Journal of Computer Science and Technology

Completed Review

Reviewer 3:

Reviewer Affiliation	Czech Technical University in Prague
Manuscript ID:	
Manuscript Type:	Original Article
Keywords:	I.2.8.d Dynamic programming < I.2.8 Problem Solving, Control Methods, and Search < I.2 Artificial Intelligence < I Computing Met , I.2.8.e Graph and tree search strategies < I.2.8 Problem Solving, Control Methods, and Search < I.2 Artificial Intelligence < I , H.2.4.m Temporal databases < H.2.4 Systems < H.2 Database Management < H Information Technology and Systems
Speciality:	Data Management and Data Mining
Date Submitted:	*blinded*
Manuscript Title:	
Date Assigned:	03-Jun-2022
Date Review Returned:	14-Jul-2022

req **Would you be willing to be recommended to other journal(s) of this field as a reviewer?**

Yes

Overview Report	**Outstanding**	**Good**	**Fair**	**Poor**	**Very Poor**
Content					
req Is the work relevant?					✓
req Is the work original?				✓	
req Are the methods/proofs/experiments/etc. sound and convincing?					✓
Presentation					
req Is the abstract an adequate summary of the work?			✓		
req Are the background and related work(s) clearly introduced?					✓
req Are the methods/proofs/experiments/etc. properly stated?					✓

req Are the conclusions clear and adequate? ✓

req Are the references adequate? ✓

req Is the presentation clear to the relevant audience? ✓

req Are the overall organization and length of the manuscript adequate? ✓

req Is the English satisfactory? ✓

Summary Report

req **Quality**

Erroneous or Trivial

Priority of Publication

If accepted, should the manuscript be prioritized for publication? No

Accepted as

Regular Paper

req **Confidence of your evaluation**

4

req **Recommendation**

Accept without revision

Accept with minor revision

Review again after major revision

Resubmit after major revision

Submit to another journal

✓ Reject

req **Would you be willing to review a revision of this manuscript?**

✓ Yes

No

Comments

Confidential Comments to the Editor

Comments to the Author

This manuscript concerns with the problem of Group Steiner Tree in temporal graphs.
Here a temporal graph consist of vertices and edges, each edge being directed, with a departure and an arrival time and a cost. Each vertex is assigned a subset of labels. The task is, given a root r being a vertex of the graph and a subset of labels L to find a minimum cost subgraph such that for each label in L there is a vertex wearing that label and a temporal path (waiting in vertices allowed) from r to that vertex in the found subgraph.

The authors consider the problem in parameterized setting with the parameter being the number of labels. The authors give several motivations for studying the problem, but these seem non-concrete and artificial.

The problem can be solved by transforming the input temporal graph into a standard oriented graph by splitting each vertex into its time occurrences. This only doubles the number of edges.

Then one can use any algorithm for Group Steiner Tree in directed graphs or (by a standard transformation) any algorithm for Directed Steiner Tree (DST) to solve the problem.
There are many such algorithms, as most algorithms for Steiner Tree in undirected graphs parameterized by the number of terminals actually are capable of solving DST with little adjustment.

The authors select one algorithm which solves a slightly more complex task (finding top-k result in a database search) as a Baseline algorithm for a comparison.

Then they propose their own algorithm for the task. The algorithm is still based on the dynamic programming of Dreyfus and Wagner from 70's, with some heuristic improvements dropping some of the table entries (or states as the call it) and some rather obvious preprocessing. The improvements include dropping some partial solution based on two simple lower bounds on the cost of completing such a solution and also changing the processing order of the states in an A*-manner. To make use of these drops, they replace the table entries with a queue of processed states. Then, when checking whether such a state was already processed they have to scan the whole queue, which results in a horrible worst-case time complexity.

However, the authors conduct also few experiments on some real word temporal graphs where the implementation of their algorithm significantly outperforms (their?) implementation of the selected Baseline algorithm. Unfortunately, the authors do not seem to provide the implementation.

The manuscript contains several proofs, e.g. that some states "can be dropped" (this is the actual formulation of the lemma). Many of them are flawed, one of them is a "proof by example". The others are buried in clumsy notation so much that I was not able to follow. For example, it is not clear whether "a state" is just the triple, an associated cost value, or the corresponding subgraph. There many other notions never defined, or if defined, used in contrariety with the definition. E.g., if you intersect a sequence of edges with a temporal graph, would you expect the outcome to be a set of vertices?

The manuscript is in general poorly written. Many sentences are not in proper English, cluttered with many unnecessary words, failing to express the idea. There are bad (single character) hyphenations. The overall structure is strange - the related work only come just before conclusion and only focuses on stating that "no one did exactly what we did", missing all the above mentioned algorithms.

I believe that any graduate student should be able to do this routine adaptation work more properly than this manuscript does. If she is not able, then this manuscript would not help her, because the description is messy. However, none of the ideas is anyhow involved, they are merely simple observations.

Hence I see no reason for this manuscript to be published and I suggest to reject it.

Detailed comments (excluding typos, grammar errors, and similar minor things):
----

p 1 l 30: Helicobacter pylori is not a virus. Apparently you do not know even basics from biology.

