A Quantitative Comparison
of Stress-Minimization Approaches
for Offline Dynamic Graph Drawing[*]

Ulrik Brandes and Martin Mader

Department of Computer & Information Science, University of Konstanz
{Ulrik.Brandes,Martin.Mader}@uni-konstanz.de

Abstract. In dynamic graph drawing, the input is a sequence of graphs for which a sequence of layouts is to be generated such that the quality of individual layouts is balanced with layout stability over time. Qualitatively different extensions of drawing algorithms for static graphs to the dynamic case have been proposed, but little is known about their relative utility. We report on a quantitative study comparing the three prototypical extensions via their adaptation for the stress-minimization framework. While some findings are more subtle, the linking approach connecting consecutive instances of the same vertex is found to be the overall method of choice.

1   Introduction

A dynamic graph is a sequence of (static) graphs, often representing an evolving structure at discrete times of observation. Dynamic graph drawing refers to the problem of generating a sequence of layouts to be used either in a small multiples representation or as frames in an animation. In the offline scenario the entire input sequence is known in advance, whereas in the online scenario the sequence is given one graph at a time.

Approaches to dynamic graph drawing most often augment a layout algorithm designed for static graphs in such a way that the resulting sequence of layouts is more stable than if each graph was drawn from scratch [5]. The motivation for this approach is generally said to be the preservation of a viewer's mental map [15], but it may also be interpreted as conveying the degree and location of structural change more accurately by aligning it with layout change.

A common objective for drawing general undirected graphs is stress minimization [10,13], a special case of multidimensional scaling applied to graph-theoretic distances. It has been found to outperform other spring embedder variants [3] and will be the basis in this study.

The simplest (and most common) approach to add stability to an iterative layout algorithm for static graphs is to initialize the computation for each graph

in the sequence with the preceding layout [12,16]. The implicit assumption is that consecutive graphs are similar in general, and thus, the initial layout is not too far from a locally optimal one. The method is therefore easy to implement, more efficient than computing a layout from scratch, and applicable in both on- and offline scenarios.

However, stability is not addressed in a controlled way, hence this approach may result in excessive and unnecessary movement of vertices, and layout quality tends to degrade over the course of the sequence. Among the first to address stability directly were [1], and [17] provides a generic problem statement. The trade-off between readability and stability is formalized in [4] and a similar principle for offline scenarios is proposed in [7].

More sophisticated attempts to increase stability are typically based on one of three approaches. Maximum stability is achieved in aggregation approaches (e.g., [2,16]) where fixed vertex positions are obtained from the layout of an aggregate of all graphs in the sequence. Alternatives are based on anchoring vertices to reference positions (e.g., [4]), or linking vertices to instances of themselves that are close in the sequence (e.g., [8,9]).

Do these methods work well? Which one to implement for a given application? While the natural response to these questions appears to be a user study [18], their design may be challenging. Controlled experiments require a thorough understanding of the way in which model parameters affect outcomes. By purely algorithmic experimentation, we therefore want to provide quantitative evidence for the differential behavior of variant approaches, and thus prepare the ground for further user studies.

Our study compares aggregation, anchoring, and linking variants of stress minimization for offline dynamic graph drawing scenarios. The latter are of increasing relevance especially in longitudinal social network analysis [16], from which we hence draw some of our test cases. Our most important conclusion is that linking compares favorably with the other approaches.

After reviewing layout methods in Sect. 2, we formulate hypotheses in Sect. 3 that are based on common, though often implicit, assumptions about these methods and serve as a guideline for the experiments in Sect. 4. The experimental results are discussed in Sect. 4.3, and we conclude in Sect. 5.

