D(j, 1) = D(j - 1, 1) = q$, then by induction hypothesis it holds that there exists an episode of density q in interval $[1, j - 1]$. We can thus conclude that for $h = 1$ the lemma holds.

Assume now that the lemma is correct for $h - 1 \geq 1$, we show that it holds for h. More precisely, we prove that Eqs. 1 and 3 hold by induction on $j \geq 1$. In the base case, when $j = 1$, then clearly $D(1, h) = -\infty$ as it is not possible to define $h \geq 2$ episodes in a single timestamp, thus proving the correctness of Eq. 3.

Consider the case $j \geq 2$. Assume that there is a set $\mathcal{S}$ of h disjoint episodes defined in interval $[1, j]$, where the last episode in $\mathcal{S}$ is defined over interval $[i + 1, z]$, with $i \leq z \leq j$ and it has density q_1. If $z < j$, then it holds that $D(j, h) = D(j - 1, h)$, and by induction hypothesis $D(j - 1, h) = q$. If $z = j$, then $\mathcal{S}$ contains $h - 1$ disjoint episodes defined in interval $[1, i]$, hence by induction hypothesis $D(i, h - 1) = q - q_1$ and, since $\mathrm{Dens}(G_A[i + 1, j]) = q_1$, it follows that $D(j, h) = q$.

Assume that $D(j, h) = q$. Since $h > 1$, by the definition of the recurrence (Eq. 1) $D(j, h) = D(j - 1, h)$ or $D(j, h) = D(i, h - 1) + \mathrm{Dens}(G_A[i + 1, j])$. In the first case, by induction hypothesis on j there exists a set of h disjoint episodes of density q defined in interval $[1, j - 1]$, thus also in $[1, j]$. In the second case, there exists a value i, with $1 \leq i < j$, such that $D(i, h - 1) = q - q_1$ and $\mathrm{Dens}(G_A[i + 1, j]) = q_1$. Then, by induction hypothesis there exists a set of $h - 1$ disjoint episodes of density $q - q_1$ defined interval $[1, i]$ and an episode in $[i + 1, j]$ of density q_1. Hence, there exist h disjoint episodes in $[1, j]$ of overall density q, thus concluding the proof. □

The previous lemma leads to the following result.

Theorem 1 k-Densest-Alignment-Episodes *can be solved in $O(t_{\max}^2 \, k \, t_{\mathrm{densest}})$ time.*

Proof We prove that the recurrence described in Eq. 1, Eq. 2 and Eq. 3 can be computed in time $O(t_{\max}^2 \, k \, t_{\mathrm{densest}})$. Notice that the correctness of recurrence follows from Lemma 1 and from the fact that an optimal solution of k-Densest-Alignment-Episodes corresponds to the entry $D(t_{\max}, k)$.

The number of entries of $D(j, h)$ is $O(t_{\max} k)$. Each entry $D(j, h)$ can be computed in $O(t_{\max})$ time, once the values $\mathrm{Dens}(G_A[i, j])$ have been computed. Hence, $D(j, h)$ can be computed in $O(t_{\max}^2 \, k)$ time, once the values $\mathrm{Dens}(G_A[i, j])$ have been computed. Finally, $\mathrm{Dens}(G_A[i, j])$, for each i and j with $1 \leq i \leq j \leq t_{\max}$, can be computed in $O(t_{\max}^2 t_{\mathrm{densest}})$ time, thus concluding the proof. □

Next, we present the polynomial-time algorithm, called L-DP, for k-ℓ-Densest-Alignment-Episodes. We recall that an ℓ-constrained episode is an episode defined on an interval of length at most ℓ. Similarly to k-Densest-Alignment-Episodes, given an alignment graph G_A over the time interval $[1, j]$, with $j \leq t_{\max}$, we define the function $D_c[j, h]$, with $h \leq k$ and $1 \leq j \leq t_{\max}$, that is equal to the density of h densest ℓ-constrained episodes in $G_A[1, j]$.

Recall that, given an alignment graph $G_A[i, j]$, $Dens(G_A[i, j])$ denotes a densest subgraph in $G_A[i, j]$, that it is defined on an interval that must include timestamp j.

