# CENTRALITY-CONSTRAINED GRAPH EMBEDDING

*Brian Baingana, Georgios B. Giannakis*

Dept. of ECE, University of Minnesota, Minneapolis, MN 55455, USA

## ABSTRACT

Visual rendering of graphs is a key task in the mapping of complex network data. Although most graph drawing algorithms emphasize aesthetic appeal, certain applications such as travel-time maps place more importance on visualization of structural network properties. The present paper advocates a graph embedding approach with centrality considerations to comply with node hierarchy. The problem is formulated as one of constrained multi-dimensional scaling (MDS), and it is solved via block coordinate descent iterations with successive approximations and guaranteed convergence to a KKT point. In addition, a regularization term enforcing graph smoothness is incorporated with the goal of reducing edge crossings. Experimental results demonstrate that the algorithm converges, and can be used to efficiently embed large graphs on the order of thousands of nodes.

***Index Terms***— MDS, graph embedding, coordinate descent.

## 1. INTRODUCTION

Graphs offer a valuable means of encoding relational information between entities of complex systems, arising in modern communications, transportation and social networks, among others. Despite the abundance of network analysis techniques, information visualization is a powerful tool for capturing patterns that may not be apparent in large-scale systems. However, most visualization algorithms focus more on aesthetic appeal than the structural characteristics of the underlying data. Such network structure is captured through graph-theoretic notions such as node centrality and network cohesion.

The present paper deals with embedding graphs for visualization while adhering to the underlying node centrality structure. Centrality measures capture the relative importance of network nodes among their peers. Betweenness centrality for instance, describes the extent to which information is routed through a specific node by measuring the fraction of all shortest paths traversing this node; see e.g., [1, p. 89]. Other measures include closeness, eigenvalue, and degree centrality. To incorporate centrality using any of these metrics, an MDS (so-termed stress [2, Chap. 3]) criterion is adopted, under radial constraints that place nodes of higher centrality closer to the origin of the graph embedding. MDS seeks a low-dimensional depiction of high-dimensional data in which pairwise Euclidean distances between embedding coordinates are close (in a well-defined sense) to the dissimilarities between the original data points. Closeness criteria (a.k.a. stress costs) are generally non-convex, and the quest for global optimality is challenging because ordinary descent methods do not have optimality guarantees, and are sensitive to initialization. Successive approximation with global and convex upper bounds is used in [2, Chap. 8] to minimize the stress cost yielding near-optimal results.

The novel approach exploits the block separability inherent to the proposed model and adapts the coordinate descent algorithm to determine the optimal embedding. Edge crossings are minimized by regularizing the cost with a smoothness promoting term weighted by a tuning parameter. Smoothness encourages nodes that share an edge to lie closer to each other in the embedding. As a result, the length and hence the number of edge crossings in the network visualization is markedly reduced. In addition, the regularization term offers the benefit of incorporating the underlying network topology when the dissimilarities considered are not graph-theoretic e.g., Euclidean distances between feature vectors associated with each node. Moreover, numerical tests illustrate that judicious selection of the tuning parameter results in fewer block coordinate descent iterations, which in turn yields a visually appealing embedding.

To place the present work in context, a prior approach iteratively minimizes a weighted stress function with iteration-dependent weights chosen to incorporate radial constraints [3]. However, it is limited to graph-theoretic dissimilarities, and offers no convergence guarantees. A heuristic algorithm for network visualization uses the $k$-core decomposition to hierarchically place nodes within "onion-like" concentric shells [4]. Although effective for large-scale networks, it has no optimality associated with it, and is limited to visualization only in 2 dimensions. The proposed approach scales well for large networks under a well-defined optimality criterion with a convergence guarantee.