2 Offline Dynamic Layout Approaches

Let $G = (V, E)$ be an undirected graph defined by a set V of n vertices, and a set E of m edges. An arbitrary pair of vertices is called *dyad*. Given a matrix D of vertex *dissimilarities* δ_{ij}, $i, j \in V$, the purpose of stress minimization is to determine positions $p_i = \langle x_i, y_i \rangle \in \mathbb{R}^2$ for every vertex $i \in V$ such that the Euclidean distances in the plane resemble the given dissimilarities as closely as possible, i.e., $\delta_{ij} \approx \|p_i - p_j\|$, where $\|\cdot\|$ denotes the Euclidean norm. For any given layout $P = (p_1, \ldots, p_n)$ this is quantified using a parameterized *stress function* stress(P),

$$\text{stress}(P) = \sum_{i<j} \omega_{ij} \left(\delta_{ij} - \|p_i - p_j\| \right)^2 \ , \tag{1}$$

where $W = (\omega_{ij})_{i,j \in V}$ is a weight matrix whose entries determine the contribution of each dyad. For graph drawing, lengths of shortest paths are a plausible choice for dissimilarities [10,13], and the objective is to find a layout of minimum stress. Because these distances are clearly not realizable for any non-trivial graph, weights $\omega_{ij} = \delta_{ij}^{-2}$ discount representation errors for distant pairs, thus emphasizing local accuracy.

Similar to other energy-based methods a solution can only be obtained by iterative stress reduction that yields a local minimum which may be far from an optimal layout. However, low-stress layouts can be routinely and efficiently computed using a two-step process [3]: In the first step, an initial layout is determined using *classical scaling*. In the second step, the representation of small distances is improved by iteratively and monotonically reducing stress using majorization [10].

2.1 Aggregation

Maximum stability is obtained when a vertex maintains its position throughout the entire sequence of diagrams. That is, given a sequence $G^{(1)} = (V, E^{(1)}), \ldots,$ $G^{(T)} = (V, E^{(T)})$ of T graphs with corresponding shortest-path distances $D^{(t)}$, $1 \le t \le T$, we are looking for one layout $\bar{P}$ for the vertices in V and let $P^{(t)} = \bar{P}$ at all times $t = 1, \ldots, T$.

We aggregate all shortest-path information by adapting input dissimilarities and weights in Eq. 1. We use $\bar{D} = (\bar{\delta}_{ij})_{i,j \in V}$, $\bar{\delta}_{ij} := \frac{1}{T} \sum_{t=1}^{T} \delta_{ij}^{(t)}$, i.e., the mean shortest-path distances, as dissimilarities, and weights $\bar{W} = (\bar{\omega}_{ij})_{i,j \in V}$ with

$$\bar{\omega}_{ij} = \frac{1}{\bar{\delta}_{ij}^2} \cdot \frac{1}{1 + \text{VAR}(\delta_{ij})} \ ,$$

where $\text{VAR}(\delta_{ij}) := \frac{1}{T} \sum_{t=1}^{T} \left(\delta_{ij}^{(t)} - \bar{\delta}_{ij} \right)^2$ is the variance of distances within a dyad across all observations. Thus, representation accuracy of dyads that are connected via short paths most of the time is emphasized. By additionally scaling with the variance, priority is given to structures that are relatively stable throughout the sequence. To obtain a layout we use the same algorithms as in the static case: Layout computation is initialized by classical scaling of mean distances; subsequently, $\text{stress}\left(\bar{P}\right)$ is reduced via majorization. Note that, in an offline scenario, infinite distances in a dyad that might occur due to temporary disconnectedness can be handled by interpolating between the two finite distances observed previously and next for this dyad, and by adding a small constant, say 1.

2.2 Anchoring

The main idea of the anchoring approach [4] is an explicit modeling of the trade-off between layout quality as measured by an objective function, and layout stability with respect to a reference drawing as measured by a difference metric [6]. A stress function quantifying the compromise between quality of each individual graph in the sequence and deviation from reference positions is

$$\mathsf{stress}_\alpha^A\left(P^{(t)}\right) = (1-\alpha) \cdot \underbrace{\mathsf{stress}\left(P^{(t)}\right)}_{\text{quality}} + \alpha \cdot \underbrace{\sum_{i \in V} \phi_i^{(t)} \left\| p_i^{(t)} - p_i \right\|^2}_{\text{stability}}, \qquad (2)$$

where $P = (p_i)_{i \in V}$ denotes the reference layout and weights $\phi_i^{(t)}$ allow for inter-vertex variation in deviation tolerance.