Then why trying to come up with biology motivation? What does "affected time" mean?

p 1 l 54: highly useful - for what purpose? Where the "group" aspect comes into play?

p 2 l 19: some given resource - do you have some example?

p 2 l 44: The algorithm dates back to Dreyfus and Wagner

p 2 l 53: an improved dynamic algorithm (menaning?) is devised in ... - there are many algorithms for ST that can be used for DST and hence also for GST in directed graphs. See https://drops.dagstuhl.de/opus/volltexte/2019/10227/

p 3 l 28: related work is to put your result in to context and should come at latest after properly defining the problem.

p 4 l 27-30: Connected tree - meaning? ; the root - was it declared or defined? Is it clear what is a root of a "connected tree"? ; paths ... are reachable - meaning?

p 4 l 48: You may also assume for simplicity that every edge exists only between t_\alpha and t_\beta (then you can simply take all paths from r to v etc.) and that L = \mathcal{L}.

p 4 l 38: GST is NP-hard - ST is already on the Karp's list of 21 NP-complete. Clearly any its generalization will be also NP-hard. No need a Thm for that and [14] is a wrong reference.

p 5 l 15-17: which is the best known exact algorithm for the keyword search in a relational database [17]. - I did not find any such clear statement in [17]. Furthermore, it definitely does not mean that it is the best algorithm for GST.

p 5 l 30: So, it cannot be applied for our problem. - It can, the changes needed are small and straightforward.

p 5 l 57: Third, we check if the new state is an optimal solution. - How do you find out?

p 5 l l 15-18: is based on the fact that the root of a state can reach to any vertex covering the labels in an undirected graph. - No it is not.

p 5 l 56 x 58: [20] is focus(ed) on how to reduce the generated states in the search space. There are no methods for reducing ... the number of states to be generated. - ?? What is the difference?

p 6 l 12: Given a temporal graph G = (V;E) and a set of query labels L. - not a sentence

p 6 l 49: this means that the path time constraint does not hold. - for what path?

p 6 step 3 - a detailed description of topsort implementation restricted to vertices without interesting labels?

p 6 l 50: may be isolated vertices - why not removing all vertices with A(v) = \infty?

p 7 l 6: done in the constant time - describe the computational model you use

p 7 l 22: to each vertex covering the labels in L. - meaning?; even covering never defined.

p 7 l 33: state-transition equation - meaning?

p 7 (1): minimum is taken over the weights not the trees itself

p 7 (2): Tg(s(e);X; ts(e)) - does not make sense! Is e fixed or not? ; what does \oplus stand for?

p 7 (3): the minimum should be taken over all X_1, X_2 and t_1, t_2 such that ...

p 7 l 50-58: This should give a recipe how to determine T_g(v,X,t).

p 8 l 5-12: again, describe how to obtain T_m(v,X,t) for given v,X,t.

p 8 l 33: what is P_{v,t}? Any particular path? From P(v,t)? This does not follow from the definition.

p 8 l 33: an intersection of a sequence of edges and a tree (which probably either a graph (a pair of vertices and edges) or a set of edges) is a set of vertices?

p 8 l 44: merge of two graphs is a state?

p 8 l 58: (r;X0; S(Pv;t)) = (r; L; S(Pv;t)) that is a tree - no, it is a state.; Recall that the corresponding tree is denoted by T(....).

p 8 l 17: What is P(v)? Why is W(T_2)=W(P') + W(T(...))? It can be less! They can share edges!

p 8 l 29: Without loss of generality, suppose that there are two common vertices u and v included in both - how come that this is without loss of generality? There can be more such vertices!

p 8 l 44: denote the cycle graph in Figure 2(a), - what if the situation looks differently than in Fig 2(a) ?

p 8 l 45: the merger - meaning?

p 8 l 54: Why is W(G')= W(P...) + W(T...)? It can be less!

p 10 l 34: Then, we push the expanded state and its cost value into Q (Line 31). - What if cost>UB?

p 10 l 44: the worse state need be removed from Q. - How is that done?

p 10 l 28-47: What if the expanded state (v,X,t) is better than existing state (v,X,t) - clumsy notation

p 12 l 40: the remaining tree - how do we know it is a tree?

p 12 l 52: \stackrel{\triangleup}{=} - meaning?

p 12 l 21: H = (r; L; t0)-(v;X; t), what does - represent here on the triples (states)? Coordinate-wise?

p 13 l 15: the tree H contains all paths from r to the vertices that cover the labels in X. - How does it follow? Meaning?

p 13 l 17: Then W(P) \le W(H). - Why?

p 13 l 59: label-based lower bound is not consistent. - example?

p 16 l 56: approximate answers - meaning?

p l6 l 30: then T_i may be pruned - It may also be pruned since it is "bad" or when taking the minimum somewhere.

table 1: the components are largest? in what sense? average? or what?

p 17 l 30: selected components - how do you select them? why more than one?

p 20 l 39: These results demonstrate that - that there is no point in progressive search. It just slows down the search for the optimum which can be found in less than twice the time.

p 21 l 37: Helicobacter pylori, staphylococcus(,) and chlamydia trachomatis are bacteria.

p 21 Case study: - What is the outcome of the "case study"? What is the conclusion?

Reviewer opted in to receive recognition on Publons? (Yes/No answer required)

✓ Yes

No

Use the below rating options to rate the reviewer on <u>this</u> submitted review. The rating