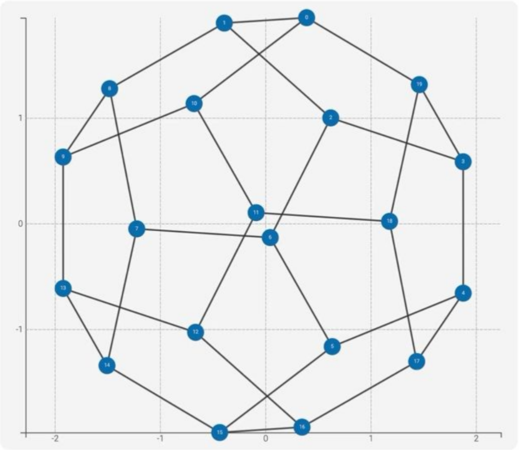
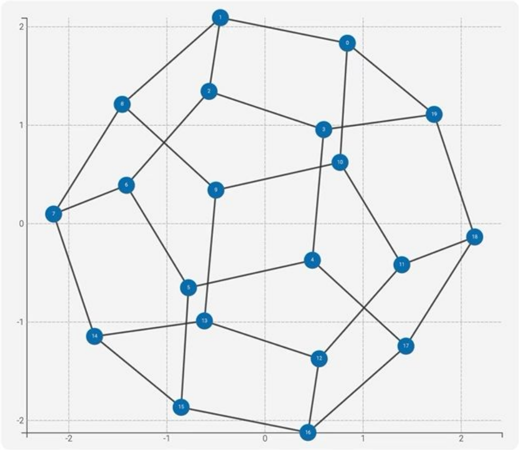




Multicriteria Scalable Graph Drawing via Stochastic Gradient Descent, $(SGD)^2$

Reyan Ahmed, Felice De Luca, Sabin Devkota, Stephen Kobourov, Mingwei Li
University of Arizona

The Humans, Data, and Computers Lab
hdc.cs.arizona.edu

THE UNIVERSITY OF ARIZONA

- Introduction
- Related Work
- The $(GD)^2$ Framework
- Properties and Measures
- Conclusions

Drawing Properties

This talk:

- Stress
- Neighborhood Preservation

More in the paper:

- Vertex Resolution
- Ideal Edge Length
- Crossing Number
- Crossing Angle Maximization
- Angular Resolution
- Aspect Ratio
- Gabriel Graph Property

Stress

- Quality measure & loss function:

$$L_{ST} = \sum_{i<j} w_{ij}(|X_i - X_j|_2 - d_{ij})^2$$

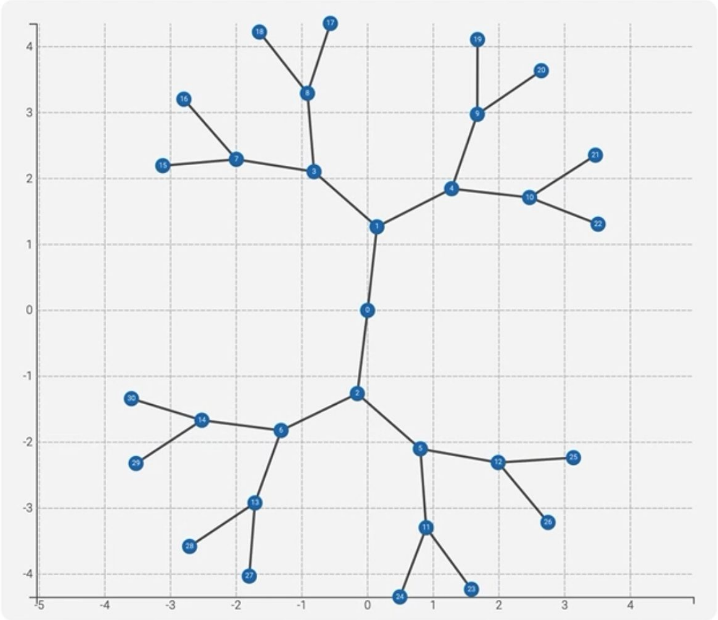
- w_{ij} - Normalizing term, we take $w_{ij} = d_{ij}^{-2}$

d_{ij} - Graph theoretical distance

X_i - coordinate of ith node in the layout

- Observation:

L_{ST} is a <u>differentiable</u> function of the layout X



Kamada and Kawai: An algorithm for drawing general undirected graphs. Information Processing Letters 31(1), 7 – 15 (1989)

Stress

- Observation:

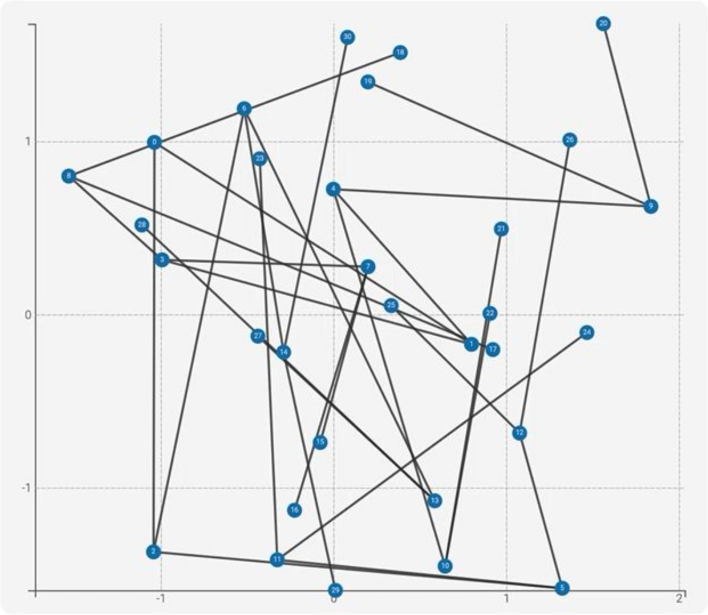
L_{ST} is a <u>differentiable</u> function of the layout X

=> Optimizable via gradient descent

```
for k in range(max_iter):
```

$$L_{ST} = \sum_{i<j} w_{ij}(|X_i - X_j|_2 - d_{ij})^2$$

$$X = X - \epsilon \cdot \nabla_X L_{ST}$$

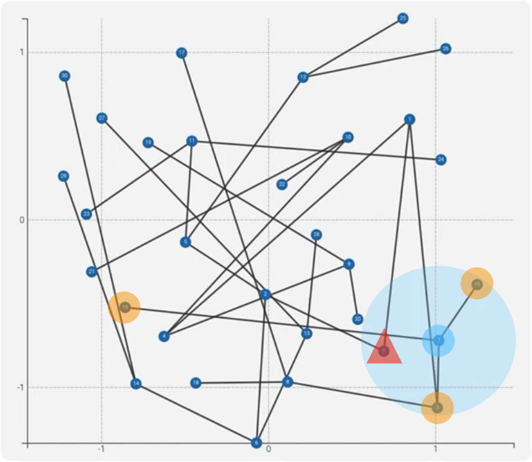


In general:

- - We find a loss function that is differentiable with respect to the layout X.
  - If the original criterion is *not* differentiable, we find a surrogate function.

Neighborhood Preservation

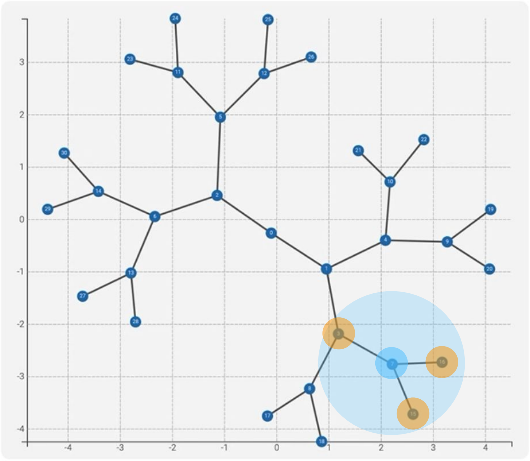
(Intuition)



Bad Neighborhood Preservation

Neighbors in layout ≠ Neighbors in graph

{ ▲, ●, ● } ≠ { ●, ●, ● }



Good Neighborhood Preservation

Neighbors in layout = Neighbors in graph

{ ●, ●, ● } = { ●, ●, ● }

Neighborhood Preservation

(Goal & Quality measure)

- Quality Measure (the higher, the better):

Q_{NP} := Jaccard Index (K, Adj)

$= \frac{|\{(i,j): K_{ij}=1 \, and \, Adj_{ij}=1\}|}{|\{(i,j): K_{ij}=1 \, or \, Adj_{ij}=1\}|}$

K - k-NN Matrix

k_i - (Chosen to be) the degree of node i

Adj - Adjacency Matrix

- Goal:

k-NN Matrix (K) ≈ Adjacency Matrix (Adj)

Kruiger et al.: Graph layouts by t-SNE. Comput. Graph. Forum36(3), 283–294 (2017)

Neighborhood Preservation

(Goal & Quality measure)

- Quality Measure (the higher, the better):

Q_{NP} := Jaccard Index (K, Adj)

$$= \frac{|\{(i,j): K_{ij}=1 \; and \; Adj_{ij}=1\}|}{|\{(i,j): K_{ij}=1 \; or \; Adj_{ij}=1\}|}$$

K - k-NN Matrix

k_i - (Chosen to be) the degree of node i

Adj - Adjacency Matrix

- Goal:

k-NN Matrix (K) ≈ Adjacency Matrix (Adj)