The stability term thus corresponds to a point-wise penalty for deviations from the reference layout, and the parameter $0 \le \alpha \le 1$ provides explicit control of the trade-off between quality (original stress) and stability. Note that minimizing stress_α^A for $\alpha = 0$ corresponds to regular stress minimization without control for stability, and $\alpha = 1$ yields the reference layout, since no deviation is tolerated.

For now, we use constant stability weights $\phi_i^{(t)} := 1$ for all i and t. More sophisticated choices, however, may be useful to compensate for cases with highly varying degrees or localized structural change. Before minimization of stress_α^A, we perform a Procrustes rotation [20] – an affine transformation that minimizes the sum of squared deviations from reference positions without changing relative distances – of the initial layout to the reference. After each layout of the sequence is obtained, we again apply Procrustes rotation subsequently to the whole sequence.

Depending on initialization and the type of reference, we obtain four anchoring methods. The first two are purely online, whereas the second two incorporate offline information by means of using the aggregate layout (Sect. 2.1) as reference:

APP initialize with previous layout (classical MDS for the first network), and also anchor to previous layout (no anchoring for the first network).
ACP initialize with classical scaling, anchor to previous layout (no anchoring for the first network).
APA initialize with previous layout (aggregate layout for the first network), anchor to aggregate layout.
ACA initialize with classical scaling, anchor to aggregate layout.

2.3 Linking

The main idea of the linking approach is to implicitly make use of all information about the networks of a sequence in an offline scenario. Instances of the same vertex are *linked* with each other, so as to stabilize their positions throughout

the sequence. In contrast to the anchoring approach, layout calculation is not performed one after each other, but the whole system is computed simultaneously.

A general formulation of a corresponding stress function is

$$\mathsf{stress}^L_\alpha \left(P^{(1)}, \ldots, P^{(T)} \right) =$$
$$(1-\alpha) \cdot \underbrace{\sum_{t=1}^{T} \mathsf{stress}\left(P^{(t)}\right)}_{\text{quality}} + \alpha \cdot \underbrace{\sum_{i \in V} \sum_{t'=1, t' \neq t}^{T} \phi_i^{(t)} \zeta(t,t') \left\| p_i^{(t)} - p_i^{(t')} \right\|^2}_{\text{stability}}, \quad (3)$$

where $\zeta(t,t')$ is a function controlling the influence of the position at a certain time t for vertices at other time points t'. Concretely, we implemented two versions w.r.t. $\zeta(t,t')$ similar to the two alternatives stated in [8]: $\zeta_G(t,t') = e^{-\frac{1}{2}(t'-t)^2}$, a Gaussian function with mean value t and variance 1 without normalization, i.e., $\zeta_G(t,t) = 1$; and $\zeta_W(t,t') = 1$ for $|t-t'| = 1$, and $\zeta_W(t,t') = 0$ otherwise, i.e., a vertex is only linked within a time-window of size 1. Again, we use $\phi_i^{(t)} = 1$, and align all layouts in the sequence by Procrustes rotation after initialization, and after stress minimization. Depending on initialization and $\zeta(t,t')$, we obtain four linking methods:

LCG initialization by classical scaling, use ζ_G.
LAG initialization by aggregate layout as described in Sect. 2.1, use ζ_G.
LCW initialization by classical scaling, use ζ_W.
LAW initialization by aggregate layout, use ζ_W.

3 Hypotheses

Explicitly addressing stability by use of the above methods instead of simply initializing with the preceding layout implies that a better compromise between quality and stability is expected. Assessment of this claim is broken down into constituent components to structure the discussion of detailed quantitative results in Sect. 4.3.

Our first hypothesis to test is thus that the methods actually display the assumed effects at all.

H 1. Aggregation, anchoring, and linking increase dynamic stability, but reduce individual quality.

Likewise, the explicit trade-off between quality and stability should be controllable via control parameter α.

H 2. In anchoring and linking, higher values of α result in more stability and less quality.

Being an iterative method, stress minimization is known to be susceptible to poor local minima and thus to depend on good initialization [3]. As a consequence, the same caveat should be in place where the outcome is not governed by the attempt to maintain stability.

H 3. For decreasing values of control parameter α, anchoring and linking are increasingly sensitive to initialization.