## 2. MODEL AND PROBLEM STATEMENT

Consider a network represented by an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{E}$ denotes the set of edges, and $\mathcal{V}$ the set of vertices with cardinality $|\mathcal{V}| = N$. Let $\delta_{ij}$ denote the pairwise dissimilarity (edge weight) between any two nodes $i$ and $j$. Given the set $\{\delta_{ij}\}$ and the prescribed embedding dimension $p$, the graph embedding task amounts to finding $p \times 1$ vectors $\{\mathbf{x}_i\}_{i=1}^N$ so that the embedding coordinates $\mathbf{x}_i$ and $\mathbf{x}_j$ satisfy $\|\mathbf{x}_i - \mathbf{x}_j\|_2 \approx \delta_{ij}$.

With $\delta_{ij} = \delta_{ji}$, it suffices to know $\{\{\delta_{ij}\}_{j=1}^N\}_{i=j+1}^N$, or, be possible to determine them from $\mathcal{G}$. Most visualization schemes assign $\delta_{ij}$ to the shortest path distance between nodes $i$ and $j$. In this work, the Euclidean commute-time distance (ECTD) is adopted because it decreases as the number of shortest paths between node pairs increases [5]. This is more reasonable since having multiple shortest paths between node pairs endows them with a higher level of accessibility by e.g., a random walker on the graph.

MDS amounts to solving the following problem:

$$\text{(P0)} \quad \{\hat{\mathbf{x}}_i\}_{i=1}^N = \arg\min_{\mathbf{x}_1,\ldots,\mathbf{x}_N} \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N [\|\mathbf{x}_i - \mathbf{x}_j\|_2 - \delta_{ij}]^2. \quad (1)$$

Turning attention to node centralities $\{c_i\}_{i=1}^N$, those can be obtained using a number of algorithms [1, Chap. 4]. Centrality structure will be imposed on (1) by constraining $\mathbf{x}_i$ to have a centrality-dependent radial distance $f(c_i)$, where $f(.)$ is a monotone decreasing function. The resulting constrained optimization problem now becomes

$$\text{(P1)} \quad \{\hat{\mathbf{x}}_i\}_{i=1}^N = \underset{\mathbf{x}_1,\ldots,\mathbf{x}_N}{\arg\min} \quad \tfrac{1}{2}\sum_{i=1}^N\sum_{j=1}^N \left[\|\mathbf{x}_i - \mathbf{x}_j\|_2 - \delta_{ij}\right]^2$$
$$\text{s. to} \quad \|\mathbf{x}_i\|_2 = f(c_i),\ i = 1,\ldots,N. \quad (2)$$

Although P0 is non-convex, standard solvers rely on gradient descent iterations but have no guarantees of convergence to the global optima [6]. Lack of convexity is exacerbated in P1 by the non-convex constraint set rendering its solution even more challenging than that of P0. However, considering a single embedding vector $\mathbf{x}_i$, and fixing the rest $\{\mathbf{x}_j\}_{j\neq i}$, the constraint set simplifies to $\|\mathbf{x}_i\|_2 = f(c_i)$, for which an appropriate relaxation can be sought. Key to the algorithm proposed next lies in this inherent decoupling of the centrality constraints.

## 3. BCD WITH SUCCESSIVE APPROXIMATIONS

By exploiting the separable nature of the cost as well as the norm constraints in (2), block coordinate descent (BCD) will be adopted in this section to arrive at a solution approaching the global optimum. To this end, the centering constraint $\sum_{i=1}^N \mathbf{x}_i = \mathbf{0}$, typically invoked to fix the inherent translation ambiguity, will be dropped first so that the problem remains decoupled across nodes. The effect of this relaxation can be compensated for by computing the centroid of the solution of (2), and subtracting it from each coordinate. The $N$ equality norm constraints are also relaxed to $\|\mathbf{x}_i\|_2 \leq f(c_i)$. Although the entire constraint set is non-convex, each relaxed constraint is a convex and closed Euclidean ball with respect to each node in the network.

