STACS 2024 review response (submission 63)

STACS 2024 <stacs2024@easychair.org>
Wed 2023-11-22 08:58
To:KLOBAS, NINA <nina.klobas@durham.ac.uk>

[You don't often get email from stacs2024@easychair.org. Learn why this is important at https://aka.ms/LearnAboutSenderIdentification]

[EXTERNAL EMAIL]

Dear Nina,

Thank you for your submission to STACS 24. The STACS 24 rebuttal period will be between Nov. 21 (AoE) and Nov 24 (AoE).

During this time, you will have access to the current state of your reviews and have the opportunity to submit a response of up to 1000 words. Please keep in mind the following during this process:

The response must focus on any factual errors in the reviews and any questions posed by the reviewers. It must not provide new research results or reformulate the presentation. Try to be as concise and to the point as possible.

The rebuttal period is an opportunity to react to the reviews, but not a requirement to do so. Thus, if you feel the reviews are accurate and the reviewers have not asked any questions, then you do not have to respond.

The reviews are as submitted by the PC members, without any coordination between them. Thus, there may be inconsistencies. Furthermore, these are not the final versions of the reviews. The reviews can later be updated to take into account the discussions at the program committee meeting, and we may find it necessary to solicit other outside reviews after the rebuttal period.

The program committee will read your responses carefully and take this information into account during the discussions. On the other hand, the program committee will not directly respond to your responses, either before the program committee meeting or in the final versions of the reviews.

Your response will be seen by all PC members who have access to the discussion of your paper, so please try to be polite and constructive.

The reviews on your paper are attached to this letter. To submit your response you should log on the EasyChair Web page for STACS 24 and select your submission on the menu.

----------------------- REVIEW 1 ---------------------
SUBMISSION: 63
TITLE: Temporal graph realization from fastest paths
AUTHORS: Nina Klobas, George Mertzios, Hendrik Molter and Paul Spirakis

----------- Overall evaluation -----------
This paper addresses a graph realization problem for shortest paths on periodic temporal graphs. The authors do not seem to be addressing the actual shortest path problem on a given periodic temporal graph, but rather given a distance matrix, the goal is to find a graph plus distances that are the same as those in the distance matrix.
One thing that is confusing is the working of the main problem (Simple TGR on page 3). They ask if there exists a graph, but in fact it seems that they are requiring the algorithm to assign distances to the underlying graph of the distance matrix (i.e. the underlying graph G is the support of the distance matrix D--although this is not really stated explicitly). Thus, it might be better phrased to say that we are given G and we are looking for periodic distance labels.

In any case, the authors give results on this problem, spending most
of their time on hardness results. THey show that the problem is
NP-hard in general and the problem is W[1]-hard with parameter
feedback vertex number. For algorithmic results, they show the
problem has polynomial-time alg when input graph G is a tree and there
is an FPT algorithm parametrized by feedback edge set size. The
motivation for using feedback edge number as a parameter is that there
is a fast algorithm for trees. This algorithm is based on
constructing an ILP. It might be better to present the algorithmic
ideas first and in more detail and put the hardness proofs in the
appendix.

The problem is interesting, but the new ideas are not particularly
well explained and lost in technical details. It seems borderline for
STACS.

comments:

figure 1 should say path from v1 to v5 rather than u to v.
If it takes 21 time units (beginning at time 0) to go from v1 to
v5, why do you say that the fastest temporal path is 15?