And finally, the principal adaptation to the offline scenario is by either anchoring to a reference position determined from the entire sequence of graphs, or by linking with future instances. These should pay off in case there is a persistent global structure.

H 4. For dynamic graphs with persistent structure, anchoring to an aggregate layout and linking outperform online approaches.

The experiments conducted in the next section are designed to provide evidence for assessing these rather qualitative associations in detail.

4 Experiments

Instead of illustrating the approaches on selected examples, we here perform algorithmic experiments to obtain more detailed and generalizable insight into the behavior of dynamic graph drawing approaches. It is thus particularly important to use realistic input graph sequences, but we also address the issue of quantifying the output in a novel way.

4.1 Data

As mentioned above, our focal application area are longitudinal social networks. Instead of using a (necessarily small) collection of benchmark networks, though, we generate random graphs that are believed to be realistic for the application scenario, because they are obtained from the two most prevalent models in this domain.

These are exponential-family random graph models (*ERGM*, [19], modeling the characteristics of single networks) to create the initial graph of each sequence, and stochastic actor-oriented models (*SAOM*, [21], modeling the evolution between two networks) to obtain the actual sequence.[1] Both models are based on network-specific characteristics, called *effects* – such as density of the network, reciprocity of edges for directed networks, or number of triangles – and associated model parameters determining whether an effect increases or decreases the probability of a network (ERGM), or of particular network changes (SAOM). Both allow for model estimation, given networks and the desired set of effects, and for simulation of networks, given a starting network and a specified model.

---

[1] Available for the open source statistical system R (packages `ergm` and `RSiena`).

A sequence of T graphs is created in the following way: Two actual observations G_1 and G_2 of a longitudinal network serve as the basis for the creation process. Using G_1, we estimate an ERGM using basic effects,[2] from which an artificial first observation G_1^{sim} is simulated. Next, a SAOM is estimated using the real observations G_1 and G_2.[3] The thus estimated SAOM is used for the following two simulations.[4] The artificial second observation G_T^{sim} is obtained by running a simulation using G_1^{sim} and G_2 as input. Then, a simulation using G_1^{sim} and G_T^{sim} as input is performed to obtain a reliable sequence of changes leading from G_1^{sim} to G_T^{sim}, which is partitioned into $T-1$ parts. Applying the corresponding changes to the initial observation G_1^{sim} yields a sequence of T networks.

As real input data, we use two data sets that are well studied in the social sciences.[5] The *s50* data set [14] comprises a sequence of three friendship networks of 50 female teenage pupils. We use the first and third observation as input for network sequence generation. The second real data set used is the *van de Bunt* data set [22], again, an evolving friendship network among 32 university freshmen comprising seven observations. We obtain input for network sequence generation by only considering edges with rating *best friendship* and *friendship* from the second and the seventh observation. Note also, that we removed vertex 18, since it is isolated at all time points.

In addition to the network generation process described above, we employ generation of unstructured artificial data by means of the $G(n,p)$ random graph model [11]. An initial observation is created with $n = 50$ and $p = \log(n)/n$, which produces connected graphs with high probability. Repeatedly, $k/2$ edges are formed uniformly at random and, likewise, $k/2$ edges deleted, where we do not allow deletion of edges just formed, and the resulting graph is made connected. In our experiments, we use $k = 2\sqrt{n}$ and $k = n$.

4.2 Measurements

To assess the quality and stability of layouts of a dynamic graph, we use the measures that constitute our approaches, that are, stress and sum of squared positional difference. Although it may be doubted whether these measures really capture either quality or stability, no other measures have been shown to better represent these concepts; it is therefore only reasonable to use the intrinsic measures of the approaches. Another problem is that both measures are not directly comparable across graphs of different sizes or structure. We solve this by relating both measures to the ones obtained from a common baseline method $\mathbf{B}$, that is, we compute static layouts for each graph in the sequence as suggested

---

[2] We use effects `edges`, `mutual`, `gwodegree` and `gwesp`.

[3] We use effects for the the number of changes (`rate parameter`), outdegree, reciprocity, and transitivity (`transitive triplets`).

