Stochastic Gradient Descent Works Really Well for Stress Minimization

Katharina Börsig, Ulrik Brandes(✉), and Barna Pasztor

ETH Zürich, Zürich, Switzerland
ubrandes@ethz.ch

Abstract. Stress minimization is among the best studied force-directed graph layout methods because it reliably yields high-quality layouts. It thus comes as a surprise that a novel approach based on stochastic gradient descent (Zheng, Pawar and Goodman, TVCG 2019) is claimed to improve on state-of-the-art approaches based on majorization. We present experimental evidence that the new approach does not actually yield better layouts, but that it is still to be preferred because it is simpler and robust against poor initialization.

Keywords: Multidimensional scaling · Stress minimization · Stochastic gradient descent · Experiments.

1 Introduction

The class of force-directed graph drawing algorithms is large both in terms of objectives and optimization algorithms [1,13]. Experimental [4] and anecdotal evidence suggest that a most desirable objective is the stress function of distance-based multidimensional scaling [14]. Given a simple undirected graph $G = (V, E)$, the layout $x = (\mathbb{R}^2)^V$ of a straight-line drawing is considered suitable, if the weighted deviation

$$stress(x) = \sum_{i<j} d_{ij}^{-2}(\|x_i - x_j\| - d_{ij})^2 \tag{1}$$

of Euclidean distances $\|x_i - x_j\|$ in the layout from shortest-path distances d_{ij} in the graph is small.

The stress function has been varied in numerous ways to accommodate additional objectives or constraints [2,5,8,9,16]. Since stress minimization is computationally intractable, similarly many approaches have been proposed to save computation time [11,15,17]. These methods are generally designed to improve an initial layout iteratively and thus yield local minima of the stress function that cannot be improved further by moving single vertices.

Here we are interested in assessing a recent proposal by Zheng, Pawar, and Goodman [18] that is based on stochastic gradient descent and claimed to outperform majorization approaches [10].

D. Auber and P. Valtr (Eds.): GD 2020, LNCS 12590, pp. 18–25, 2020.
https://doi.org/10.1007/978-3-030-68766-3_2

Our own computational experiments suggest that the new approach does not lead to better layouts, but that it is still preferable due to its simplicity and, crucially, indifference to initialization. We do not address actual running times because any comparison would be relative to the choice of speed-up techniques and the overall similarity of the computation suggests that the same algorithm engineering techniques could be used in either approach.

The remainder is organized as follows. In Sect. 2, we briefly describe the proposal of Zheng et al. in the context of previous approaches. The results of our experiments are presented and discussed in Sect. 3, and we conclude with some general implications in Sect. 4.

2 Stress Minimization

We very briefly review some major developments in the use of multidimensional scaling in graph drawing. This is not to provide the details of each method but to contrast the approach based on stochastic gradient descent with previous approaches.

Gradient Descent. While first uses of multidimensional scaling for graph drawing date back to the 1960s, it was popularized by Kamada and Kawai [12], who also introduced a localized version of the gradient descent approach used until then. Since a necessary condition for a local minimum of the stress function is that all partial derivatives are zero, they iteratively pick a vertex for which the vector of partial derivatives with respect to its two coordinates has maximum length. Then a two-dimensional Newton-Raphson method is applied to the stress function with all other vertices fixed. Their layout is thus obtained by iteratively moving one vertex at a time toward a position where the different stress terms cancel each other out.

Majorization. Ganser, Koren, and North [10] proposed to use majorization [7] instead. Here, the complex stress function is replaced with a convex function that is larger for each layout but the current, for which it is equal. Minimizing this function leads to a new layout that is guaranteed to have lower stress, and the process is iterated until it converges to a local minimum.

The process can also be localized to move only a single vertex such that the majorizing function is reduced. This yields an intuitive algorithm because the update

$$x_i \leftarrow \frac{1}{\sum\limits_{j \neq i} d_{ij}^{-2}} \sum_{j \neq i} d_{ij}^{-2} \cdot \frac{x_j + d_{ij}(x_i - x_j)}{\|x_i - x_j\|}$$

places vertex i directly into a position that balances out the influences of all other vertices. One iteration consists of an update of each vertex.

Because of its simplicity and guaranteed convergence, this approach is considered the state of the art.

Stochastic Dradient Descent. In this method the gradient is replaced by an unbiased estimator. For additive objective functions such as the stress function in Eq. (1), the estimator may simply be a single term of the sum. Since stress has one term for every pair of vertices, the contribution of this term can be reduced by moving the two vertices either closer together or farther apart.