Let $\mathbf{x}_i^r$ denote the minimizer of the optimization problem over block $i$, when the remaining blocks $\{\mathbf{x}_j\}_{j\neq i}$ are fixed during the BCD iteration $r$. By fixing the blocks $\{\mathbf{x}_j\}_{j\neq i}$ to their values from the most recent iterations, the sought embedding is obtained as

$$\{\hat{\mathbf{x}}_i\}_{i=1}^N = \underset{\{\mathbf{x}_i : \|\mathbf{x}_i\|_2 \leq f(c_i)\}}{\arg\min} \frac{1}{2}\sum_{i=1}^N\sum_{j=1}^N \left[\|\mathbf{x}_i - \mathbf{x}_j\|_2 - \delta_{ij}\right]^2 \quad (3)$$

or equivalently as

$$\underset{\{\mathbf{x}_i : \|\mathbf{x}_i\|_2 \leq f(c_i)\}}{\arg\min} \frac{(N-1)}{2}\|\mathbf{x}_i\|_2^2 - \mathbf{x}_i^T(\sum_{j<i}\mathbf{x}_j^r + \sum_{j>i}\mathbf{x}_j^{r-1})$$
$$-\sum_{j<i}\delta_{ij}\|\mathbf{x}_i - \mathbf{x}_j^r\|_2 - \sum_{j>i}\delta_{ij}\|\mathbf{x}_i - \mathbf{x}_j^{r-1}\|_2 \quad (4)$$

where $\sum_{j<i}(.) := \sum_{j=1}^{i-1}(.)$ and $\sum_{j>i}(.) := \sum_{j=i+1}^{N}(.)$. With the last two sums in (4) being non-convex and non-smooth, convergence of the BCD algorithm cannot be guaranteed [7, p. 272]. Moreover, it is desired to have each per-iteration subproblem solvable to global optimality, in closed form and at a minimum computational cost. The proposed approach seeks a global upper bound of the objective with the desirable properties of smoothness and convexity. To this end, consider the function $\Psi(\mathbf{x}_i) := \psi_1(\mathbf{x}_i) - \psi_2(\mathbf{x}_i)$, where

$$\psi_1(\mathbf{x}_i) := \frac{(N-1)}{2}\|\mathbf{x}_i\|_2^2 - \mathbf{x}_i^T(\sum_{j<i}\mathbf{x}_j^r + \sum_{j>i}\mathbf{x}_j^{r-1}) \quad (5)$$

and

$$\psi_2(\mathbf{x}_i) := \sum_{j<i}\delta_{ij}\|\mathbf{x}_i - \mathbf{x}_j^r\|_2 + \sum_{j>i}\delta_{ij}\|\mathbf{x}_i - \mathbf{x}_j^{r-1}\|_2. \quad (6)$$

Note that $\psi_1(\mathbf{x}_i)$ is a convex quadratic function, and that $\psi_2(\mathbf{x}_i)$ is convex (with respect to $\mathbf{x}_i$) but non-differentiable. The first-order approximation of (6) at any point in its domain is a global underestimate of $\psi_2(\mathbf{x}_i)$. Despite the non-smoothness at some points, such a lower bound can always be established using its subdifferential. As a consequence of the convexity of $\psi_2(\mathbf{x}_i)$, it holds that [7, p. 731]

$$\psi_2(\mathbf{x}) \geq \psi_2(\mathbf{x}_0) + \mathbf{g}^T(\mathbf{x}_0)(\mathbf{x} - \mathbf{x}_0), \forall \mathbf{x} \in \text{dom}(\psi_2) \quad (7)$$

where $\mathbf{g}(\mathbf{x}) \in \partial\psi_2(\mathbf{x})$ is a subgradient within the subdifferential set, $\partial\psi_2(\mathbf{x})$ of $\psi_2(\mathbf{x})$. The subdifferential of $\|\mathbf{x}_i - \mathbf{x}_j\|_2$ with respect to $\mathbf{x}_i$ is given by

$$\partial_{\mathbf{x}_i}\|\mathbf{x}_i - \mathbf{x}_j\|_2 = \begin{cases} \frac{\mathbf{x}_i - \mathbf{x}_j}{\|\mathbf{x}_i - \mathbf{x}_j\|_2}, & \text{if } \mathbf{x}_i \neq \mathbf{x}_j \\ \mathbf{y} \in \mathbb{R}^p : \|\mathbf{y}\|_2 \leq 1, & \text{otherwise} \end{cases} \quad (8)$$

which implies that

$$\partial_{\mathbf{x}_i}\psi_2(\mathbf{x}_i) = \sum_{j=1}^N \delta_{ij}\partial_{\mathbf{x}_i}\|\mathbf{x}_i - \mathbf{x}_j\|_2. \quad (9)$$