----------------------- REVIEW 2 ---------------------
SUBMISSION: 63
TITLE: Temporal graph realization from fastest paths
AUTHORS: Nina Klobas, George Mertzios, Hendrik Molter and Paul Spirakis

----------- Overall evaluation -----------
The authors introduce a problem on realizing (periodic) temporal graphs.
In graph realization problems, one is given some observed data and asks whether there is a graph that fits this observed data.
The most basic graph realization problem is given an nxn distance matrix D and the task is to determine whether a graph exists, where the distance between any two vertices v_i and v_j is equal to D_{i,j}.
This problem is well analyzed and shown to be polynomial time solvable.
In this paper, the authors generalize the problem to (periodic) temporal graphs, where each edge occurs exactly once per period.
Here, a (periodic) temporal graph is a (periodic) sequence of static graphs.
The authors measure the distance from vertex u to vertex v by the shortest duration of any temporal path from u to v, that is, a path from u to v, where per time step at most one edge of this path can be traversed.

In contrast to the realization problem on static graphs, the authors show that the considered problem on temporal graphs is much harder.
They show that the problem is NP-hard even when the period of the temporal graph is 3.
Based on this hardness, the authors try to find efficient algorithms to solve the problem under certain conditions.
They show that the problem is polynomial time solvable on trees and analyze whether FPT-algorithms (algorithms run in f(k) poly(n) time) for parameters k that measure the distance to trees.
On the negative side, for the vertex deletion distance to trees (feedback vertex set number), the authors show that an FPT-algorithm is unlikely.
On the positive side, they show that parameterized by the edge deletion distance to trees (feedback edge set), an FPT-algorithm can be achieved.
The latter algorithm relies on many guessing steps that are validated by an ILP.

The proofs are mostly contained in the Appendix and the main body of the paper mainly consists of intuitions.
Based on the length of the Appendix, I was not able to validate all proofs but the given intuition seems plausible.
I think this problem is very interesting and may lead to many promising new research directions on (periodic) temporal graphs.
Moreover, since all the presented results are very technical, I recommend this paper for STACS, even though most of the technical parts are not contained in the short version.

Minor remarks:

I found no definition/small explanation what an FPT-running time is. Please provide some.
It seems that you define \mod as: x \mod x = x and not x \mod x = 0. Please note this in the preliminaries.

l 230: Please explain more in detail how this construction works. How do you add these new edges in a way that each vertex still has degree at least one in each other color class?
l 234: Maybe add some sentence to clarify that each edge still only achieves a finite label.
l 338: Please reformulate the sentences that refer to note that some properties for the distance matrix hold. In the short version of the paper, the distance matrix is not defined and the reader cannot 'note' that.
l 401: "e' s closer to e* than e" This is only true if you chose e' to be the edge that let the algorithm add edge e to the queue. Otherwise, e' and e may be of same distance to e*
l 413 + 414: v_\ell -> v_{\ell + 1}
l 413 + 414: v_j -> v_{j + 1}
l 435: "can be upper bounded by a function" It seems to be folklore but could you please still add some reference
l 441: why are the names of the three subsets relevant? I think for the intuition you could skip this.
l 455: I do not completely understand what you want to say with the last item of the itemize. Could you plese reformulate the sentences?
l 467: Please note somewhere that G' is still connected.
Figure 5: Why is none of the two degree 1 neighbors of the bottom left green vertex part of Z*?
l 479: Please provide more argumentation for this fact.
l 482: and others: Z is defined as a vertex set but you often use it as a graph.
l 487: "we assume that a tree" -> "we denote by" (?)
l 505: Is the segment S_{u,v} unique? If a single edge between two vertices of U counts as a segment, then I assume that there might be at least two segments between these vertices. Please clarify whether single edges count as segments. Please also highlight that each inner vertex of a segment S_{u,v} is contained only in the segments S_{u,v} and S_{v,u} and has no neighbors outside of these segments.
l 523 + 528: line segment -> segment

----------------------- REVIEW 3 ---------------------
SUBMISSION: 63
TITLE: Temporal graph realization from fastest paths
AUTHORS: Nina Klobas, George Mertzios, Hendrik Molter and Paul Spirakis

----------- Overall evaluation -----------
The authors study a graph realization problem, where one has to decide whether a temporal graph exists such that the given matrix values correspond to the fastest travel times between the vertices of the graph. Numerous non-trivial results are presented including NP-hardness and W[1]-hardness results, exact polynomial-time algorithms and finally a FPT algorithm which makes use of ILP.