A single update thus moves both vertices along the vector δ to extend or shrink the line segment $\overline{x_i x_j}$ to match the target length d_{ij} more closely,

$$\begin{aligned} x_i &\leftarrow x_i - \tfrac{\mu(t)}{2} \cdot \delta \\ x_j &\leftarrow x_j + \tfrac{\mu(t)}{2} \cdot \delta \end{aligned} \qquad \text{where} \qquad \delta = \frac{\|x_i - x_j\| - d_{ij}}{\|x_i - x_j\|} \cdot (x_i - x_j),$$

and $\mu(t) = \min\{1, d_{ij}^{-2}\eta(t)\}$ is a weighted step width capped at 1. Since an individual move is almost certainly in conflict with the desired distances of other pairs, the method does not converge in general. Instead, the unweighted step width $\eta(t)$ is made to exhibit an exponential decay over iteration time t, and convergence is thus enforced.

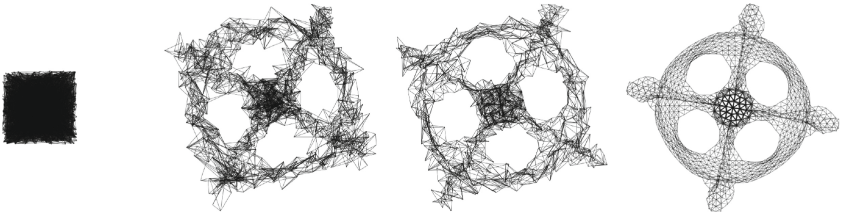

Fig. 1. Example run of stochastic gradient descent on graph `dwt_1005` with random initialization and intermediate layouts after 1, 6, and 15 iteration.

One iteration consists of an update of all pairs of vertices in random order. The method is thus similar to localized majorization but instead of aggregating the influence of all other vertices before moving one, those influences are considered separately in random order. The running time of one iteration is in $\Theta(n^2)$ for both stochastic gradient descent and localized majorization, but instead of over a linear number of linear-time vertex movements the computation is spread out over a quadratic number of constant-time dyadic updates.

3 Experiments

Our experiments address the claim [18] that stochastic gradient descent (SGD) outperforms majorization (SMACOF). The graphs used as benchmarks are from the University of Florida sparse matrix collection [6].

On Par, But Not Better. The claim of superior performance is based on experiments in which both approaches are initialized with a random layout as in the example in Fig. 1. It was already concluded from earlier experiments, however, that the performance of majorization depends on the initialization and that random initialization leads to poor local minima [4].