[4] The first `RSiena` simulation uses the unconditional method of moments, the second uses the maximum likelihood method.
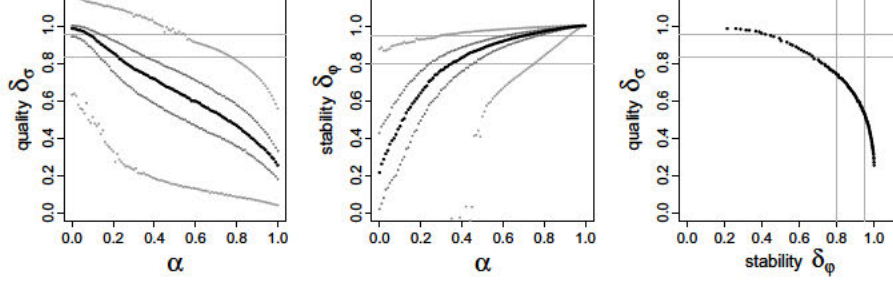
[5] Publicly available at `http://www.stats.ox.ac.uk/~snijders/siena/`

Fig. 1. Five-point summary for measurements of $\delta_\sigma^{\mathrm{APA}}$ and $\delta_\varphi^{\mathrm{APA}}$ subject to trade-off parameter α for 50 network sequences generated from the s50 data set

in the beginning of Sect. 2, and align the sequence by Procrustes rotation after layout calculation.

Let P_M be the layout for a graph obtained by method M. We measure quality δ_σ^M as the fraction of $\mathsf{stress}(P_B)$ and $\mathsf{stress}(P_M)$, i.e.,

$$\delta_\sigma^M = \frac{\mathsf{stress}(P_B)}{\mathsf{stress}(P_M)} \, .$$

Since we assume that the baseline layout is of relatively high quality, and that quality degrades when mechanisms to increase stability are employed, the range of δ_σ^M should be $[0,1)$, with δ_σ^M decreasing for higher values a.

Let $\varphi(P^{t-1}, P^t) = \sum_{i \in V} \left\| p_i^t - p_i^{t-1} \right\|^2$ be the sum of squared positional difference between two subsequent layouts P^{t-1} and P^t, where these have been aligned by Procrustes rotation, i.e., $\varphi(P^{t-1}, P^t)$ is minimal w.r.t. translation and rotation of P^{t-1} and P^t. Stability is measured as the relative decrease of positional difference w.r.t. the baseline layout, i.e.,

$$\delta_\varphi^M = 1 - \frac{\varphi(P_M^{t-1}, P_M^t)}{\varphi(P_B^{t-1}, P_B^t)} \, .$$

The assumption is that the baseline layout exhibits a high positional difference, that will decrease whenever mechanisms to increase stability are employed. Note that all methods presented yield the same layout for all graphs in the sequence for $\alpha = 1$, therefore δ_φ^M must be 1 for all methods in this case. Thus the range is expected to be $[0,1]$, with δ_φ^M increasing for higher values α.

For each network sequence generator, we created 50 network sequences comprising 10 graphs each. We measured δ_σ and δ_φ corresponding to trade-off parameter $\alpha \in \{0, 0.01, 0.02, \ldots, 1\}$. Thus, per generator, method, and value of α, we obtain 500 measurements of δ_σ (450 for **APP** and **ACP**, since for the first observation of each sequence in these cases $\delta_\sigma = 1$ for all α), and 450 measurements of δ_φ (not applicable to each first observation). Figure 1 shows a five point summary, i.e., minimum, first quartile, median, third quartile, and maximum,

Table 1. Median values for δ_σ and δ_φ at certain selected values of α for measurements on sequences generated from the s50 data set. Note that only measurements belonging to either the anchoring or linking approaches can be compared directly. Sequences from other generators reveal similar tendencies.

