NiceGraph Program and its Applications in Chemistry

Tomaž Pisanski, Bor Plestenjak

IMFM / TCS, University of Ljubljana, Jadranska c. 19,
SI-61111 Ljubljana, Slovenia

and

Ante Graovac[a]

MARS, Texas A&M University at Galveston, Galveston,
Texas 77553-1675, USA

Recently, the problem of drawing graphs has become a hot subject in mathematical and computer sciences. In the present paper, two of the graph drawing algorithms, namely those of Kamada-Kawai and Fruchterman-Reingold, are for the first time applied to chemistry in their original two dimensional (2D) versions as well as in their generalized three dimensional (3D) version developed by us. In addition, the algorithm based on the adjacency matrix eigenvectors has been also tested.

All three algorithms in their 2D and 3D versions have been tested on a series of molecules, especially on fullerenes and toroidal pure carbon cages, the so-called torusenes. The conformations obtained offer a rather good guess of starting geometries for more sophisticated methods. The drawings obtained by the Fruchterman-Reingold algorithm are superior to those generated by the Kamada-Kawai algorithm.

In addition, all molecular graphs studied have also been represented by the so-called Schlegel diagrams for whose generation a novel algorithm was developed. Schlegel diagrams are important for identifying and analyzing the topological properties of large spatial carbon clusters.

INTRODUCTION

When representing molecules with graphs, the individuality of constituent atoms and the character of chemical bonds is mostly suppressed but the connectivity is emphasised in its pure form.

---

[a] Permanent address: The R. Bošković Institute, HR-41001 Zagreb, POB 1016, Croatia

Atoms are represented by vertices of graphs, which are conveniently depicted as points, and chemical bonds by edges which are drawn as straight lines. As the positions of points are completely arbitrary, there is an infinite number of ways to draw a given graph.

However, by imposing some aesthetic or other criterion, the admissible number of ways of drawing a graph is reduced. Recently the problem of drawing graphs became a hot subject in mathematical and computer sciences. A series of international conferences on the subject have been organized annually since 1992.[1] This topic has important applications in key computer technologies, such as software engineering, database design and visual interfaces. Further applications can be found in architectural and circuit design, project management, in mathematical fields such as computational geometry, topological graph theory, ordered sets, and many others. However, graph drawing algorithms have been developed primarily for presentations of graphs in plane.

In the present paper, two of the graph drawing algorithms[2,3] are for the first time applied to chemistry in their original 2D version as well as in their generalized three dimensional version developed by us. Also, the algorithm based on the adjacency matrix eigenvectors has been tested. Moreover, the algorithm to present the so-called Schlegel diagram of a polyhedron is developed.

All the above algorithms belong to the class of the so-called spring embedding algorithms.[4] They are all descendants of Eades' algorithm.[5] Since all of them give aesthetically acceptable drawings, we name them NiceGraph models. The methods have been successfully implemented and form a part of the package Vega developed at the IMFM/TCS in Ljubljana.

The methods developed here are not a substitute for more advanced quantum-chemical methods but they rather offer a noble guess of plausible starting geometries for more sophisticated methods. As they are simple to apply, we recommend them for quick determination of moleculear geometries, especially when many isomers of a given molecule have to be searched. This is, for instance, the case of fullerene molecules which are the object of intensive current research in chemistry, physics and material sciences.[6]

THE ALGORITHM OF KAMADA AND KAWAI

Let $G = (V,E)$ be a graph with $n = |V|$ vertices and $m = |E|$ edges. The graph-theoretical distance d_{ij} between vertices v_i and v_j is the smallest number of edges between v_i and v_j. Distances d_{ij} are integers ranging from 1 for the first neighbours up to diameter D, the largest distance in a graph.

Let us make a drawing of graph G in three dimensional Euclidean space. To each vertex v_i, a point $\vec{r_i} = (x_i, y_i, z_i)$ is asociated in 3D-space. The Euclidean distance $D_{ij} = |\vec{r_i} - \vec{r_j}|$ is associated with a pair of vertices v_i and v_j. The quantity $(D_{ij} - d_{ij})^2$ measures the deviation of the Euclidean from the graph-theoretical distance. Overall deviation of the Euclidean with respect to graph-theoretical distances in a graph G is given by

$$E = E\left(\vec{r_1}, \vec{r_2}, \ldots, \vec{r_n}; G\right) = \sum \frac{1}{2} k_{ij} (D_{ij} - d_{ij})^2$$

where summation goes over all possible pairs of vertices.