The recurrence to compute $D_c(j, h)$, for each $j \in \{1, 2, \ldots, t_{\max}\}$ is defined as follows:

For $h \geq 2$ and $j \geq 2$:

$$D_c(j, h) = \max \begin{cases} \max_{2 \leq i \leq j} D_c(i-1, h-1) + Dens(G_A[i, j]) , \\ with\, j - i + 1 \leq \ell \\ D_c(j-1, h) \end{cases} \quad (4)$$

For $h = 1$ and $j \geq 2$:

$$D_c(j, 1) = \max_{1 \leq i \leq j} \begin{cases} Dens(G_A[i, j]) & \text{with } j - i + 1 \leq \ell \\ D_c(j-1, 1) \end{cases} \quad (5)$$

For $j = 1$:

$$D_c(1, h) = \begin{cases} -\infty & \text{if } h \geq 2 \\ Dens(G_A[1, 1]) & \text{if } h = 1 \end{cases} \quad (6)$$

Similarly to k-Densest-Alignment-Episodes, we can prove the correctness of the recurrence described in Eqs. 4, 5 and 6.

Lemma 2 $D_c(j, h) = q$ *if and only if there exist h ℓ-constrained episodes in $G_A[1, j]$ of overall density q.*

Proof We prove the lemma by induction on h and on j.

If $h = 1$, then we prove the correctness of Eqs. 5 and 6 by induction on $j \geq 1$. In the base case, when $j = 1$, then $D_c(1, 1) = q$ if and only if there exists a densest subgraph defined in timestamp 1 of density q. Now, we consider the case $j \geq 1$ and we prove that it is correct, assuming the correctness of Eqs. 5 and 6 for $j - 1$. Consider one ℓ-constrained episodes of maximum density in $[1, j]$, then either it is defined in timestamp j, hence it has density equal to $\max_{1 \leq i \leq j} Dens(G_A[i, j])$ such that $j - i + 1 \leq \ell$, or it is not defined in position j and then by induction hypothesis has density equal to $D_c(j-1, 1)$.

Assume that $D_c(j, 1) = q$. If it is defined as $D_c(j, 1) = \max_{1 \leq i \leq j} Dens(G_A[i, j]) = q$, with $j - i + 1 \leq \ell$, then there exist one ℓ-constrained episode defined in $[i, j]$ of density q. Assume that $D_c(j, 1) = D_c(j-1, 1) = q$, then by induction hypothesis it holds that there exists an ℓ-constrained episode of density q in $G_A[1, j-1]$.

Assume that the lemma holds for $h - 1 \geq 1$, we show that it holds for h. We prove that Eqs. 4 and 6 hold by induction on $j \geq 1$. In the base case, when $j = 1$, then clearly $D_c(0, h) = -\infty$ as it is not possible to define more than one episode in a single timestamp.

Consider a set $\mathcal{S}$ of h disjoint ℓ-constrained episodes in $G_A[1, j]$, where the last episode in $\mathcal{S}$ is defined over interval $[i, z]$, with $i \leq z \leq j$ and $z - i + 1 \leq \ell$, and it has density q_1. If $z < j$, it holds that $D_c(j, h) = D_c(j-1, h)$, and by induction hypothesis it holds that $D_c(j-1, h) = q$. If $z = j$, then there exists $h - 1$ ℓ-constrained episodes of density $q - q_1$, hence, by induction hypothesis, it holds that $D_c(i, h-1) = q - q_1$ and since $Dens(G_A[i+1, j]) = q_1$, it follows that $D_c(j, h) = q$.

Assume that $D_c(j, h) = q$. Since $h > 1$, by the definition of the recurrence $D_c(j, h) = D_c(j-1, h)$ or $D_c(j, h) = \max_{2 \leq i \leq \ell} D_c(i, h-1) + Dens(G_A[i+1, j])$, where $j - i + 1 \leq \ell$. In the first case, by induction hypothesis on j there exists a set of h disjoint ℓ-constrained episodes of density q in $G_A[1, j-1]$. In the second case, it holds that $D_c(i, h-1) = q - q_1$ and $Dens(G_A[i+1, j]) = q_1$. Then, by induction hypothesis there exists a set of $h - 1$ disjoint ℓ-constrained episodes of density $q - q_1$ and an ℓ-constrained episode in $[i+1, j]$ of density q_1 (the episode is ℓ-constrained since $j - i + 1 \leq \ell$). Hence, there exist h disjoint episodes in $[1, j]$ of overall density q, thus concluding the proof. $\square$

Theorem 2 k-ℓ-Densest-Alignment-Episodes *can be solved in* $O(t_{\max} \ell k\, t_{densest})$.