Kruiger et al.: Graph layouts by t-SNE. Comput. Graph. Forum36(3), 283–294 (2017)

Neighborhood Preservation

(Goal & Quality measure)

- Goal / Intuition

k-NN Matrix (K) ≈ Adjacency Matrix (Adj)

- Quality Measure (the higher, the better):

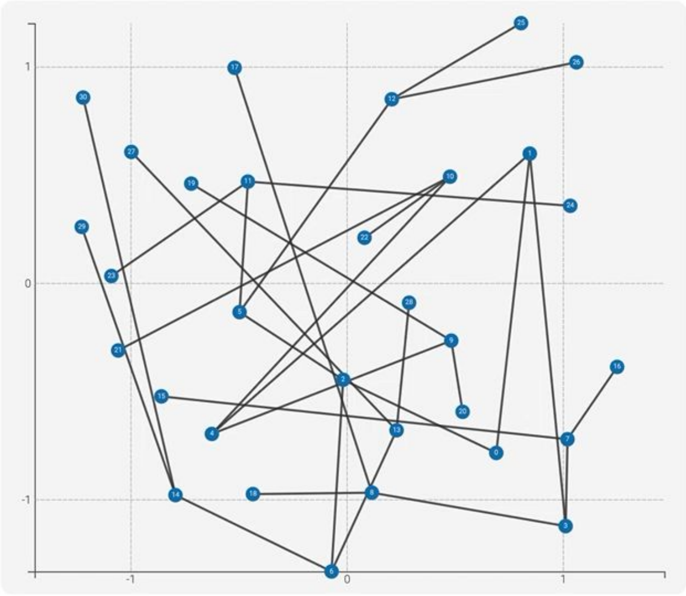
Q_{NP} := Jaccard Index (K, Adj) $= \frac{|\{(i,j): K_{ij}=1 \, and \, Adj_{ij}=1\}|}{|\{(i,j): K_{ij}=1 \, or \, Adj_{ij}=1\}|}$

- Notation:

k_i - chosen to be the degree of node i

Adj - ground truth, constant

K - layout-dependent, but *not* (yet) differentiable



Kruiger et al.: Graph layouts by t-SNE. Comput. Graph. Forum36(3), 283–294 (2017)

Neighborhood Preservation

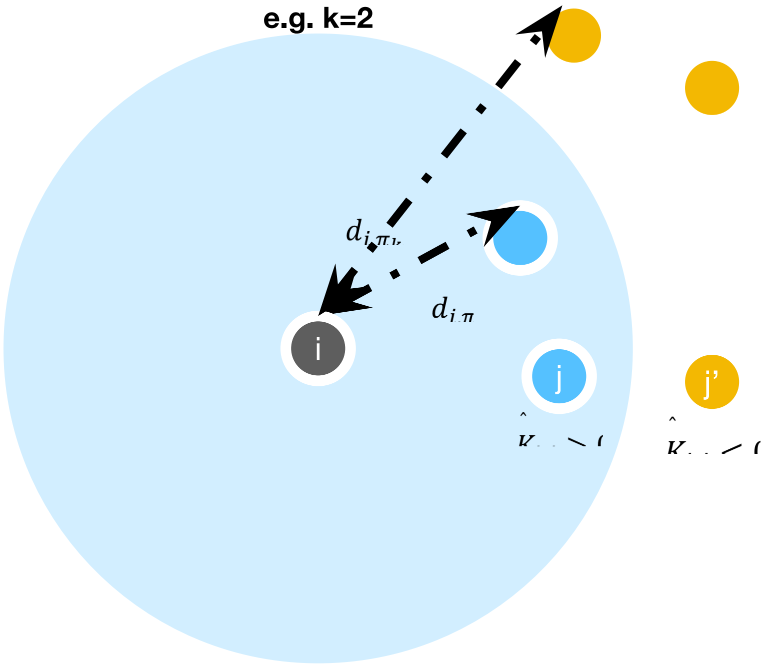
(Making K differentiable)

k-NN Matrix (K) ≈ Adjacency Matrix (Adj)

distance to the k^th nearest neighbor of node i

$$\hat{K}_{i,j} = \begin{cases} -(\|X_i - X_j\| - \frac{d_{i,\pi_k} + d_{i,\pi_{k+1}}}{2}) & \text{if } i \neq j \\ 0 & \text{if } i = j \end{cases}$$

- $\hat{K}_{i,j}$ is differentiable function of X_i and X_j
- $\hat{K}_{i,j} \geq 0$ if node j is one of the k nearest neighbors of node i
- $\hat{K}_{i,j} < 0$ otherwise



Kruiger et al.: Graph layouts by t-SNE. Comput. Graph. Forum36(3), 283–294 (2017)

