0.02 € on Embedding

W. Andreas Svrcek-Seiler

Institute for Theoretical Chemistry and Structural Biology
University Vienna
http://www.tbi.univie.ac.at/~svrci/

Bled, Feb. 2004

Definition (from `http://mathworld.wolfram.com`):

“An embedding is a representation of a topological object, manifold, graph, field, etc. in a certain space in such a way that its connectivity or algebraic properties are preserved.”

In the following, we will be concerned with a special instance of embedding, i.e. embedding a set of distances in Euclidean space $\mathbb{R}^N$.

Why would would one do that ?

- Graph embedding (with edge weigts interpreted as distances).
- Determination of molecular structure from distance information obtained from NMR measurements.
- Fun.
- $\cdots$

How could one do that,
given “enough” distances are available ?

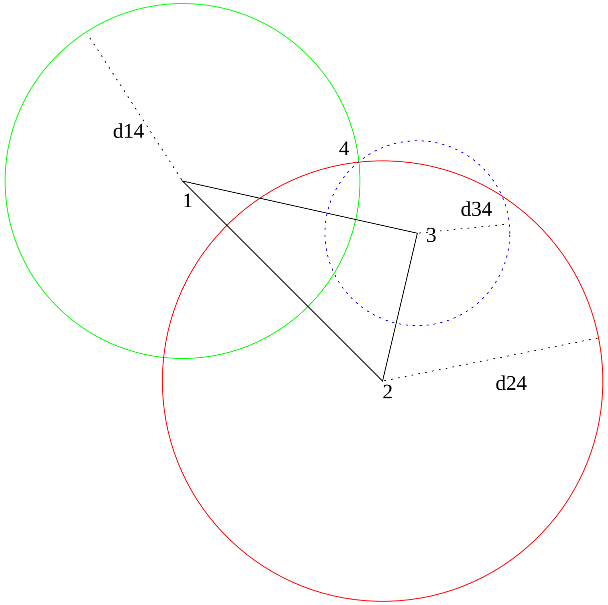
(a) By “direct construction”.

(b) From the metric matrix.

(c) By Stochastic Proximity Embedding.

(d)-(z) ...various other embedding schemes exist.

Direct Construction...



...shown here for 2 dimensions, is trivial and could be done using a pencil, a ruler and compasses. On a computer it is an $O(N)$-type procedure.

Direct construction is also quite trivial in three dimensions, but I mention it since this method was recently "advertised":
Q. Deng, Z. Wu. A linear-time algorithm for solving for solving the molecular distance geometry problem with exact interatomic distances. *J. Global Optim.* **22**, 365-375, 2002 (!)

Metric Matrix embedding:

Some semi-straightforward math allows to calculate distance d_{i0} of each point $\vec{r}_i$ to the centroid $\vec{r}_0$ of all points:

$$d_{i0}^2 = \frac{1}{N} \sum_{j=1}^{N} d_{ij}^2 - \frac{1}{N^2} \sum_{j<k}^{N} d_{jk}^2$$

Then the so called metric matrix $G_{ij} = \vec{r}_i \cdot \vec{r}_j$ can be obtained by applying the law of cosines.

Obviously , G_{ij} could also be written as

$$G = \begin{pmatrix} x_1 & y_1 & z_1 & 0 & \dots \\ x_2 & y_2 & z_2 & 0 & \dots \\ x_3 & y_3 & z_3 & 0 & \dots \\ . & . & . & 0 & \dots \\ . & . & . & 0 & \dots \\ x_n & y_n & z_n & 0 & \dots \end{pmatrix} \cdot \begin{pmatrix} x_1 & x_2 & x_3 & \dots & x_n \\ y_1 & y_2 & y_3 & \dots & x_n \\ z_1 & z_2 & z_3 & \dots & x_n \\ 0 & 0 & 0 & \dots & 0 \\ . & . & . & \dots & 0 \\ . & . & . & \dots & 0 \end{pmatrix}$$

So the “square root” of the metric matrix G (obtained by diagonalization) contains the coordinates. The associated computational cost is $O(N^3)$.