Proof The correctness of the recurrence follows from Lemma 2 and from the fact that an optimal solution of k-Densest-Episodes corresponds to the entry $D_c(t_{\max}, k)$.

Equations 4, 5 and 6 can be computed in $O(t_{\max} \ell\, k\, t_{densest})$ time. Indeed, the number of entries of $D_c(j, h)$ is $O(t_{\max} k)$. Each entry $D_c(j, h)$ can be computed in $O(\ell)$ time, once the values $Dens(G_A[i, j])$ have been computed. Hence, $D_c(j, h)$ can be computed in $O(t_{\max} \ell\, k)$ time, once the values $Dens(G_A[i, j])$ have been computed.

Finally, consider the time-complexity to compute $Dens(G_A[i, j])$. Similarly to Theorem 1, $Dens(G_A[i, j])$, for each i and j with $1 \leq i \leq j \leq t_{\max}$ and $j - i \leq \ell$, can be computed in $O(t_{\max} \ell\, t_{densest})$ time, thus concluding the proof. $\square$

3.2 A Heuristic for k-Densest-Alignment-Episodes

The time complexity of the dynamic programming to solve k-Densest-Alignment-Episodes makes it non-practical even for medium size temporal graphs. Hence, we propose

a heuristic for k-Densest-Alignment-Episodes, which consists of two phases:

1. The L-DP algorithm solves k-ℓ-Densest-Alignment-Episodes, with $\ell = \log_2(t_{\max})$, hence having time complexity $O(t_{\max} \log_2 t_{\max} k\, t_{\text{densest}})$
2. We apply a local search procedure, called LocExt that, that starts from a solution returned in the first phase, and it aims at improving its density by local modifications (described later)

Next, we describe the LocExt phase. LocExt starts from a solution S of k-ℓ-Densest-Alignment-Episodes and applies a procedure to possibly improve its density. Notice that an interval I of $\mathcal{T}$ is said to be *uncovered* by a solution S if there is no subgraph of S that contains a timestamp in I. LocExt looks for an improvement of S by greedily applying the following procedure:

- It considers two temporal subgraphs (I_j, W_j) and (I_{j+1}, W_{j+1}) in S, and merge them in a temporal graph (I', W') (notice that there may be no episode defined in an interval between I_j and I_{j+1})
- It applies the dynamic programming algorithm described in Sect. 3 for k-Densest-Alignment-Episodes to an uncovered interval in $\mathcal{T}$; let (I_c, W_c) be the subgraph computed
- If it holds that $\text{dens}(I_c, W_c) + dens(I', W') > \text{dens}(I_j, W_j) + \text{dens}(I_{j+1}, W_{j+1})$, then it replaces (I_j, W_j) and (I_{j+1}, W_{j+1}) with (I_c, W_c) and (I', W').

We apply also the following post-processing procedure. If a subgraph returned by the algorithm consists of more than one connected component, then we replace the subgraph with its connected component that has largest density.

4 Experimental analysis

In this section, we provide an experimental evaluation of the heuristic on synthetic and we provide a case study on a real network.

4.1 Synthetic networks

In the first part of our experimental analysis, we describe the synthetic datasets that we have produced for analysis.

4.1.1 Datasets

The synthetic temporal graphs consist of a set of k cliques planted communities and a complementary background graph. These k communities are planted over distinct and non-overlapping intervals with each community existing on a mutually exclusive set of vertices. Additionally, the weight of each edge within the planted communities is set to a uniform value of 10. The background graph covers all the vertices that are a part of the planted communities over a discrete time domain $\mathcal{T}$ and is generated using the Erdős–Rényi model that employs parameters such as $p = 1/|V|$, $p = 3/|V|$, and $p = 5/|V|$. The edges of the background graph are uniformly defined in the timestamps of the time domain. Moreover, the weight of each edge in the background graph is randomly assigned from a uniform distribution in interval $[0, 4]$.