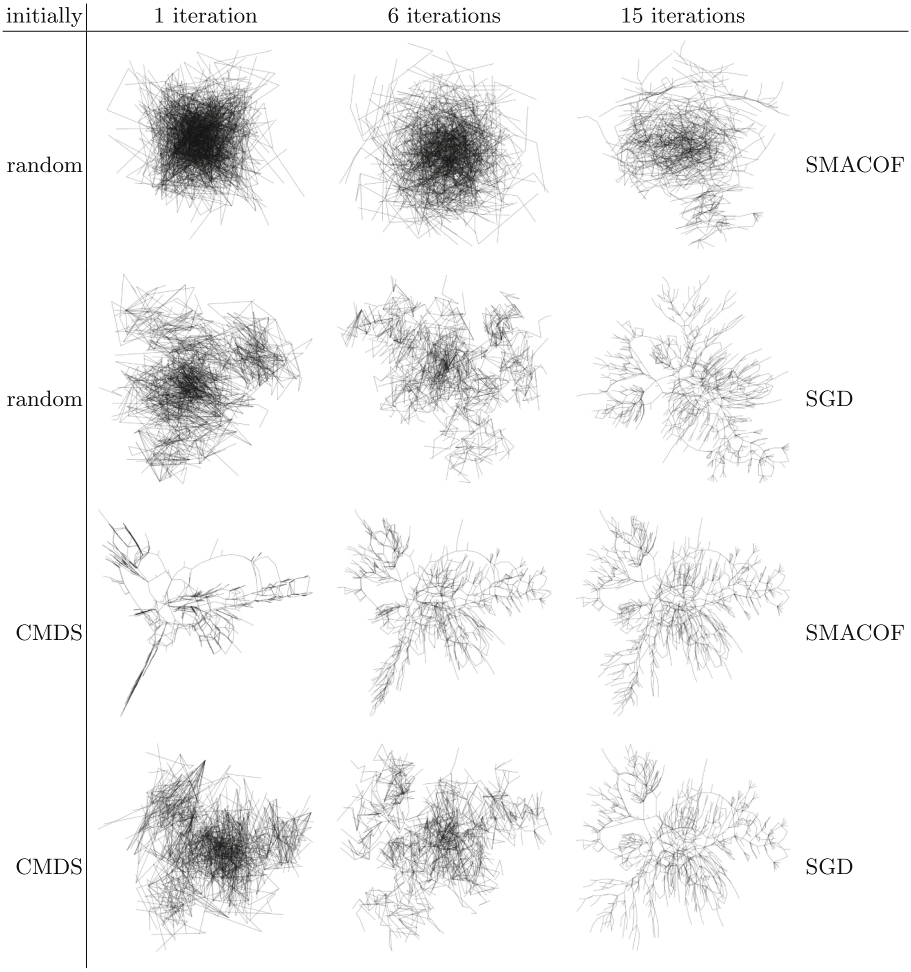

Fig. 2. An example graph (`1138_bus`) after 1, 6, and 15 iterations.

We therefore ran experiments comparing the reduction in stress when initializing at random or with classical MDS (CMDS). Classical MDS results in layouts that are essentially unique and represent large distances well. Moreover, it can be approximated at comparatively negligible cost using PivotMDS [3]. Two typical examples of the results are shown in Fig. 3, and for a better intuition, we also

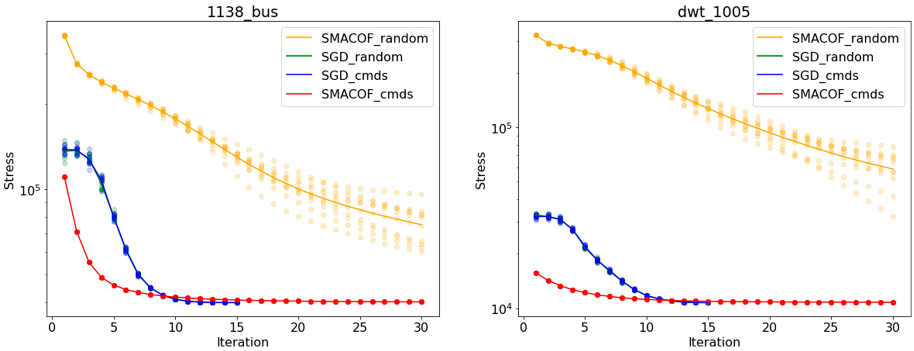

Fig. 3. Stress values for SGD and SMACOF on two example graphs. Random initialization is within a unit square whilst classical MDS is used at an appropriate scale. The plots show results of 10 runs for each algorithm, with circles representing single runs and lines interpolating through the means of all 10 runs. Initial stress omitted.

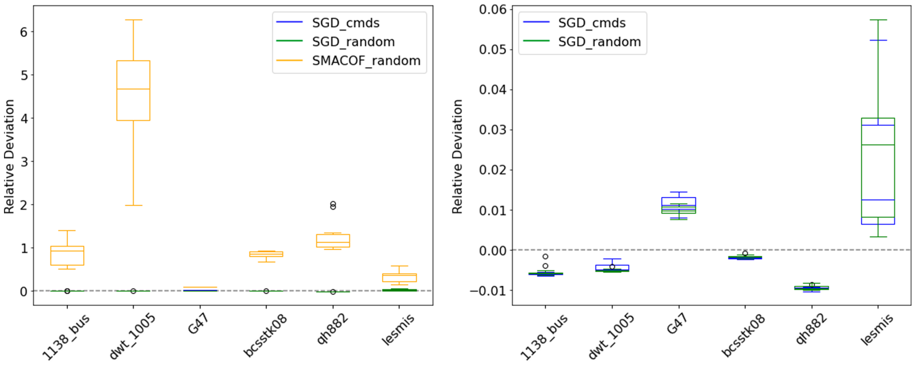

Fig. 4. Stress relative to baseline from SMACOF after CMDS. With 10 runs for each instance, we find that random initialization results in significantly higher stress for SMACOF (left chart). The stress obtained from SGD differs by about ±1% (rescaled on the right).

show some of the corresponding layouts in Fig. 2. While the result on all benchmark graphs confirm that SGD indeed yields much lower stress than majorization when initialized with a random layout, there is no noteworthy difference in the final stress when the initial layout takes care of the global arrangement. Notably, the result of SGD is largely independent of the initialization strategy.