Using (7), it is possible to lower bound (6) by

$$\psi_2'(\mathbf{x}_i, \mathbf{x}_0) = \sum_{j<i}\delta_{ij}\left[\|\mathbf{x}_0 - \mathbf{x}_j^r\|_2 + (\mathbf{g}_j^r)^T(\mathbf{x}_0)(\mathbf{x}_i - \mathbf{x}_0)\right]$$
$$+\sum_{j>i}\delta_{ij}\left[\|\mathbf{x}_0 - \mathbf{x}_j^{r-1}\|_2 + (\mathbf{g}_j^{r-1})^T(\mathbf{x}_0)(\mathbf{x}_i - \mathbf{x}_0)\right]. \quad (10)$$

Consider now $\Phi(\mathbf{x}_i, \mathbf{x}_0) := \psi_1(\mathbf{x}_i) - \psi_2'(\mathbf{x}_i, \mathbf{x}_0)$, and note that $\Phi(\mathbf{x}_i, \mathbf{x}_0)$ is convex and upper bounds globally the cost in (4). The proposed BCD algorithm involves successive approximations using (10), and yields the following QCQP for each block

$$\text{(P2)} \qquad \underset{\{\mathbf{x}_i : \|\mathbf{x}_i\|_2 \leq f(c_i)\}}{\arg\min} \quad \Phi(\mathbf{x}_i, \mathbf{x}_0). \quad (11)$$

For convergence, $\mathbf{x}_0$ must be selected to satisfy the following conditions [8]:

$$\Phi(\mathbf{x}_0, \mathbf{x}_0) \quad = \quad \Psi(\mathbf{x}_0), \quad \forall \mathbf{x}_0 \in \mathcal{C}, \forall i \quad (12a)$$
$$\Phi(\mathbf{x}_i, \mathbf{x}_0) \quad \geq \quad \Psi(\mathbf{x}_i), \quad \|\mathbf{x}_i\|_2 \leq f(c_i), \forall i \quad (12b)$$

where $\mathcal{C} := \bigcup_{i=1}^N\{\mathbf{x}_i : \|\mathbf{x}_i\|_2 \leq f(c_i)\}$. In addition, $\Phi(\mathbf{x}_i, \mathbf{x}_0)$ must be continuous in $(\mathbf{x}_i, \mathbf{x}_0)$. Upon selecting $\mathbf{x}_0 = \mathbf{x}_i^{r-1}$, the iterate $\mathbf{x}_i^{r-1}$ satisfies (12a) and (12b). Taking successive approximations around $\mathbf{x}_i^{r-1}$ in P2, ensures the uniqueness of

$$\mathbf{x}_i^r \quad = \quad \underset{\{\mathbf{x}_i : \|\mathbf{x}_i\|_2 \leq f(c_i)\}}{\arg\min} \quad \tfrac{(N-1)}{2}\mathbf{x}_i^T\mathbf{x}_i$$
$$-\mathbf{x}_i^T[\sum_{j<i}(\mathbf{x}_j^r + \delta_{ij}\mathbf{g}_j^r(\mathbf{x}_i^{r-1}))$$
$$+\sum_{j>i}(\mathbf{x}_j^{r-1} + \delta_{ij}\mathbf{g}_j^{r-1}(\mathbf{x}_i^{r-1}))] \quad (13)$$