To perform our analysis, we generated three sets of synthetic networks: *Synthetic-small*, *Synthetic1*, and *Synthetic2*. In each of these sets, we varied the time domain, number of communities, and number of vertices/edges in both the background graph and the communities. The *Synthetic-small* dataset is generated specifically for comparison with optimal solutions. Each graph in this set includes a background graph with 70 vertices and a time domain of 70 timestamps with k equal to 4. Additionally, each community in the graph has 12 vertices. For the *Synthetic1* set, the background graph includes 1000 vertices and is defined in a time domain of 1000 timestamps. The value of k is equal to 20, and each community in the graph has 25 vertices. Finally, in the *Synthetic2* set, we create a background graph with 10000 vertices over a time domain of 10000 timestamps. In this set, k is equal to 40, and each community in the graph has 50 vertices.

By varying the parameters in these three sets of synthetic networks, we hope to gain insight into the behavior of the heuristic under different conditions.

4.1.2 Outcome

We present now the performance of the proposed heuristic in synthetic networks. The evaluation criteria used in this study include density, running time, and the quality of identified intervals and communities.

The F-measure[2] is used to evaluate the accuracy of the heuristic in finding the planted communities and intervals, and the results are reported for the two phases of the heuristic: L-DP for k-Densest-Alignment-Episodes and LocExt.

Table 1 provides information on the outcomes and running time of both the optimal dynamic programming algorithm (DP) for k-ℓ-Densest-Alignment-Episodes and our heuristic approach. Due to the time complexity of the optimal dynamic programming algorithm, the comparison is performed only on the *Synthetic-small dataset*. Table 1 shows that the proposed heuristic is able to compute

---

[2] F-measure is the fraction of recall and precision.

Table 1 Density, running time and quality of the interval and subgraphs solutions (F-measure) on *Synthetic-small* dataset with parameter $p = 1/|V|$, $p = 3/|V|$, and $p = 5/|V|$ for two phases (L-DP and LocExt) of the heuristic and DP

	Time	Density	Intervals F-measure	Subgraphs F-measure
$p = 1/|V|$				
DP	304.45	127.08	1	1
L-DP	0.37	89.90	0.52	0.84
LocExt	0.06	94.98	0.65	0.95
$p = 3/|V|$				
DP	339.90	127.12	1	1
L-DP	0.44	89.96	0.50	0.80
LocExt	0.09	96.69	0.66	0.94
$p = 5/|V|$				
DP	392.76	127.18	1	1
L-DP	0.49	90.05	0.52	0.83
LocExt	0.07	95.13	0.64	

Running time is in seconds and the results are averaged over 100 examples for each value of p

solutions of density approximately 75% of the optimal density for all values of p. For the interval quality (that is the timestamps in the planted intervals that are correctly identified by the heuristic), the average F-measure is between 64 and 66%, showing that the heuristic is able to identify correctly most of the timestamps in the planted intervals. Similarly, for subgraphs quality, the average F-measure is between 93 and 95%, showing that the heuristic is able to identify correctly a large part of the nodes in the planted communities.

The experimental results show that LocExt is capable of improving the detected solution of the first phase of the heuristic (L-DP) on the *Synthetic-small* dataset in a reasonable amount of time. Specifically, even in the worst-case scenario where $p = 3/|V|$, LocExt only requires 20% of the L-DP running time. Additionally, LocExt is able to improve the density of L-DP by a minimum of 5.6%

(for $p = 5/|V|$) and up to 7.5% (for $p = 3/|V|$). In terms of interval and subgraph quality, LocExt improves the F-measure compared to the solution provided by L-DP. The F-measure of interval quality is improved by at least 23% (for $p = 5/|V|$) and up to 32% (for $p = 3/|V|$), while the F-measure of subgraph quality is improved by a minimum of 12% (for $p = 5/|V|$) and up to 17.5% (for $p = 3/|V|$). Lastly, the running time of the heuristic on the *Synthetic-small* dataset is much faster than DP, with a maximum speedup of 708 times (for $p = 1/|V|$) and a minimum of 641 times (for $p = 3/|V|$).