I did not have the time to go through the (large) appendix, and thus I am not certain if the W[1]-hardness proof and the feedback edge number FPT algorithm are correct. The authors tried their best to give lots of details concerning some parts but refer to the appendix at other crucial parts. Some intuition is given instead but even this fell short for me personally. More on this in my major remarks.

The problem is original, simple to understand, and (seemingly) hard to analyse; several remarkable techniques are used and combined in non-trivial ways; and lastly the paper was mostly a pleasure to read (outside the gaps in reasoning). The quality and subject are in my opinion suitable for STACS, although the size of this work makes it an awkward fit for a conference paper and seemingly more suited for a journal.
If another reviewer or the program committee can ensure that the paper's results are indeed valid, then I don't see why this paper shouldn't be accepted. That is of course a big "if", I apologize for not doing it myself.

MAJOR REMARKS:

- Most details for the W[1] hardness proof are in the main paper, but it is missing one crucial part: we know how to construct the graph which is built from the instance of Multicoloured Clique, and which gives us the entries of the matrix for Simple TGR which are exactly of value 1; also we know how to correct the infinity values in the matrix and make the temporal graph periodic of a precise period. BUT, what are all the other values for the matrix? This is key to understand what temporal paths are created through where and with what duration, if the "a" and "b" paths of vertices are shortcuts or "speedbumps" instead, etc. Is it too optimistic of me to think that now that the whole structure of the instance of Clique has already been transformed into the underlying graph, the remaining values of the matrix depend solely on this underlying graph and should thus be relatively easy to describe?

- Concerning the FPT algorithm for feedback edge number: please elaborate on what "guessing" means. This term was introduced with quotation marks but is never actually explained and is used all over the proof. Does it mean that all possible "guesses" are enumerated and tested? Or rather that an existing labeling is extended by one more label on one more incident edge such as in the algorithm for trees? It is simply unclear and makes the whole proof and thus result unsteady.

- Some more on the FPT algorithm, more precisely "guesses" G-1 up to G-12:

None seem to bother with the edges of F (or are hidden in the appendix) which seems to indicate that no fastest paths (but the ones between the edge vertices) travel through F, which are chosen seemingly arbitrarily through the choice of some initial spanning tree. How do the authors prove that this is in fact the case, and does this imply a multitude of solutions exist if one exists, corresponding to all possible spanning trees/F?

MINOR REMARKS:

- A discussion on why the authors focused on periodic temporal graphs may be interesting, i.e. does this make the problem harder, or is this why it is possible to simplify to one label (per period), ...?

- Similarly, why fastest? A discussion on foremost and shortest temporal paths instead may be interesting, and also increasing vs non-decreasing time labels.

- line 107: "form"
"from"

- Shouldn't the abbreviation for "Simple Periodic Temporal Graph Realization" be "Simple Periodic TGR" instead of only "Simple TGR"? This may be confusing especially since the presented reduction is first to a non-periodic version of the problem, and afterwards to the periodic version (so in a sense "Simple TGR" exists in the paper as well, although you focus on "Simple Periodic TGR").

- lines 436-438:
Why not? If f(k) is just applied O(n^2) times, the result is still an FPT algorithm, correct? Is there maybe an issue with all (or some of) the O(n^2) results interdepending, thus not making it multiple independent problems?

- Z*: I'm not sure why Z* exists, surely having the fastest path to the clip vertex suffices to know the fastest paths to the vertices of the trees, and if not, why does having one arbitrary vertex of the tree in the important vertices matter exactly?

- lines 496-498:
This should be a lemma with proof because I don't know how these numbers are obtained.

- line 506 "Su,v \neq Sv,u"
To be more precise: one path is the reverse of the other?

Best wishes,