Neighborhood Preservation

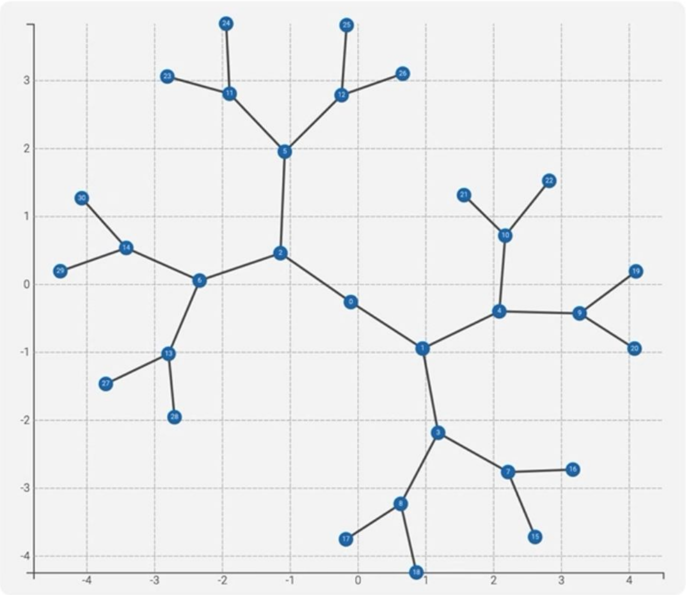
(Making K differentiable)

Q_{NP} := Jaccard Index (K, Adj)

distance to the k[th] nearest neighbor of node i

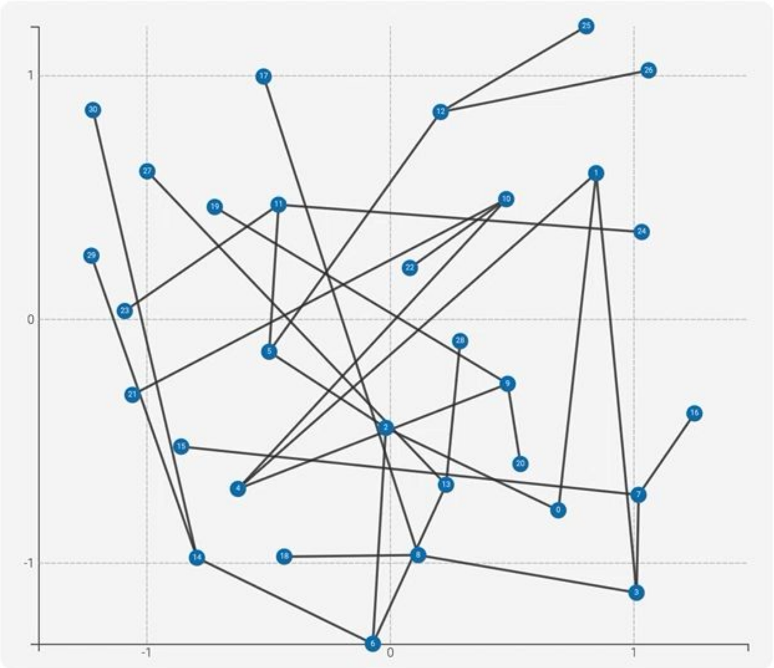
$$\hat{K}_{i,j} = \begin{cases} -(||X_i - X_j|| - \frac{d_{i,\pi_k} + d_{i,\pi_{k+1}}}{2}) & \text{if } i \neq j \\ 0 & \text{if } i = j \end{cases}$$

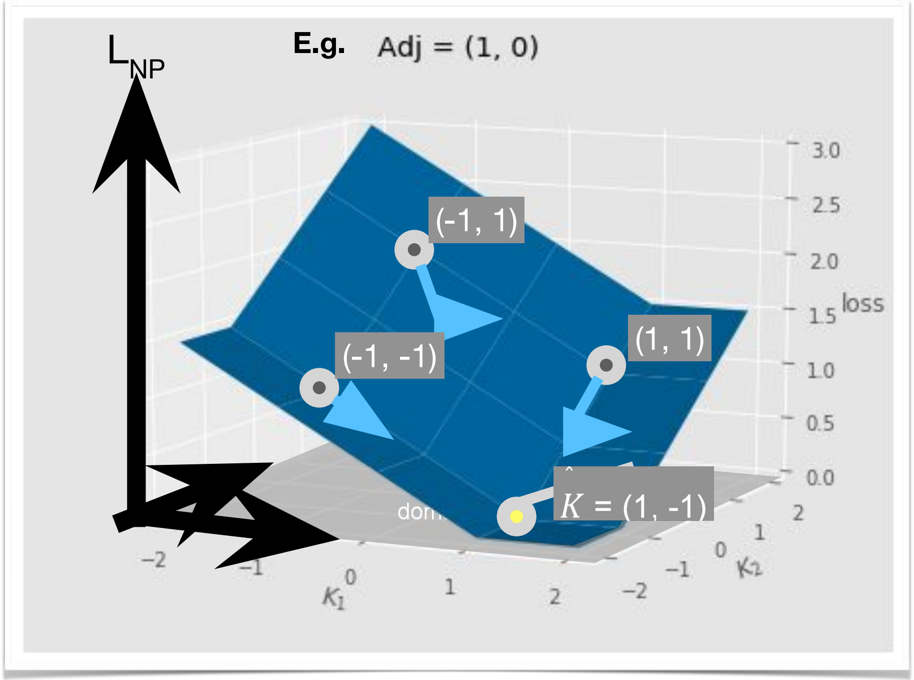
- $\hat{K}_{i,j}$ is differentiable function of X_i and X_j
- $\hat{K}_{i,j} \geq 0$ if node j is one of the k nearest neighbors of node i
- $\hat{K}_{i,j} < 0$ otherwise