On the *Synthetic1* dataset, the results of Table 2 show that LocExt improves the density of solutions generated by L-DP for k-ℓ-Densest-Alignment-Episodes by at least 12.9% (for $p = 1/|V|$) and at most 16.9% (for $p = 5/|V|$), while improving the F-measure of interval and subgraph quality by at least 50% (for both $p = 1/|V|$ and $p = 3/|V|$) and at most

Table 2 Density, running time and quality of the interval and subgraphs solutions (F-measure) on *Synthetic1* dataset with parameter $p = 1/|V|$, $p = 3/|V|$, and $p = 5/|V|$ for two phases (L-DP and LocExt) of the heuristic

	Time	Density	Intervals F-measure	Subgraphs F-measure
$p = 1/|V|$				
L-DP	24.97	607.17	0.36	0.66
LocExt	10.35	685.4	0.54	0.77
$p = 3/|V|$				
L-DP	24.99	607.17	0.38	0.70
LocExt	9.38	689.51	0.57	0.82
$p = 5/|V|$				
L-DP	32.89	607.18	0.41	0.75
LocExt	13.70	709.89	0.63	0.89

Running time is in seconds and the results are averaged over 100 examples for each value of p

Table 3 Density, running time and quality of the interval and subgraphs solutions (F-measure) on *Synthetic2* dataset with parameter $p = 1/|V|$, $p = 3/|V|$, and $p = 5/|V|$ for two phases (L-DP and LocExt) of the heuristic

	Time	Density	Intervals F-measure	Subgraphs F-measure
$p = 1/|V|$				
L-DP	738.07	1413.38	0.18	0.45
LocExt	491.28	1772.62	0.41	0.57
$p = 3/|V|$				
L-DP	843.30	1413.38	0.18	0.47
LocExt	521.10	1755.78	0.40	0.57
$p = 5/|V|$				
L-DP	960.78	1413.38	0.21	0.54
LocExt	382.40	1733.23	0.41	0.64

Running time is in seconds and the results are averaged over 100 examples for each value of *p*

54% (for $p = 5/|V|$), and by at least 16.7% (for $p = 1/|V|$) and at most 18.7% (for $p = 5/|V|$), respectively. The running time of LocExt is also reasonable, being only 37–42% of L-DP running time for k-ℓ-Densest-Alignment-Episodes, depending on the value of parameter *p*.

Table 3 displays the results of the heuristic on the larger *Synthetic2* dataset, which indicates that LocExt improves the density of the solutions produced by L-DP by a minimum of 22.6% (for $p = 5/|V|$) and a maximum of 25.4% (for $p = 1/|V|$). Additionally, LocExt improves the F-measure of interval and subgraph quality by at least 95% (for $p = 5/|V|$) and up to 128% (for $p = 1/|V|$), at least 18% (for $p = 5/|V|$) and at most 27% (for $p = 1/|V|$), respectively. The running time of LocExt is also reasonable, being only 40–67% of L-DP running time, depending on the value of parameter *p*.

Overall, the results on the synthetic datasets show that LocExt is able to improve the quality of solutions returned by the L-DP heuristic for k-ℓ-Densest-Alignment-Episodes. However, further analysis of the proposed method using a real-world dataset is presented in the next part of the experimental.

4.2 Real network

In the second phase of our experimental analysis, we test the heuristic using a real dual network dataset.

4.2.1 Dataset

DBLP-network. We build a DBLP-network dataset extracting a list of research papers available in the DBLP computer

Table 4 Properties of the alignment network

Graph	Representation	Nodes	Temporal edges	Timestamps
DBLP-network	co-authorship	188	339	10

science bibliography. We focus our analysis on the community of researchers on algorithms for bioinformatics between 1993 and 2002, a period when this community started to establish. The dataset is build starting from a well-known author in this community, "*Dan Gusfield*," then considering his co-authors and the co-authors of the co-authors. The dataset is built considering ten timestamps, one for each year between 1993 and 2002.

We constructed a conceptual graph in each timestamp as a co-authorship-weighted edge network, that is two authors are joined by an edge when they published at least one paper together in that year.

The edge weights are determined considering the number of shared publications between two authors and then by applying the *logistic regression* function with parameter $c = 0.6$, in order to obtain standardized values between 0 and 1.

The physical graph is built by defining an edge that connects two authors when they participate a same conference (i.e., they both published a paper in the conference) in a specific year. Informally, this graph represents the relation between researchers who share interests but are not necessarily co-authors.

Then, the aligned graph is built by aligning the conceptual and physical graph in each timestamp. Table 4 reports number of nodes, overall temporal edges and timestamps of the alignment network.