α :	0.1	0.2	0.3	0.7	0.8	0.9	0.1	0.2	0.3	0.7	0.8	0.9
APP	0.94	0.86	0.78	0.48	0.40	0.31	**0.49**	0.63	0.72	0.93	0.95	0.98
ACP	**0.97**	**0.90**	**0.80**	0.48	0.40	0.31	0.29	0.54	0.70	0.93	0.95	0.98
APA	0.94	0.86	0.77	**0.54**	**0.47**	0.37	0.46	**0.64**	**0.76**	**0.95**	**0.97**	**0.99**
ACA	0.96	0.87	0.79	0.53	**0.47**	**0.38**	0.22	0.52	0.72	**0.95**	**0.97**	**0.99**
LAG	0.96	0.91	0.86	0.68	0.62	0.54	0.51	0.69	0.78	**0.95**	0.97	0.99
LCG	**0.98**	**0.93**	**0.89**	**0.69**	**0.63**	0.54	0.35	0.59	0.72	0.94	0.97	0.99
LAW	0.95	0.90	0.85	0.68	0.62	**0.55**	**0.57**	**0.73**	**0.81**	**0.95**	0.97	0.99
LCW	0.97	0.92	0.87	0.68	**0.63**	**0.55**	0.43	0.64	0.75	0.94	0.97	0.99
			$\hat{\delta}_\sigma$						$\hat{\delta}_\varphi$			

of the measurements obtained for the 50 network sequences generated from the s50 data set when applying the **APA** method. The gray horizontal lines indicate thresholds used in our experiments, that are, 5% and 20% more stress w.r.t. the baseline for quality measurements, and 80% and 95% reduction in positional difference w.r.t. to the baseline for stability measurements. Note that, for both δ_σ and δ_φ, there are outliers that contradict the intuitive assumptions regarding the range of the measures. We can only explain these by the heuristic nature of stress minimization. Still, most of the measured values are within a reasonable range around the median values, as can be observed by the inter-quartile range. Thus, we will argue about the approaches by means of the median measurements, denoted by $\hat{\delta}_\sigma$ and $\hat{\delta}_\varphi$, respectively. Table 1 shows values of $\hat{\delta}_\sigma$ and $\hat{\delta}_\varphi$ at selected levels of α for network sequences generated from the s50 data set, and Fig. 2 summarizes measurements for all methods and data sets.

4.3 Results

Figure 2 (upper row, right endpoints) shows that already a slight compromise in quality (5% additional stress compared to static baseline layouts) yields a large increase in positional stability (ranging from 24% to 82% reduction of total movement). If we allow a 20% increase of stress (left-hand side of each upper graph), all methods reduce movement by more than 50%. Across all experiments, reduction in positional difference increases very rapidly at lower ranges of α as exemplified in Fig. 1. This provides evidence that the methods are largely having the desired effects (H 1). Moreover, the actual values corroborate earlier findings that low stability mechanisms appear to be most effective [18].

The monotone behavior of median values of δ_σ and δ_φ in Table 1 support the expected dependencies on α (H 2). Also, sensitivity to initialization (H 3) is confirmed: Although small, there are noticeable differences in quality at the lower range ($\alpha \leq 0.3$) in favor of initialization with classical scaling. There are, however, large differences in stability in favor of initialization with the previous
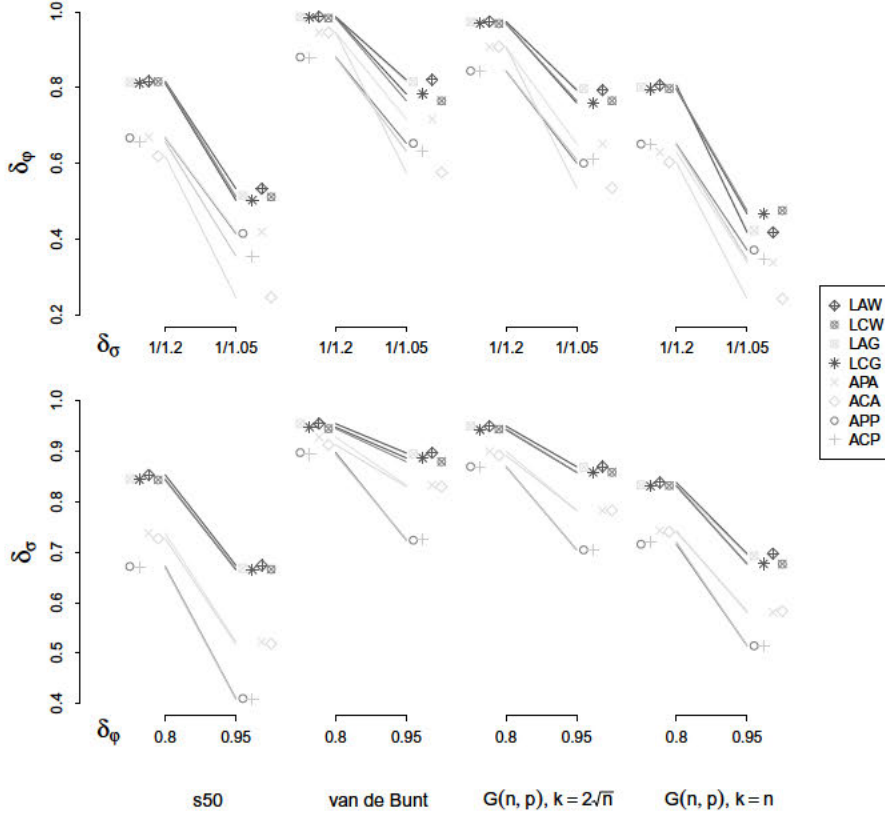
Fig. 2. Overview of median values for δ_φ at selected median values of δ_σ (upper part) and vice versa (lower part) for all measurements