Stochastic proximity embedding

D. K. Agrafiotis *J. Comp. Chem.* **24**, 10, 1215-1221,2003.

Procedure: Let x_i be the coordinates, d_{ij} the current distance between points i and j and r_{ij} their target distance. Besides, let ϵ have its usual meaning.

1 Initialize the coordinates (e.g. randomly).

2 Randomly select a pair of points i and j and update their coordinates by:

$$\begin{aligned} \vec{x}_i &\leftarrow \vec{x}_i + \frac{\lambda}{2}\frac{r_{ij} - d_{ij}}{d_{ij} + \epsilon}(\vec{x}_i - \vec{x}_j) \\ \vec{x}_j &\leftarrow \vec{x}_j + \frac{\lambda}{2}\frac{r_{ij} - d_{ij}}{d_{ij} + \epsilon}(\vec{x}_j - \vec{x}_i) \end{aligned}$$

3 Repeat step 2 for a prescribed number of steps S.

4 Decrease the “learning rate” λ by a prescribed decrement $\delta\lambda$.

5 Repeat steps 2 to 4 for a prescribed number of cycles C.

Another look at the correction term:

$$\Delta\vec{x}_i = \frac{\lambda}{2}\frac{r_{ij} - d_{ij}}{d_{ij} + \epsilon}(\vec{x}_i - \vec{x}_j)$$

With $\lambda = 1$, the correction corresponds to the gradient of the penalty ij^{th} contribution to the penalty function.

$$S = \sum_{j>i}^{N}(d_{ij} - r_{ij})^2$$

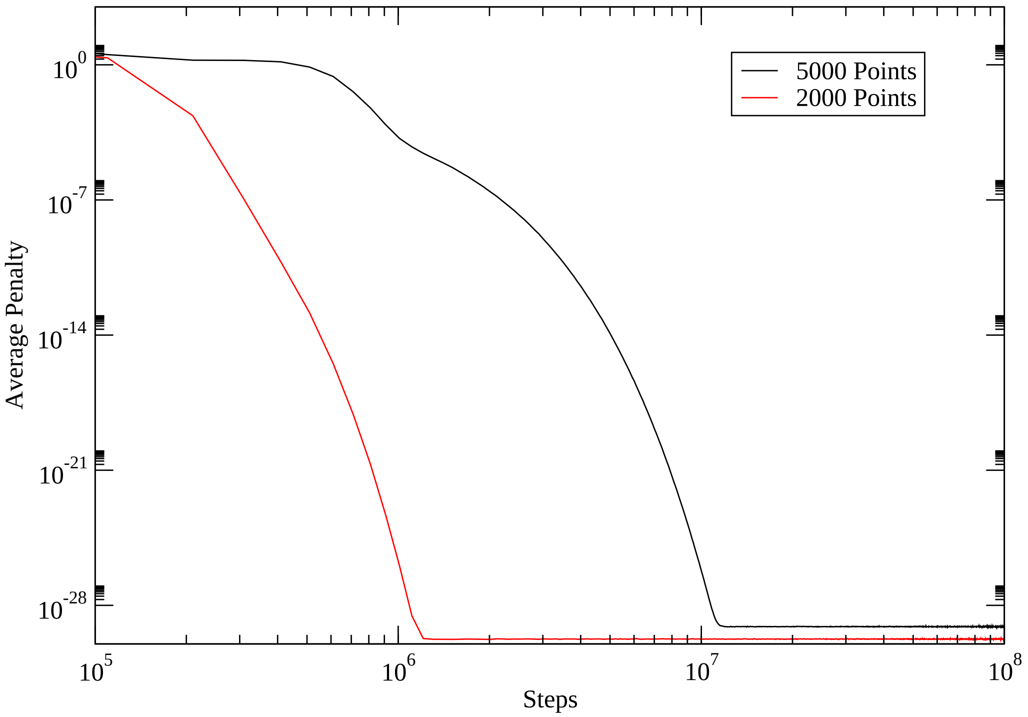
Furthermore $\lambda = 1$ implies that each chosen pair of points is immediately set to the desired distance.