Solving (13) amounts to obtaining the solution of the unconstrained QP, $(\mathbf{x}_i^*)^r$, and projecting it onto $\{\mathbf{x}_i : \|\mathbf{x}_i\|_2 \leq f(c_i)\}$; that is,

$$\mathbf{x}_i^r = \begin{cases} \frac{(\mathbf{x}_i^*)^r}{\|(\mathbf{x}_i^*)^r\|_2} f(c_i), & \text{if } \ \|(\mathbf{x}_i^*)^r\|_2 > f(c_i) \\ (\mathbf{x}_i^*)^r, & \text{otherwise} \end{cases} \quad (14)$$

where

$$\begin{aligned}(\mathbf{x}_i^*)^r &= \tfrac{1}{N-1}\big[\sum_{j<i}(\mathbf{x}_j^r + \delta_{ij}\mathbf{g}_j^r(\mathbf{x}_i^{r-1}))\\ &\quad + \sum_{j>i}(\mathbf{x}_j^{r-1} + \delta_{ij}\mathbf{g}_j^{r-1}(\mathbf{x}_i^{r-1}))\big]. \qquad (15)\end{aligned}$$

It is desirable but not necessary that the algorithm converges because depending on the application, reasonable network visualizations can be found with fewer iterations. In fact, successive approximations merely provide a more refined graph embedding that maybe more aesthetically appealing.

Although the proposed algorithm is guaranteed to converge, the solution is only unique up to a rotation and a translation (cf. MDS). In order to eliminate the translational ambiguity, the embedding can be centered at the origin. Assuming that the optimal blocks determined within outer iteration $r$ are reassembled into the embedding matrix $\mathbf{X}^r := \left[(\mathbf{x}_1^r)^T, \ldots, (\mathbf{x}_N^r)^T\right]^T$, the final step involves subtracting the mean from each coordinate using the centering operator as follows, $\mathbf{X} = (\mathbf{I} - N^{-1}\mathbf{1}\mathbf{1}^T)\mathbf{X}^r$, where $\mathbf{I}$ denotes the $N \times N$ identity matrix, and $\mathbf{1}$ is the $N \times 1$ vector of all ones.

The novel graph embedding scheme is summarized as Algorithm 1 with matrix $\mathbf{\Delta}$ having $(i, j)$th entry the dissimilarity $\delta_{ij}$.

**Algorithm 1** BCD algorithm with successive approximations

```
Input: $\{c_i\}_{i=1}^N$, $\mathbf{\Delta}$, $\epsilon$
Initialize $\mathbf{X}^0$, $r = 0$
repeat
    $r = r + 1$
    for $i = 1 \ldots N$ do
        Compute $\mathbf{x}_i^r$ according to (14) and (15)
        $\mathbf{X}^r(i, :) = (\mathbf{x}_i^r)^T$
    end for
until $\|\mathbf{X}^r - \mathbf{X}^{r-1}\|_F \leq \epsilon$
$\mathbf{X} = (\mathbf{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^T)\mathbf{X}^r$
```

## 4. ENFORCING GRAPH SMOOTHNESS

In this section, the MDS stress in (3) is regularized through an additional constraint that encourages smoothness over the graph. Intuitively, despite the requirement that the node placement in low-dimensional Euclidean space respects inherent network structure, through preserving e.g., node centralities, neighboring nodes in a graph-theoretic sense (meaning nodes that share an edge) are expected to be close in Euclidean distance within the embedding. Such a requirement can be captured by incorporating a constraint that discourages large distances between neighboring nodes. In essence, this constraint enforces smoothness over the graph embedding.

A popular choice of a smoothness-promoting function is $h(\mathbf{X}) := \text{Tr}(\mathbf{X}^T \text{Ł} \mathbf{X})$, where $\text{Tr}(.)$ denotes the trace operator, and $\text{Ł} := \mathbf{D} - \mathbf{A}$ is the graph Laplacian with $\mathbf{D}$ a diagonal matrix whose $(i, i)$th entry is the degree of node $i$, and $\mathbf{A}$ the adjacency matrix. It can be shown that $h(\mathbf{X}) = (1/2)\sum_{i=1}^{N}\sum_{i=1}^{N} a_{ij}\|\mathbf{x}_i - \mathbf{x}_j\|_2^2$, where $a_{ij}$ is the $(i, j)$th entry of $\mathbf{A}$. Motivated by penalty methods in optimization, the cost in (2) will be augmented as follows