layout for anchoring approaches, or the aggregate layout for linking approaches, as expected. At higher range, we observe very similar results for each pair of anchoring approaches belonging together w.r.t. the choice of reference, and generally similar results for the linking approaches, which is also visible in the lower graphs of Fig. 2. This indicates diminishing influence of initialization at higher stability requirements.

Regarding H 4, we first posit that data generated from ERGMs and SAOM learned from real data contains more persistent structures, since change in these networks is actually based on structural processes unlike the random changes in the $G(n, p)$ generator. Rather surprisingly, though, we cannot observe a general trend in Fig. 2 that **A?A** approaches perform better than **A?P** approaches at very low stability levels (stress increase of 5%). Indeed method **ACA** consistently performs worse, which indicates that initialization with classical scaling is in conflict with anchoring to the aggregate. However, the more stability is sought, the more is the difference in increase of stress between the **A?A** and

the **A?P** approaches, in favor of the former, and regardless of the data set. The linking approaches perform better than the anchoring approaches throughout. Thus, our evaluation of H 4 is inconclusive, since no statement can be made for very low stability, and methods incorporating offline information apparently perform better regardless of structure for moderate, and especially, high stability requirements. To our surprise, the choice between the two functions ζ_G and ζ_W does not seem to considerably influence the results.

5 Conclusion

We compared dynamic variants of the stress-minimization approach for general undirected graphs in which vertices are at the same position throughout the sequence of layouts (aggregation), attracted by a reference position (anchoring), or attracted by positions of their copies in neighboring time slices (linking).

The comparison was based on a novel form of measurement of the trade-off between quality (in terms of stress) and stability (in terms of vertex movement): Measures were related to a baseline, determined from Procrustes aligned static layouts, to normalize over graphs of different sizes and structure.

A second novel aspect is our use of more sophisticated graph generators that eliminate reliance on small benchmark data sets and still produce application-typical data. Here, ERGMs were used for boundary observations and SAOMs for the evolution.

Our results suggest that linking is a generally preferable approach. Since it is computationally demanding, a faster alternative is anchoring to an aggregate layout initialized with the previous one in the sequence.

The present study is an attempt to move towards more precise measurement of those aspects of the performance of graph drawing algorithms that are not easily characterized analytically, but we are left with more new questions than answers to old ones. The focus on stress-minimization approaches and two particular criteria for quality and stability allowed for better comparison and more detailed insights, but different quantities may turn out important as well. Other avenues for future research include refined data generation procedures (e.g., including behavioral effects), in-depth discussion of outliers and other observations, and dependencies on specific graph structures and change sequences.