Our experiments on a much larger set of benchmark graphs support these conclusions. The evaluations in Fig. 4 confirm quantitatively that majorization with random initialization is a poor baseline because it results in significantly higher stress compared to majorization after classical scaling. Whether SGD or the latter combination yield lower stress depends on the graph, but relative differences are small, anyway.

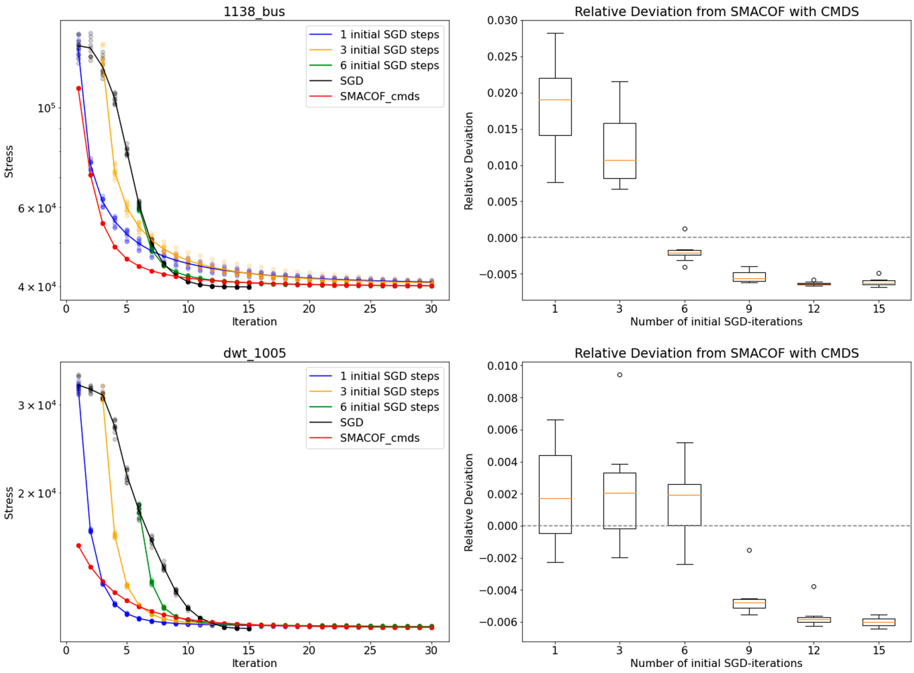

Fig. 5. Stress reduction by SMACOF initialized with CMDS or a few iterations of SGD (left) and the relative deviation of the final stress from the baseline of SMACOF with CMDS (right) on example graphs `1138_bus` and `dwt_1005`. The initial iterations of SGD start from a random initialization in the unit square, and each instance was run 10 times.

Self-initializing. The seeming indifference of SGD to the initial layout prompted a second suite of experiments.

We hypothesized that the initially large displacements in SGD are responsible for the overall quality of the final outcome. If this was the case, then the differences between SGD and SMACOF should disappear when we initialize SMACOF with a small number of SGD iterations.

As illustrated in Fig. 5 this is indeed the case. Even a single step of SGD prevents majorization from sinking into a poor local minimum. After about seven iterations of SGD, majorization yields layouts that are even slightly better than those obtained from initialization with CMDS. We also note that in the next iterations, SMACOF reduces stress faster than SGD, but the number of iterations to the final layout is roughly the same for both. This number becomes smaller than for SMACOF initialized with CMDS, offsetting the higher cost of SGD iterations compared to PivotMDS.

We conclude that a, if not the, major advantage of the approach based on stochastic gradient descent is the reliable untangling of any initial layout during the first few iterations. No separate initialization strategy is required.

Well Designed. We performed a number of additional experiments that generally confirm the recommendations given for stochastic gradient descent [18], and indicate that little can be gained by straightforward attempts at improvement such as an initial focus on long distances or the integration of majorization steps.

4 Conclusions

We have presented computational experiments comparing two approaches for graph drawing by multidimensional scaling of shortest-path distances.

Contrary to claims by the authors, we do not find that stochastic gradient descent, which was recently proposed as an alternative to majorization, leads to better layouts [18]. We find no significant differences in stress, provided that majorization is initialized appropriately.

The true advantage of stochastic gradient descent appears to lie in its indifference to initialization. It is striking that this very simple and uniform algorithm yields results that are on par with the state of the art.