According to Kamada and Kawai, a graph is modelled as a system of balls and springs. If so, the energy function E could be understood as the elastic energy associated with a particular drawing $(\vec{r_1}, \vec{r_2}, \ldots, \vec{r_n})$ of graph G. The contribution $k_{ij}(D_{ij} - d_{ij})^2/2$ is the elastic energy of a spring of the force constant k where d plays the role of the equilibrium distance.

The problem of graph drawing is reduced to finding positions of vertices (balls) in such a way that the energy of a system of springs becomes minimal.

It is reasonable to assume that k_{ij} are inversely proportional to d_{ij}:

$$k_{ij} = K/d_{ij}$$

where K is an arbitrary positive constant.

The minimum of E is determined by:

$$\frac{\partial E}{\partial x_i} = 0, \quad \frac{\partial E}{\partial y_i} = 0, \quad \frac{\partial E}{\partial z_i} = 0, \quad i = 1, 2, \ldots, n$$

i.e. by the system of $3n$ non-linear equations. We solve these equations numerically. For each vertex v_i the quantity

$$\Delta_i = \sqrt{\frac{\partial E^2}{\partial x_i} + \frac{\partial E^2}{\partial y_i} + \frac{\partial E^2}{\partial z_i}}$$

is calculated.

In each step of the algorithm, the vertex v_i with the maximum Δ_i is chosen and E is considered as the function of only three variables x_i, y_i and z_i. By applying the Newton method, the related equations become linear and we have to solve the linear system in three variables until Δ_i becomes less than some threshold value ε. After that, the next maximal Δ_i is found and the procedure is repeated. In other words, only one point is moved at each iteration.

By omitting the third coordinate z, the two dimensional version of the algorithm is derived. Indeed, the 2D version was first developed. The 2D and 3D versions of the Kamada and Kawai algorithm will be denoted in further text by KK2 and KK3, respectively.

Time complexity for the calculation of the graph-theoretical distances is $\mathcal{O}(n^3)$. In the first step of the algorithm, all Δ_i have to be calculated. Time complexity for this is $\mathcal{O}(n^2)$. Time complexity for each following step is $\mathcal{O}(n)$ since values Δ_i only have to be updated.

THE ALGORITHM OF FRUCHTERMAN AND REINGOLD

In this model, the graph is again modelled as a physical system. Repulsive forces are calculated between each pair of vertices and attractive forces are calculated between each pair of adjacent vertices. Forces are used to calculate velocity for every time quantum (instead of acceleration as it is usual in physical systems). The aim of the algorithm is to find the static equilibria, *i.e.* the state with zero resultant forces for all vertices. The 2D and 3D versions of the Fruchterman and Reingold algorithm will be denoted in further text by FR2 and FR3, respectively.

If d is the distance between the two vertices, then the attractive force f_a is calculated as

$$f_a = d^2/k$$

and the repulsive force f_r is calculated as

$$f_r = k^2/d.$$

Here, k denotes the optimal distance between the vertices calculated at the beginning of the algorithm as

$$k = C \sqrt{\text{area/number of vertice}}$$

where the constant C is found experimentally. It is easy to see that for the path of length two, k is the distance where the forces would cancel out each other while in a general graph, k is the average distance where the resultant forces for all vertices equal zero.

In each step of the algorithm, the resultant forces of all vertices are calculated and all vertices are moved in the directions of the resultant forces. The displacements are controlled by the temperature parameter T and by the borders of the area. Vertices cannot be displaced outside the frame and the size of displacement δ_v of vertex v with the resultant force F_v is

$$\delta_v = \min(T, |F_v|) .$$

In each step, the temperature is reduced using some cooling function.

One step of the algorithm has time complexity $\mathcal{O}(n^2 + m)$.

THE ALGORITHM FOR DRAWING SCHLEGEL DIAGRAMS

Schlegel diagrams are planar representations of polyhedral graphs and, therefore, they are suitable for representing fullerenes. These diagrams clearly show the connectivity of atoms in fullerenes.

A graph can be modelled as a system of balls and elastic bands where the vertices of the outer face are fixed on a regular polygon. We seek for the state of static equilibria. This algorithm is derived from the Fruchterman and Reingold algorithm by deleting all repulsive forces and fixing the vertices of an outer face.