References

1. Böhringer, K.F., Paulisch, F.N.: Using constraints to achieve stability in automatic graph layout algorithms. In: Proc. of the SIGCHI Conference on Human Factors in Computing Systems (CHI 1990), pp. 43–51. ACM (1990)
2. Brandes, U., Corman, S.R.: Visual unrolling of network evolution and the analysis of dynamic discourse. Information Visualization 2(1), 40–50 (2003)
3. Brandes, U., Pich, C.: An Experimental Study on Distance-Based Graph Drawing. In: Tollis, I.G., Patrignani, M. (eds.) GD 2008. LNCS, vol. 5417, pp. 218–229. Springer, Heidelberg (2009)

4. Brandes, U., Wagner, D.: A Bayesian Paradigm for Dynamic Graph Layout. In: DiBattista, G. (ed.) GD 1997. LNCS, vol. 1353, pp. 236–247. Springer, Heidelberg (1997)

5. Branke, J.: Dynamic Graph Drawing. In: Kaufmann, M., Wagner, D. (eds.) Drawing Graphs. LNCS, vol. 2025, pp. 228–246. Springer, Heidelberg (2001)

6. Bridgeman, S.S., Tamassia, R.: Difference metrics for interactive orthogonal graph drawing algorithms. Journal of Graph Algorithms and Applications 4(3), 47–74 (2000)

7. Diehl, S., Görg, C.: Graphs, they are Changing. In: Goodrich, M.T., Kobourov, S.G. (eds.) GD 2002. LNCS, vol. 2528, pp. 23–30. Springer, Heidelberg (2002)

8. Erten, C., Harding, P., Kobourov, S., Wampler, K., Yee, G.: Graphael: Graph Animations with Evolving Layouts. In: Liotta, G. (ed.) GD 2003. LNCS, vol. 2912, pp. 98–110. Springer, Heidelberg (2004)

9. Erten, C., Kobourov, S., Le, V., Navabi, A.: Simultaneous graph drawing: Layout algorithms and visualization schemes. Journal of Graph Algorithms and Applications 9(1), 165–182 (2005)

10. Gansner, E., Koren, Y., North, S.: Graph Drawing by Stress Majorization. In: Pach, J. (ed.) GD 2004. LNCS, vol. 3383, pp. 239–250. Springer, Heidelberg (2005)

11. Gilbert, E.N.: Random graphs. The Annals of Mathematical Statistics 30(4), 1141–1144 (1959)

12. Huang, M.L., Eades, P., Wang, J.: On-line animated visualization of huge graphs using a modified spring algorithm. Journal of Visual Languages and Computing 9(6), 623–645 (1998)

13. Kamada, T., Kawai, S.: An algorithm for drawing general undirected graphs. Information Processing Letters 31, 7–15 (1989)

14. Michell, L., Amos, A.: Girls, pecking order and smoking. Social Science & Medicine 44(12), 1861–1869 (1997)

15. Misue, K., Eades, P., Lai, W., Sugiyama, K.: Layout adjustment and the mental map. Journal on Visual Languages and Computing 6(2), 183–210 (1995)

16. Moody, J., McFarland, D.A., Bender-deMoll, S.: Dynamic Network Visualization. American Journal of Sociology 110(4), 1206–1241 (2005)

17. North, S.C.: Incremental Layout with DynaDag. In: Brandenburg, F.J. (ed.) GD 1995. LNCS, vol. 1027, pp. 409–418. Springer, Heidelberg (1996)

18. Purchase, H.C., Samra, A.: Extremes are Better: Investigating Mental Map Preservation in Dynamic Graphs. In: Stapleton, G., Howse, J., Lee, J. (eds.) Diagrams 2008. LNCS (LNAI), vol. 5223, pp. 60–73. Springer, Heidelberg (2008)

19. Robins, G., Pattison, P., Kalish, Y., Lusher, D.: An introduction to exponential random graph (p*) models for social networks. social networks 29(2), 173–191 (2007)

20. Sibson, R.: Studies in the robustness of multidimensional scaling: Procrustes statistics. Journal of the Royal Statistical Society. Series B (Methodological) 40(2), 234–238 (1978)

21. Snijders, T.A.B.: The statistical evaluation of social network dynamics. Sociological Methodology 31, 361–395 (2001)

22. Van De Bunt, G.G., Van Duijn, M.A., Snijders, T.A.: Friendship networks through time: An actor-oriented dynamic statistical network model. Computational & Mathematical Organization Theory 5, 167–192 (1999)