Neighborhood Preservation

(Relax Jaccard Index)





-
-
-

L_N

≈ 2

Berman et al.: The Lovász-softmax loss: a tractable surrogate for the optimization of the intersection-over-union measure in neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4413–4421 (2018)

In general:

- We find a loss function that is <u>differentiable</u> with respect to the layout X.
- If the original criterion is *not* differentiable, we find a <u>surrogate function</u>.

Vertex Resolution

- Goal / Intuition:

Distribute nodes evenly

- Quality measure (higher the better):

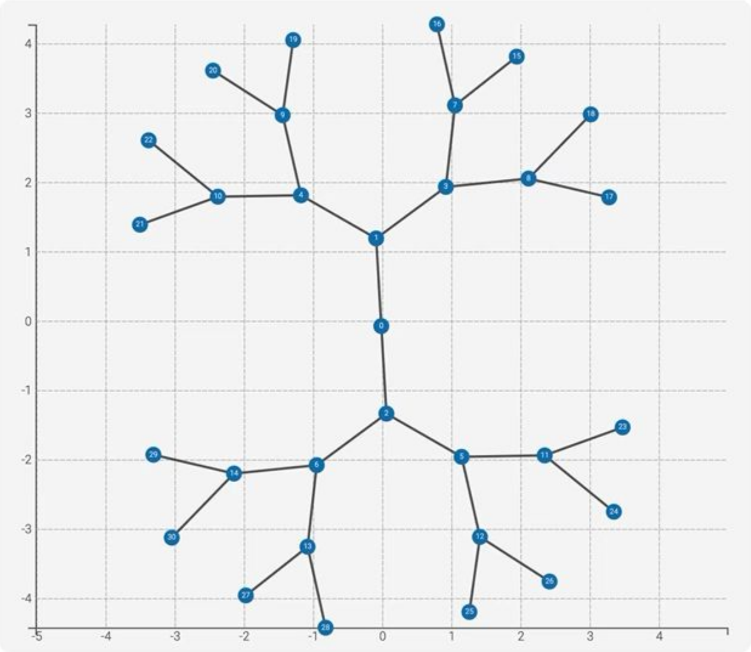
$$Q_{VR} = \frac{\min_{i \neq j} ||X_i - X_j||}{\underbrace{max_{k \neq l} ||X_k - X_l||}_{d_{max}}}$$

- Loss function:

For any target resolution $r \in [0,1]$

$$L_{VR} = \sum_{i,j \in V, i \neq j} \left[ReLU\left(1 - \frac{||X_i - X_j||}{r \cdot d_{max}}\right)\right]^2$$

- We take $r := \frac{1}{\sqrt{|V|}}$



Vertex Resolution (cont.)

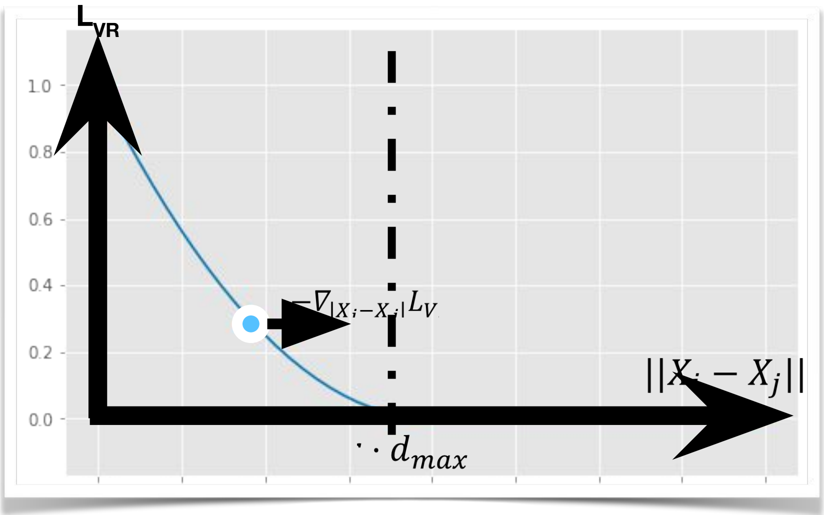
$$Q_{VR} = \frac{\min_{i \neq j} ||X_i - X_j||}{\underbrace{max_{k \neq l} ||X_k - X_l||}_{d_{max}}}$$