$$\begin{aligned}\text{(P3)} \quad \underset{\mathbf{x_1}, \ldots, \mathbf{x_N}}{\arg\min} \quad & \tfrac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\left[\|\mathbf{x}_i - \mathbf{x}_j\|_2 - \delta_{ij}\right]^2\\ & + \tfrac{\lambda}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} a_{ij}\|\mathbf{x}_i - \mathbf{x}_j\|_2^2\\ \text{s. to} \quad & \|\mathbf{x}_i\|_2 = f(c_i), i = 1, \ldots, N \qquad (16)\end{aligned}$$

where the scalar $\lambda \geq 0$ controls the degree of smoothness. The penalty term has a separable structure and is convex with respect to $\mathbf{x}_i$. Consequently, P3 lies within the framework of successive approximations required to solve each per-iteration subproblem. Following the same relaxations and invoking the successive upper bound approximations described earlier, yields the following QCQP

$$\begin{aligned}\mathbf{x}_i^r &= \underset{\{\mathbf{x}_i : \|\mathbf{x}_i\|_2 \leq f(c_i)\}}{\arg\min} \quad (N + \lambda d_{ii} - 1)\mathbf{x}_i^T\mathbf{x}_i\\ &\quad - \mathbf{x}_i^T\big[\sum_{j<i}((1 + \lambda a_{ij})\mathbf{x}_j^r + \delta_{ij}\mathbf{g}_j^r(\mathbf{x}_i^{r-1}))\\ &\quad + \sum_{j>i}((1 + \lambda a_{ij})\mathbf{x}_j^{r-1} + \delta_{ij}\mathbf{g}_j^{r-1}(\mathbf{x}_i^{r-1}))\big] \qquad (17)\end{aligned}$$

with $d_{ii} := \sum_{j=1}^{N} a_{ij}$ denoting the degree of node $i$.

The solution of (17) can be expressed as [cf. (14)]

$$\begin{aligned}(\mathbf{x}_i^*)^r &= \tfrac{1}{N + \lambda d_{ii} - 1}\big[\sum_{j<i}((1 + \lambda a_{ij})\mathbf{x}_j^r + \delta_{ij}\mathbf{g}_j^r(\mathbf{x}_i^{r-1}))\\ &\quad + \sum_{j>i}((1 + \lambda a_{ij})\mathbf{x}_j^{r-1} + \delta_{ij}\mathbf{g}_j^{r-1}(\mathbf{x}_i^{r-1}))\big]. \qquad (18)\end{aligned}$$

With $\lambda$ given, Algorithm 2 summarizes the steps to determine the constrained embedding with a smoothness penalty.

**Algorithm 2** Incorporating smoothness in Algorithm 1

```
Input: $\mathbf{A}$, $\{c_i\}_{i=1}^N$, $\mathbf{\Delta}$, $\epsilon$, $\lambda$
Initialize $\mathbf{X}^0$, $r = 0$
repeat
    $r = r + 1$
    for $i = 1 \ldots N$ do
        Compute $\mathbf{x}_i^r$ according to (14) and (18)
        $\mathbf{X}^r(i, :) = (\mathbf{x}_i^r)^T$
    end for
until $\|\mathbf{X}^r - \mathbf{X}^{r-1}\|_F \leq \epsilon$
$\mathbf{X} = (\mathbf{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^T)\mathbf{X}^r$
```

## 5. NUMERICAL EXPERIMENTS

### 5.1. Visualizing the London Tube

In the first experiment, an undirected graph of 307 nodes representing the London tube, an underground train transit network,[1] is considered. The nodes represent stations whereas the edges represent the routes connecting them. The objective is to generate an embedding in which stations traversed by most routes are placed closer to the center, thus highlighting their relative significance in metro transit. Such information is best captured by the *betweenness centrality*, which is defined as $c_i := \sum_{j \neq k \neq i \in \mathcal{V}} \sigma_{j,k}^i / (\sum_{i \in \mathcal{V}} \sigma_{j,k}^i)$, where $\sigma_{j,k}^i$ is the number of shortest paths between nodes $j$ and $k$ through node

