## Topic:

* “The work is suitable for the conference as graph layout is certainly a core topic and the system presented is valuable.” – R2
* “Otherwise, the article is very well written and the system is nice, smart and useful.” – R3
* “The strengths of this paper are that it makes headway in a significant problem in the application of constrained layouts and that the choices made in terms of defining sets and constraints seem reasonable and are clearly explained.” – R1
* “It addresses a clear problem (of low-level constraint specification being arduous) and makes a useful contribution to the literature” – R4
* “It appears from the writing and examples that SetCoLa is focused on presentation over exploratory analysis — there is a discussion of tweaking individual node constraints and matching existing presentations. Is this the intent? What design considerations were made to support the intent?” – R1

## Engineering/Systems Contribution:

* “While the scientific contribution is low, the engineering level seems to be quite high. Thus with rather limited scientific research but a focus on the engineering the paper is **clearly in the systems category** but still fits well from the topic.” – R2
* “It's definitely more of an engineering effort than contributing to scientific research, but the result seems quite useful.” – R2
* “my only major suggestions would be for making the **code available** and mentioning this from the paper (possibly just left out for anonymous review purposes)” – R4

## Prior Work:

* “It does provide some insight into the design options and choices as well as comparison to other systems and frameworks, but its a bit tight.” – R2
* “The previous work is also reviewed in a suitable manner.” – R2
* “Citations seem fairly complete, with the exception of: **Yunhai Wang et al.** InfoVis 2017 (TVCG Jan 2018) “Revisiting Stress Majorization as a Unified Framework for Interactive Constrained Graph Visualization.””
* “The approach described in the paper shares more in common with **libcola** (the percursor to WebCoLa) than is described in the Related Work section. libcola has the concept of high-level constraints over groups of nodes, which it automatically turns into a set of low-level separation constraints. These **compound constraints** (listed in the [documentation](http://www.adaptagrams.org/documentation/classcola_1_1CompoundConstraint.html)), as they are called, are alignment, boundary, distribution, fixed relative position, multi-separation, and page boundary constraints. While similar (boundary constraints have the same purpose as SetCoLa position constraints, multi-separation constraints have the same purpose as SetCoLa order constraints), these compound constraints each work with a list of nodes rather than being able to be specified among sets or groups, i.e., ordering of groups, as they are in SetCoLa. I'd like to see this discussed.” – R4tre

## Description of system:

* “With the limited scope of a conference paper, even though it's published as CGF article, maybe it's a **bit light on the system aspects**.” – R2
* “The description is good but within the limited scope of a conference article it feels a bit constrained. Nevertheless, it does a good job of a top-down description of the framework.” – R2
* “The high-level constraints introduced by SetCoLa consist in specifying constraints on "sets" of vertices. Multiple constraints can apply to the same sets or overlapping sets. SetCoLa defines several operators, although a manageable number, to specify partitioning, selecting of nodes into sets with different mechanisms, and applying constraints to the sets.” – R3
* “The paper explains well how the SetCoLa compiler transforms the high-level constraints into lower-level ones that WebCoLa can understand, with the notable exception of circular constraints that require more advances processing/hacking.” – R3
* “This paper presents SetCoLa, a domain-specific language for specifying constraints to WebCoLa. The paper describes the SetCoLa basic operations of defining sets of nodes and applying constraints over them. Some of the constraints are natively supported by WebCoLa and others are hinted to WebCoLa through a more complicated set of dummy nodes and links generated and later removed by SetCoLa and new to this work.” – R1
* “I also like the inclusion of automatic calculation of common graph theoretic properties to augment the properties already included in the user’s dataset.” – R1
* “discussion of some aspects of the system I expect would be helpful, such as a **compilation & layout steps**, **compilation runtime**, and **design trade offs** are not included.” – R1
* “some further discussion of how well (or not) SetCoLa works when the graph has no starting positions (or starts from a r**andom layout**).” – R4
* “My only thought is that I wonder if it would be worth **clarifying the term constraint** early in the paper (i.e., saying the are geometric relationships among positions of nodes in either the x- or y-dimension that must be maintained), this is sort of assumed.” – R4
* “I would also like to know the **edges that are added to produce the circle constraints**. It wasn't clear to me which nodes the 56 new edges were between to produce the 16 node circle. From the description it looks like 16 edges to link each node to the centre, and 16 edges to link each node as a chain for the circle. What are the other 24 edges? Where does the ordering of nodes for the chain links come from? This can have a big affect on where the final layout is good, and ideally you would determine the ordering after some initial layout. I'd like to see this discussed.” – R4