Technical sidenote: ϵ can be left out, e.g. by ‘‘`if (dij ==0.0) continue;`’’

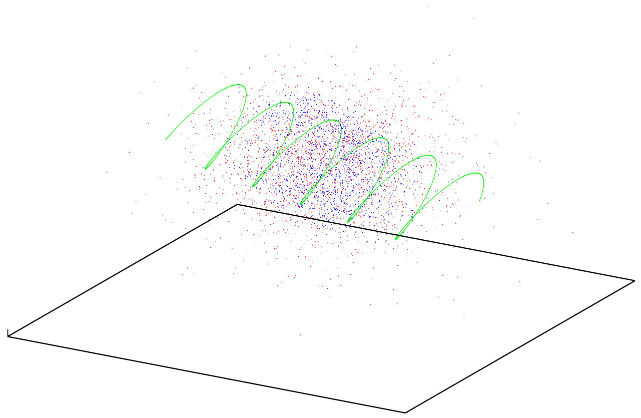
Obvious (?) features of the algorithm:

It is very simple to implement for arbitrary dimension.
For a “sufficient” set of distances, $\lambda = 1$ leads to convergence.
Overall chirality is truly random, in contrast to the metric matrix approach, where it is arbitrary, but not random.
When embedding a set of distances in an Euclidean space of “too low” dimensionality, decreasing λ as given by Agrafiotis is advisable.

Computational experiments....
a) Behavior of the penalty function $S(\#\text{steps})$

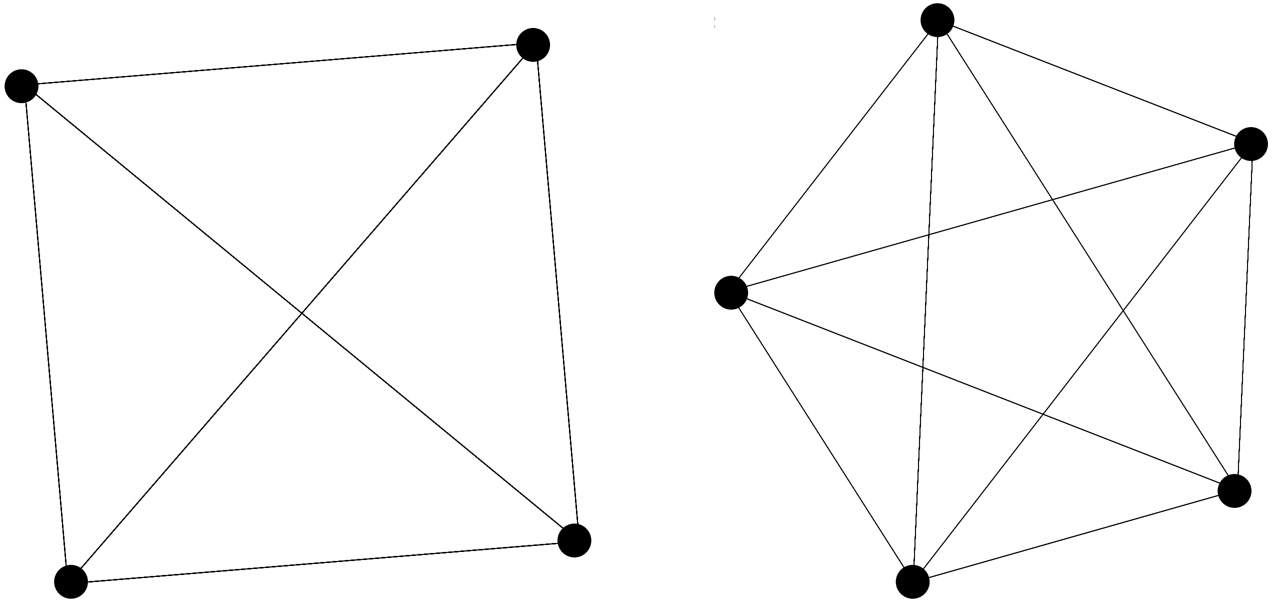


b) A graphical impression of the progress:



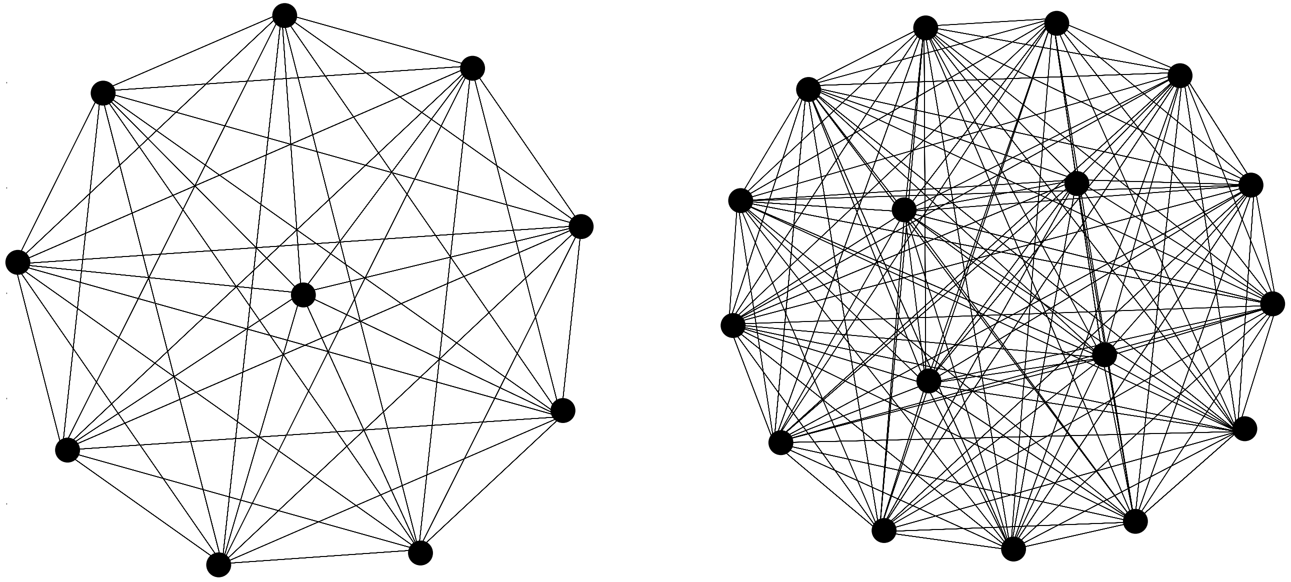
Spiral consisting of 2000 points, after 10^5 (red), 10^6 (blue) and 10^7 (green) iterations.

c) Some nice graph representations...



Completely connected graphs with 4 and 5 vertices (embeddable in 3 and 4 dimensions, respectively).

Further examples:



Completely connected graphs with 10 and 17 vertices.

Conclusions (pro)

The stochastic proximity embedding algorithm is easy to implement and reasonably fast.
Without any mathematical rigor one might state that it allows "nice" representations of *some* graphs, especially if they are highly connected.

Conclusions (con)

Choosing the number of iterations is based on educated guess and/or numerical experiment.

Once one is reasonably close to the target configuration, conjugate gradient minimization of the applied penalty function should lead to faster convergence.

The chirality of embedded three-dimensional structures is random.

Thank you for your attention !

...Any contributions, especially to the chirality problem, are welcome !