- Loss function:

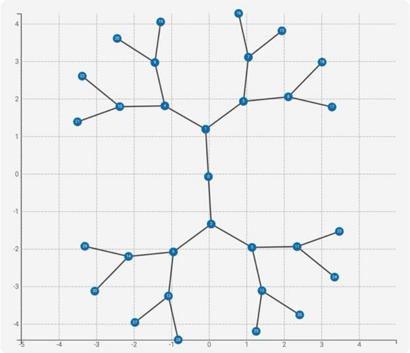
For any target resolution $r \in [0,1]$

$$L_{VR} = \sum_{i,j \in V, i \neq j} [ReLU(1 - \frac{||X_i - X_j||}{\underbrace{r}_{\text{Target Resolution}} \cdot d_{max}})]^2$$

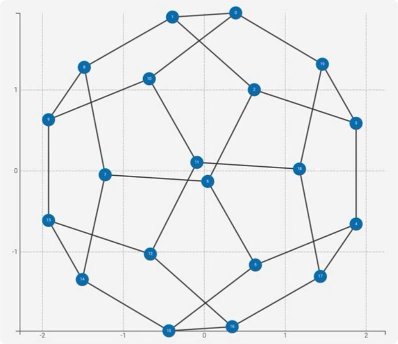
- We take $r := \frac{1}{\sqrt{|V|}}$



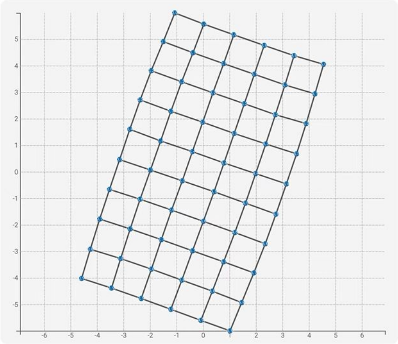
Seven more, details in the paper



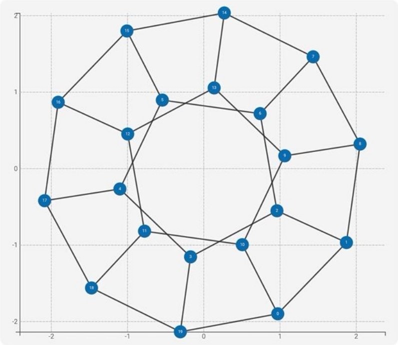
Vertex Resolution



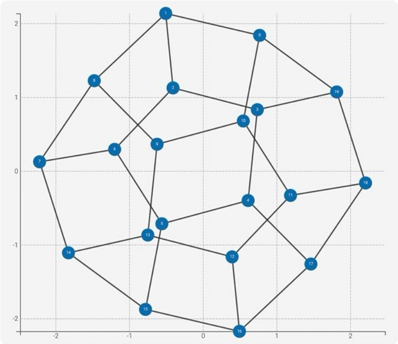
Crossing Angle Maximization



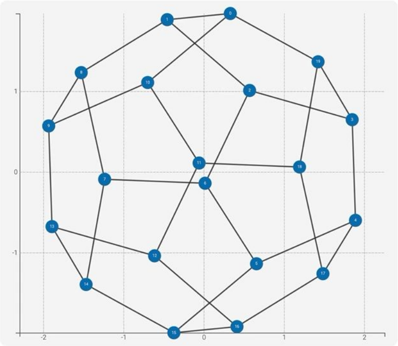
Aspect Ratio



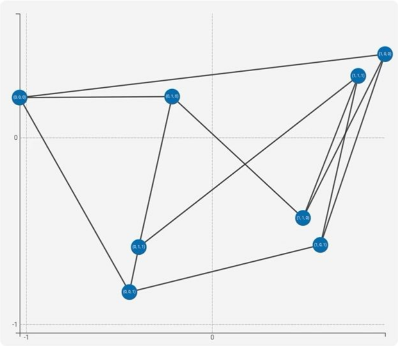
Gabriel Property



Ideal Edge Length

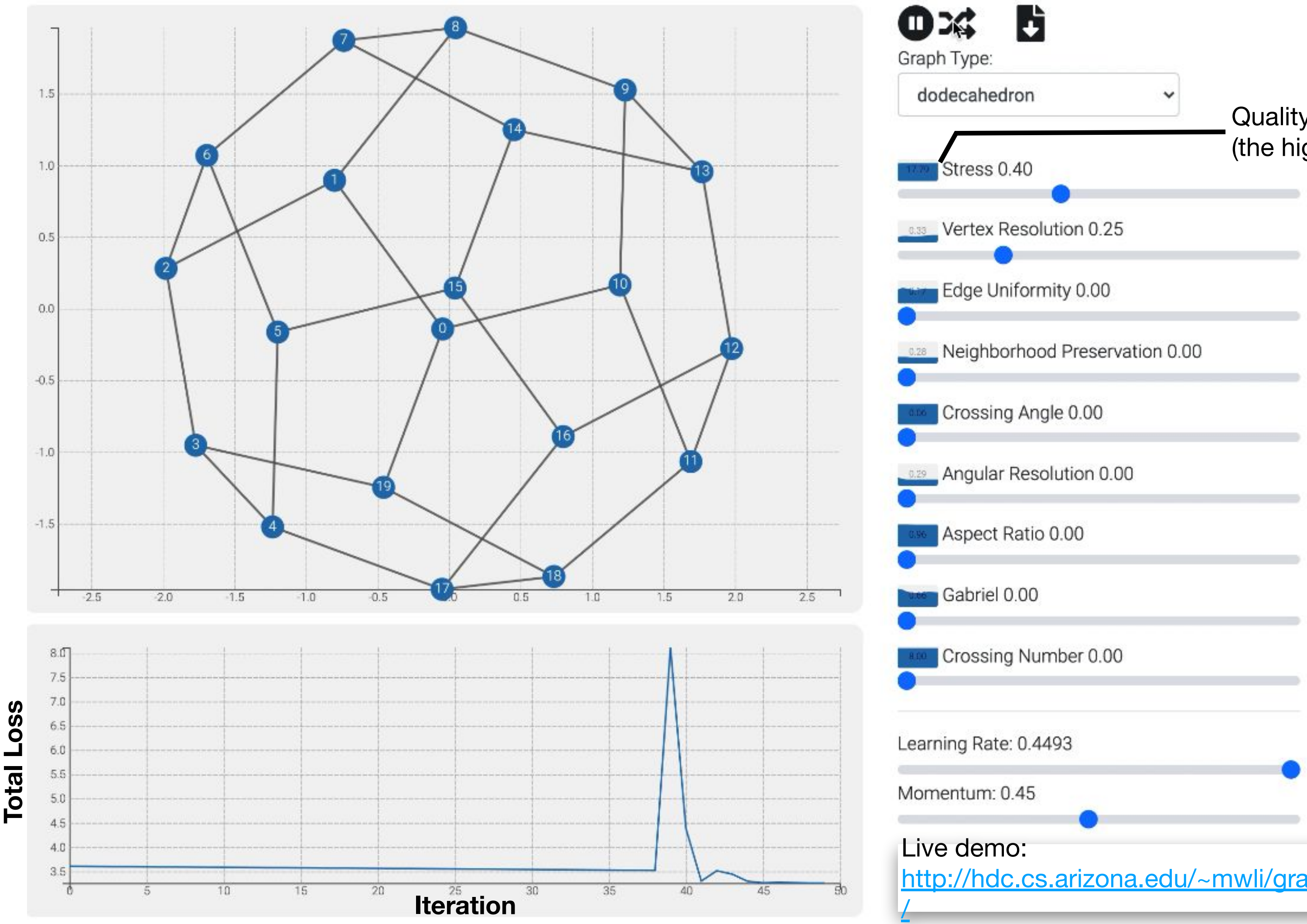


Angular Resolution