[1] https://wikis.bris.ac.uk/display/ipshe/London+Tube

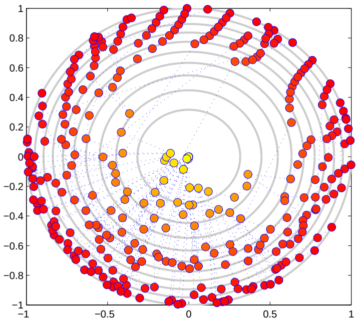


**Fig. 1**. Centrality-constrained embedding of the London tube

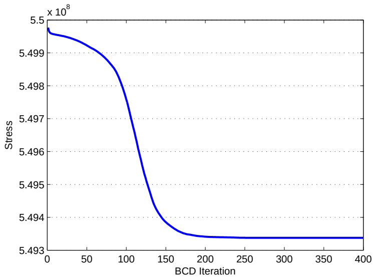


**Fig. 2**. MDS stress iterations

$i$ [10]. The centrality values were transformed as follows:

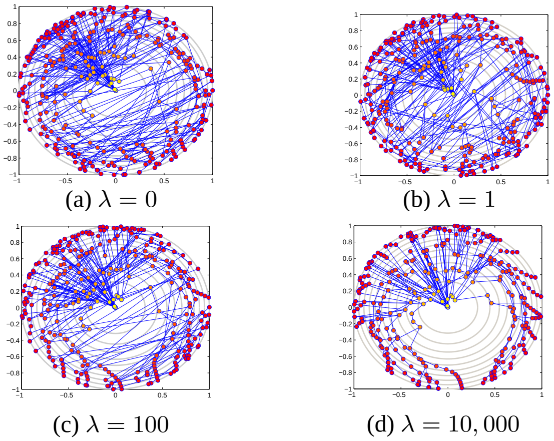
$$f\left(c_i\right) = \frac{\operatorname{diam}\left(\mathcal{G}\right)}{2}\left(1 - \frac{c_i - \min\limits_{i\in\mathcal{V}} c_i}{\max\limits_{i\in\mathcal{V}} c_i - \max\limits_{i\in\mathcal{V}} c_i}\right) \tag{19}$$

with $\operatorname{diam}(\mathcal{G})$ denoting the diameter of $\mathcal{G}$. Simulations were run for several values of $\lambda$ starting with $\lambda = 0$, and the resultant two-dimensional ($p = 2$) embeddings were plotted. Figure 1 depicts the optimal embedding obtained without a smoothness penalty. The color grading reflects the centrality levels of the nodes from highest (yellow) to lowest (red). Algorithm 1 converged after approximately $150$ outer iterations as shown in Figure 2.

Figure 3 illustrates the effect of including the smoothness penalty. Increasing $\lambda$ promotes embeddings in which edge crossings are minimized. This intuitively makes sense because by forcing single-hop neighbors to lie close to each other, the average edge length decreases, leading to fewer edge crossings. In addition, increasing $\lambda$ yielded embeddings that were aesthetically more appealing under fewer iterations. For instance, setting $\lambda = 10,000$ required only $30$ iterations for a visualization that is comparable to running $150$ iterations with $\lambda = 0$. An application of this work is travel time cartography in which edge lengths reflect the amount of time it takes to travel between stations. In this case, $f(c_i)$ is equivalent to the transit time from a station of interest to any other station $i$. By selecting a station and specifying the travel time to all other nodes, an informative radial map centered at the station of interest can be generated.

### 5.2. Collaboration network of Arxiv General Relativity

In this experiment, a large social network is considered from the e-print arXiv repository covering scientific collaborations between authors on papers submitted to the "General Relativity and Quantum Cosmology" category (January $1993$ to April $2003$) [9]. The nodes represent authors and an edge exists between nodes $i$ and $j$ if authors $i$ and $j$ co-authored a paper. Although the network contains $5,242$ nodes, the embedding considered only its largest strongly connected component comprising $4,158$ nodes. The objective was to embed the network so that authors whose research is most related to the majority of the others are placed closer to the center.