Fig. 5 Subgraphs detected in the temporal network of BDLP by our heuristic with $k = 2$
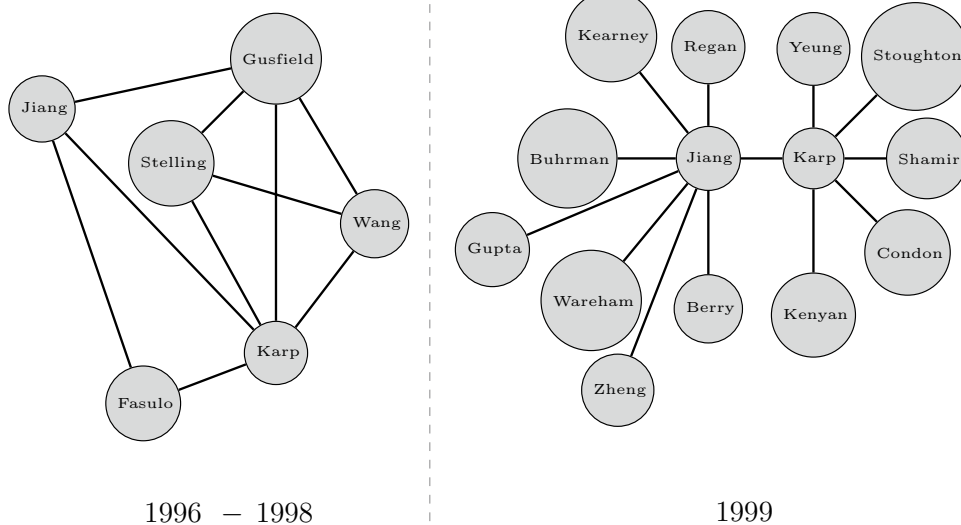

$$1996 - 1998$$

$$1999$$

Table 5 Densities, intervals and running time (in seconds) on real network dataset of the LocExt

Dataset		$k = 2$	
DBLP-network	Interval	(1996,1998)	(1999,1999)
	Density	1.96	1.61
	Time	0.068	

4.2.2 Outcome

We consider a modest value of k, that is, $k = 2$, because the real-network dataset we investigate is defined over a short number of timestamps (i.e., 10). Table 5 reports the densities and intervals of the solutions returned by LocExt, as well as the LocExt's running time. Moreover, the subgraphs discovered by LocExt on the DBLP dataset are displayed in Fig. 5. In this dataset, the LocExt found a temporal subgraph of six nodes with a density of 1.96 over a three-year interval (1996–1998); then a temporal subgraph of fourteen nodes with a density of 1.61 on interval of one timestamp (1999). Notice that these two subgraphs share two nodes, "Karp" and "Jiang," two well-known and active researchers in algorithms for bioinformatics community. Notice also that the 1999 subgraph contain a bridge between "Karp" and "Jiang," and hence its removal disconnects this subgraph. Furthermore, each node, except "Karp" and "Jiang," have degree one. The 1996–1998 subgraph has a different structure and all the nodes have degree at least two.

5 Conclusion

We presented a novel network model called temporal dual networks, which addresses the challenge of modeling interactions that change over time and integrates information from two different networks. We tackled a fundamental problem in graph mining by finding densest subgraphs in the new proposed model through network alignment and dynamic programming. However, due to the computational complexity of dynamic programming, we introduced a heuristic approach that constraints the interval lengths of the temporal subgraphs we look for and furthermore exploits a local search procedure. We provided experimental results that demonstrated the effectiveness of our method on synthetic datasets, even for temporal graphs with 10000 vertices and 10000 timestamps. Finally, we presented a case study on a real case obtained by extracting data from DBLP.

Future works include an extension of the experiments, particularly considering more real cases to investigate. Another interesting future direction is providing parallel implementations of our methods, particularly for the exact dynamic programming algorithm, to deal with larger datasets.

Declarations

Conflict of interest The authors declare they have no financial interests

References

Akrida EC, Mertzios GB, Spirakis PG, Zamaraev V (2020) Temporal vertex cover with a sliding time window. J Comput Syst Sci 107:108–123

Braha D, Bar-Yam Y (2009) Time-dependent complex networks: Dynamic centrality, dynamic motifs, and cycles of social interactions. Adaptive networks. Springer, Cham, pp 39–50

Castelli M, Dondi R, Hosseinzadeh MM (2020) Genetic algorithms for finding episodes in temporal networks. In: Cristani M, Toro C, Zanni-Merk C, Howlett RJ, Jain LC (eds.) Knowledge-based and intelligent information & Engineering systems: proceedings of the 24th international conference KES-2020, Virtual Event, 16-18 September 2020. Procedia Computer Science, vol. 176. Elsevier, pp 215–224