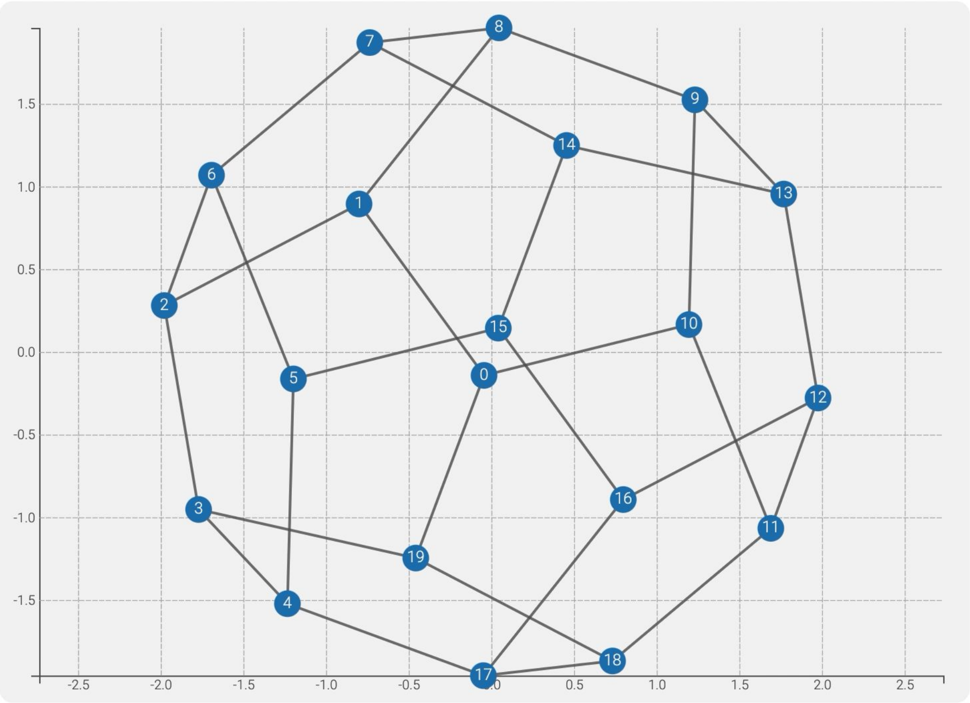
Crossing Number

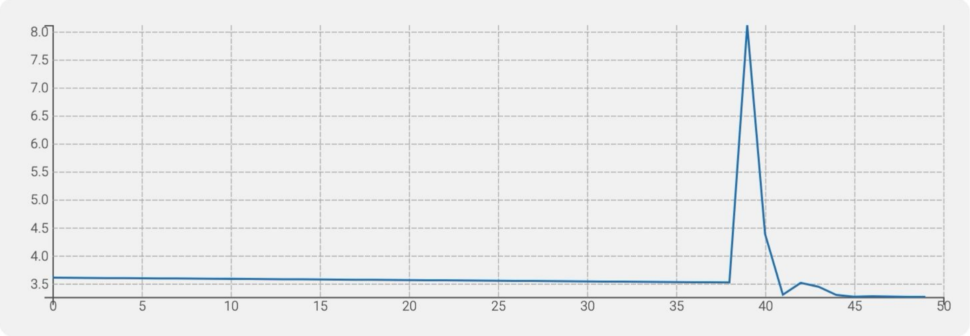
Graph Drawing via Gradient Descent, $(GD)^2$ by Reyan Ahmed, Felice De Luca, Sabin Devkota, Stephen Kobourov and Mingwei Li https://arxiv.org/abs/2008.05584



Quality Measure
(the higher, the better)

Live demo:
http://hdc.cs.arizona.edu/~mwli/graph-drawing/





Graph Type:

dodecahedron

17.79 Stress 0.40

0.33 Vertex Resolution 0.25

0.17 Edge Uniformity 0.00

0.28 Neighborhood Preservation 0.00

0.06 Crossing Angle 0.00

0.29 Angular Resolution 0.00

0.96 Aspect Ratio 0.00

0.66 Gabriel 0.00

8.00 Crossing Number 0.00

Learning Rate: 0.4493

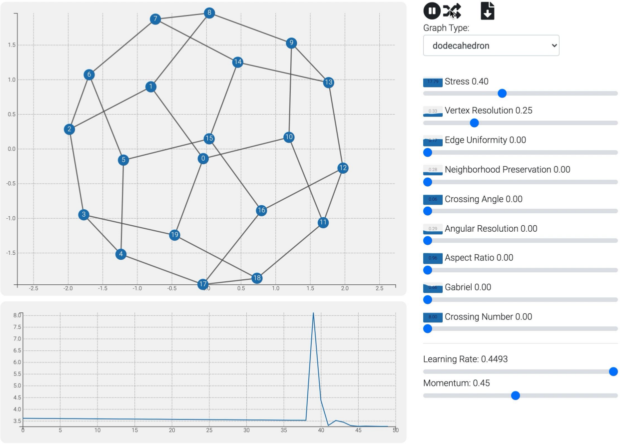
Momentum: 0.45

- Introduction
- Related Work
- The $(GD)^2$ Framework
- Properties and Measures
- Conclusions

Discussion?

Conclusion

- We proposed a general framework, **(GD)^2**, that optimizes **multiple** drawing criteria for graph layouts.
- To optimize multiple criteria jointly, we take a **weighted sum** of individual loss functions.
- For each criterion, we either optimize it directly or find a **surrogate** loss function if the criterion is not differentiable.



Live demo: http://hdc.cs.arizona.edu/~mwli/graph-drawing/

Paper (arXiv): https://arxiv.org/abs/2008.05584
Graph Drawing via Gradient Descent, (GD)^2
by Reyan Ahmed (abureyanahmed@email.arizona.edu), Felice De Luca, Sabin Devkota, Stephen Kobourov and Mingwei Li (mwli@email.arizona.edu)