## Clarifications:

* “In some place(s) the paper describes SetCoLa as a domain specific language, but it actually claims to be independent of the application domain. So I'm a bit confused. I think it's rather an application domain independent graph layout tool.” – R2
* “I didn't understand the last sentence in section 4.1: "The specified offsets enable changes to the alignment orientation of the nodes." Is "orientation" meant to refer to the aligned position (e.g., left, right, centre)? This could be made clearer.” – R4
* “The sentence at the end of paragraph 4 in section 5.3 that reads ", and the non-overlap constraint" might be clearer as ", and the generated non-overlap constraints."” – R4
* “The following sentence was hard to follow though: “When a named set is defined, the set element promotes the properties that are equivalent for all elements in the set to a property of the set element”” – R1

## Conflicting / Unsatisfiable Constraints:

* “The issue of conflicting or unsatisfied constraints is mentioned, but I have the feeling that this is even a much more important issue and should be addressed in more detail.” – R2
* “The paper clearly describes some of the limitations of the approach. My only suspicion is whether the constraints or additional edges to enforce structure **prevent the overall layout from untangling** (due to non-overlap constraints and the underlying force-directed layout model). This is discussed a little in the last paragraph of section 6.3, but this only really says it's a problem. I imagine this tangling could be quite bad if there were "free" nodes not part of position constraints with other nodes and which had a bad initial position. This can often be a problem if the chosen ordering for things to which constraints are applied to is not the optimal ordering. For SetCoLa the fact that the order constraint order is determined by the user mitigates this in one respect, but it also means the user can't distribute a set of nodes without specifying a specific order. I'd like to see a little more discussion of this. In terms of the untangling fighting non-overlap constraints and the idea of ordering the application of constraints, it could be worth mentioning that WebCoLa (via arguments to the "start" method) does allow specification of a number of unconstrained iterations, a number of interactions with user constraints (such as from SetCoLa) and then an number of iterations with all user and generated constraints (i.e., non-overlap). Though admittedly I don't understand the exact behaviour of this and I've only had limited success using it.” – R4

## Debugging:

* “Some approach that solves or clearly identifies most of the possible conflicts would be a great (and maybe needed?) addition.” – R2
* “I know by experience that constraint systems fail somewhat unpredictably and can raise frustration from users, so I would be surprised if SetCola were not surprising at times.” – R3

## Demonstration:

* “In the examples, like in Fig.4b) the used constraints allow to recreate a known layout example. But how easy is it to come up with **new interesting things**? I get it from the paper that many existing layouts can be recreated, and can be transported to other data, but I think the most likely use is to work with new data and then it's important to somehow easily find new and good layouts. This is not demonstrated.” – R2
* “The examples shown are interesting because they show how some existing layouts, usually done manually or by specialized tools, can be replicated more or less using SetCoLa. However, the article is not very clear about **examples that cannot be replicated** and why. There are two reasons why some layouts could be hard or impossible to replicate: either the expressive power of the system is not sufficient, or users are challenged to master some mechanisms in the system. In both cases, I would have been interested in having a few examples and explanations to get a better feel on where the system breaks.” – R3
* “The approach is evaluated by compression in code written versus the current method as well as demonstration that layouts from custom approaches can be replicated in this manner.” – R1
* “While proof of the **transferrability claims** of these scripts and the usability by intended audiences would strengthen the paper, it makes a significant step in providing a general way for more people to have access to customizable constrained layouts.” – R1
* “The paper evaluates SetCoLa by **demonstrating the ratio between** constraints written in SetCoLa and the number of WebCoLa constraints it then generates. It also demonstrates the ability to replicate existing biological and ecological network with relatively few lines of SetCoLa.” – R1
* “the claim that these scripts are **transferrable** is not supported” – R1
* “The paper claims that because constraints need no longer be written per node, the constraint scripts generated in SetCoLa are **transferrable** within a domain. This is not demonstrated within the paper. Writing the Limitations and Future Work in a more concise manner would provide more room to show the same script applied to multiple graphs from the same domain.” – R1
* “For each of the three examples, it is not clear if these were produced from a graph with no initial positions, or whether they were all beautifications from the starting positions of the original networks from the literature. From my own experience, I would guess it was the later. It would be good to have an explanation of this since whether it works well without **initial positions** can significantly affect it's usefulness.” – R4

## Usability:

* “Overall, I liked the paper, but while I am more or less convinced that the authors of the paper can use it at its full power, I am not totally convinced anyone could use it. Having some feedback from **real users** outside of the core designers/developers would be useful to understand who can specify high-level constraints, and what problems they bumped into.” – R3
* “While proof of the transferrability claims of these scripts and the **usability by intended audiences** would strengthen the paper, it makes a significant step in providing a general way for more people to have access to customizable constrained layouts.” – R1
* “remains unclear whether people outside the authors are effective at using these constraints” – R1