Charikar M (2000) Greedy approximation algorithms for finding dense components in a graph. In: Approximation algorithms for combinatorial optimization, third international workshop, APPROX 2000, Proceedings. pp 84–95

Chen J, Saad Y (2010) Dense subgraph extraction with application to community detection. IEEE Trans Knowl Data Eng 24(7):1216–1230

Chen T, Bonchi F, Garcia-Soriano D, Miyauchi A, Tsourakakis CE (2022) Dense and well-connected subgraph detection in dual networks. In: Proceedings of the 2022 SIAM international conference on data mining (SDM). SIAM, pp 361–369

Cinaglia P, Cannataro M (2022) Network alignment and motif discovery in dynamic networks. Netw Model Anal Health Inform Bioinform 11(1):38

Dondi R, Guzzi PH, Hosseinzadeh MM (2023) Integrating temporal graphs via dual networks: dense graph discovery. In: Cherifi H, Mantegna RN, Rocha LM, Cherifi C, Micciche S (eds) Complex networks and their applications XI. Springer International Publishing, Cham, pp 523–535

Dondi R, Hosseinzadeh MM (2021) Dense sub-networks discovery in temporal networks. SN Comput Sci 2(3):1–11

Dondi R, Hosseinzadeh MM, Guzzi PH (2021) A novel algorithm for finding top-k weighted overlapping densest connected subgraphs in dual networks. Appl Netw Sci 6(1):40

Dondi R, Hosseinzadeh MM, Mauri G, Zoppis I (2021) Top-k overlapping densest subgraphs: approximation algorithms and computational complexity. J Comb Optim 41(1):80–104

Galbrun E, Gionis A, Tatti N (2016) Top-k overlapping densest subgraphs. Data Min Knowl Discov 30(5):1134–1165

Galicia JC, Guzzi PH, Giorgi FM, Khan AA (2020) Predicting the response of the dental pulp to sars-cov2 infection: a transcriptome-wide effect cross-analysis. Genes Immun 21(5):360–363

Goldberg AV (1984) Finding a maximum density subgraph. Tech. rep, Berkeley

Gu S, Jiang M, Guzzi PH, Milenković T (2022) Modeling multi-scale data via a network of networks. Bioinformatics 38(9):2544–2553

Guzzi PH, Milenković T (2018) Survey of local and global biological network alignment: the need to reconcile the two sides of the same coin. Brief Bioinform 19(3):472–481

Guzzi PH, Salerno E, Tradigo G, Veltri P (2020) Extracting dense and connected communities in dual networks: an alignment based algorithm. IEEE Access 8:162279–162289

Guzzi PH, Tradigo G, Veltri P (2021) Using dual-network-analyser for communities detecting in dual networks. BMC Bioinform 22(15):1–16

Holme P, Saramäki J (2012) Temporal networks. Phys Rep 519(3):97–125

Hosseinzadeh MM (2020) Dense subgraphs in biological networks. International conference on current trends in theory and practice of informatics. Springer, Cham, pp 711–719

Hosseinzadeh MM, Cannataro M, Guzzi PH, Dondi R (2023) Temporal networks in biology and medicine: a survey on models, algorithms, and tools. Netw Model Anal Health Inform Bioinform 12(1):10

Kawase Y, Miyauchi A (2018) The densest subgraph problem with a convex/concave size function. Algorithmica 80(12):3461–3480

Kempe D, Kleinberg J, Kumar A (2002) Connectivity and inference problems for temporal networks. J Comput Syst Sci 64(4):820–842

Kostakis O, Tatti N, Gionis A (2017) Discovering recurring activity in temporal networks. Data Min Knowl Discov 31(6):1840–1871

Milano M, Milenković T, Cannataro M, Guzzi PH (2020) L-hetnetaligner: a novel algorithm for local alignment of heterogeneous biological networks. Sci Rep 10(1):1–20

Rozenshtein P, Gionis A (2019) Mining temporal networks. In: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining. ACM, pp 3225–3226

Wu H, Cheng J, Huang S, Ke Y, Lu Y, Xu Y (2014) Path problems in temporal graphs. Proc VLDB Endow 7(9):721–732

Wu Y, Zhu X, Li L, Fan W, Jin R, Zhang X (2016) Mining dual networks - models, algorithms, and applications. TKDD