(a) $\lambda = 0$

(b) $\lambda = 1$

(c) $\lambda = 100$

(d) $\lambda = 10,000$

**Fig. 3**. Visualizing the London tube with a smoothing penalty

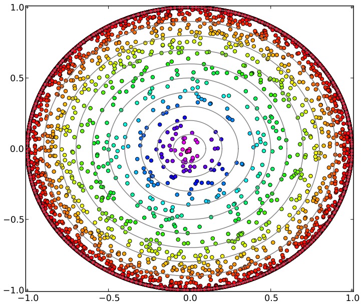


**Fig. 4**. Embedding of a large social network

This behavior is best captured by the *closeness centrality* that is defined as $c_i := \left(\sum_{j\in\mathcal{V}} d_{ij}\right)^{-1}$, where $d_{ij}$ is the geodesic distance (lowest sum of edge weights) between nodes $i$ and $j$; and captures the extent to which any node lies close to all other nodes [1, p. 88]. An informative mapping was obtained within $30$ outer iterations. For clarity and emphasis of the node positions, edges were not included in the visualization. Drawings of graphs as large as the autonomous systems within the Internet typically thin out most of the edges.

Figure 4 shows the embedding with color coding reflecting variations in centrality measure. The proposed approach based on first-order methods leads to a fast algorithm for visualizing such large networks.

## 6. CONCLUSIONS

In this work, MDS-based means of embedding graphs with certain structural constraints were proposed. In particular, an optimization problem was formulated under centrality constraints that are used to capture relative levels of importance between the nodes. A block coordinate descent solver with successive approximations was developed to deal with the non-convexity and non-smoothness of the constrained MDS stress minimization problem. In addition, a smoothness penalty term was incorporated to minimize the edge crossings in the resultant network visualizations. Tests on real-world networks were run and the results demonstrated that convergence is guaranteed, and large networks can be visualized relatively fast.

## 7. REFERENCES

[1] E. D. Kolaczyk, *Statistical Analysis of Network Data: Methods and Models*, Springer, New York, NY, USA, 2009.

[2] I. Borg and P. J. F. Groenen, *Modern Multidimensional Scaling: Theory and Applications*, Springer, New York, NY, 2005.

[3] U. Brandes and C. Pich, "More flexible radial layout," *J. of Graph Algorithms and Apps.*, vol. 15, pp. 157–173, Feb. 2011.

[4] J. Alvarez-Hamelin, L. Dall'Asta, A. Barrat, and A. Vespignani, "Large scale networks fingerprinting and visualization using the k-core decomposition," *Advances in Neural Info. Proc. Sys.*, vol. 18, pp. 41–50, May 2006.

[5] F. Fouss, A. Pirotte, J. M. Renders, and M. Saerens, "Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation," *IEEE Trans. on Knowledge and Data Eng.*, vol. 19, pp. 355–369, March 2007.

[6] A. Buja, D. F. Swayne, M. L. Littman, N. Dean, H. Hofmann, and L. Chen, "Data visualization with multidimensional scaling," *J. of Comp. and Graph. Stats.*, pp. 444–472, June 2008.

[7] D. P. Bertsekas, *Nonlinear programming*, Athena Scientific, Belmont, MA, 1999.

[8] M. Razaviyayn, M. Hong, and Z. Q. Luo, "A unified convergence analysis of block successive minimization methods for nonsmooth optimization," *arXiv preprint arXiv:1209.2385v1*, Sept. 2012.

[9] J. Leskovec, J. Kleinberg, and C. Faloutsos, "Graph evolution: Densification and shrinking diameters," *ACM Trans. on Knowledge Discovery from Data*, vol. 1, article 2, March 2007.

[10] L. C. Freeman, "A set of measures of centrality based on betweenness," *Sociometry*, vol. 40, pp. 35–41, March 1977.