We did not compare running times in this short paper because both approaches largely perform the same operations in different order and speed-up techniques such as subsampling and spatial aggregation abound. Since many of these apply similarly to both approaches, we expect differences to be too subtle for any general claims. Since pairs in a maximal matching can be updated without interference, stochastic gradient descent appears to be more amenable to parallelization, though.

References

1. Brandes, U.: Force-directed graph drawing. In: Kao, M.Y. (ed.) Encyclopedia of Algorithms, pp. 1–6. Springer, New York (2014). https://doi.org/10.1007/978-3-642-27848-8_648-1
2. Brandes, U., Mader, M.: A quantitative comparison of stress-minimization approaches for offline dynamic graph drawing. In: van Kreveld, M., Speckmann, B. (eds.) GD 2011. LNCS, vol. 7034, pp. 99–110. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-25878-7_11
3. Brandes, U., Pich, C.: Eigensolver methods for progressive multidimensional scaling of large data. In: Kaufmann, M., Wagner, D. (eds.) GD 2006. LNCS, vol. 4372, pp. 42–53. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-70904-6_6
4. 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). https://doi.org/10.1007/978-3-642-00219-9_21
5. Brandes, U., Pich, C.: More flexible radial layout. J. Graph Alg. Appl. **15**(1), 157–173 (2011). https://doi.org/10.7155/jgaa.00221
6. Davis, T.A., Hu, Y.: The University of Florida sparse matrix collection. ACM Trans. Math. Softw. **38**(1), 1 (2011). https://doi.org/10.1145/2049662.2049663
7. De Leeuw, J.: Applications of convex analysis to multidimensional scaling. In: Barra, J.R., Brodeau, F., Romier, G., Van Cutsem, B. (eds.) Recent Developments in Statistics, pp. 133–145. North Holland Publishing Company (1977). http://www.stat.ucla.edu/~deleeuw/janspubs/1977/chapters/deleeuw_C_77.pdf

8. Dwyer, T., Koren, Y., Marriott, K.: Constrained graph layout by stress majorization and gradient projection. Discrete Math. **309**(7), 1895–1908 (2009). https://doi.org/10.1016/j.disc.2007.12.103
9. Gansner, E.R., Hu, Y., North, S.C.: A maxent-stress model for graph layout. IEEE Trans. Vis. Comput. Graph. **19**(6), 927–940 (2013). https://doi.org/10.1109/TVCG.2012.299
10. Gansner, E.R., Koren, Y., North, S.: Graph drawing by stress majorization. In: Pach, J. (ed.) GD 2004. LNCS, vol. 3383, pp. 239–250. Springer, Heidelberg (2005). https://doi.org/10.1007/978-3-540-31843-9_25
11. Ingram, S., Munzner, T., Olano, M.: Glimmer: multilevel MDS on the GPU. IEEE Trans. Vis. Comput. Graph. **15**(2), 249–261 (2009). https://doi.org/10.1109/TVCG.2008.85
12. Kamada, T., Kawai, S.: An algorithm for drawing general undirected graphs. Inf. Process. Lett. **31**, 7–15 (1989). https://doi.org/10.1016/0020-0190(89)90102-6
13. Kobourov, S.G.: Force-directed drawing algorithms. In: Tamassia, R. (ed.) Handbook of Graph Drawing and Visualization, pp. 383–408. CRC Press, Oxford (2013)
14. McGee, V.E.: The multidimensional scaling of 'elastic' distances. Br. J. Math. Stat. Psychol. **19**(2), 181–196 (1966). https://doi.org/10.1111/j.2044-8317.1966.tb00367.x
15. Meyerhenke, H., Nöllenburg, M., Schulz, C.: Drawing large graphs by multilevel maxent-stress optimization. IEEE Trans. Visual Comput. Graphics **24**(5), 1814–1827 (2018). https://doi.org/10.1109/TVCG.2017.2689016
16. Nocaj, A., Ortmann, M., Brandes, U.: Untangling the hairballs of multi-centered, small-world online social media networks. J. Graph Alg. Appl. **19**(2), 595–618 (2015). https://doi.org/10.7155/jgaa.00370
17. Ortmann, M., Klimenta, M., Brandes, U.: A sparse stress model. J. Graph Alg. Appl. **21**(5), 791–821 (2017). https://doi.org/10.7155/jgaa.00440
18. Zheng, J.X., Pawar, S., Goodman, D.F.M.: Graph drawing by stochastic gradient descent. IEEE Trans. Visual Comput. Graphics **25**(9), 2738–2748 (2019). https://doi.org/10.1109/TVCG.2018.2859997