Some further modifications were made in order to obtain better figures. If we want to end with approximately equally arranged faces, bands close to the peripheral ring should be stronger than the bands in the middle of the figure. Otherwise, we end with a large number of crowded small faces in the middle and large faces on the border of the figure. For this purpose, *periphericity* p_v, of a vertex v is introduced as the length of the shortest path between the vertex and the outer face.

The size of the attractive force between vertices u and v is calculated as

$$f_a = d^2 \exp\left(A \, \frac{2p_{\max} - p_u - p_v}{p_{\max}} \right)$$

where p_{max} is the maximum periphericity in the graph and constant A is found experimentally. Like in previous algorithms FR2 and FR3, the resultant forces of all vertices are calculated in each step and the vertices are displaced in the directions of the resultant forces for an amount equal to the minimum of the force and temperature T.

The algorithm can be applied to non-planar graphs as long as the peripheral ring is specified. In such cases, this algorithm often provides a better insight into the connectivity of vertices.

The Schlegel diagram algorithm will be denoted in further text by SCH. One step of the algorithm has time complexity $\mathcal{O}(m)$.

THE ALGORITHM BASED
ON THE ADJACENCY MATRIX EIGENVECTORS

Another model being accepted by some fullerene research groups[7–9] is based on the consideration of the eigenvectors $\vec{x_1}, \vec{x_2}, \ldots, \vec{x_n}$, of the adjacency matrix A of a graph with n vertices. Eigenvalues are ordered as: $\lambda_1 \geq \lambda_2 \geq \ldots$. Three consecutive eigenvectors, like $\vec{x_2}, \vec{x_3}$ and $\vec{x_4}$ are taken to build $n \times 3$ matrix $B = [\vec{x_2}, \vec{x_3}, \vec{x_4}]$. By reading the i-th row of B, (x_{2i}, x_{3i}, x_{4i}) as the 3D-coordinates of vertex i and running over all i's, in many cases a rather decent 3D-drawing of G is achieved. In the case when a graph is not regular, the results are generally better if the Laplacean matrix of a graph is taken instead of its adjacency matrix.[10]

Adjacency matrix eigenvectors algorithm will be abbreviated AME in further text. Standard numerical algorithms that are available for Mathematica users were employed (Eigensystem, GramSchmidt).

RESULTS

The basic information on a graph is the connectivity of its vertices. Here, it is given by the adjacency matrix of a graph from which the graph-theoretical distances are calculated.

The starting configuration of points in Euclidean space, if not otherwise specified, is given at random. It could also be drawn on the screen by using graphical interface or given by some rule. For instance, graph G of the snub cube could be depicted[11] as a rotagraph $\omega_4(C_6 + \{\{2,6\},\{3,5\},\{1,1\},\{1,2\},\{6,2\},\{6,3\},\{5,3\},\{5,4\},\{4,4\}\})$, *i.e.* by repeating four times the cycle of length 6 with the connectivity between neighbouring paths given as in Figure 1. Its SCH, AME, KK2, KK3, FR2 and FR3 drawings are depicted in Figure 2.
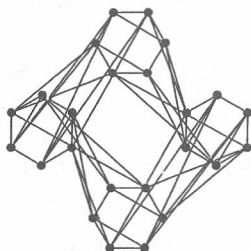

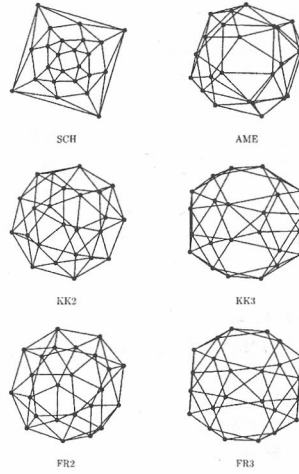Figure 1. The graph of the snub cube represented as a rotgraph.

Figure 2. The Schlegel (SCH), adjacency matrix eigenvectors (AME), Kamada-Kawai 2D (KK2) and 3D (KK3), and Fruchterman-Reingold 2D (FR2) and 3D (FR3) NiceGraph drawings of the snub cube graph.

Note that the drawings obtained by the optimization in 2D-space look like 2-dimensional projections of optimized 3D-drawings. The same also applies to other graphs.

The starting Schlegel diagram of the celebrated icosahedral $I_h : C_{60}$ fullerene and its NiceGraph drawings are depicted in Figure 3.
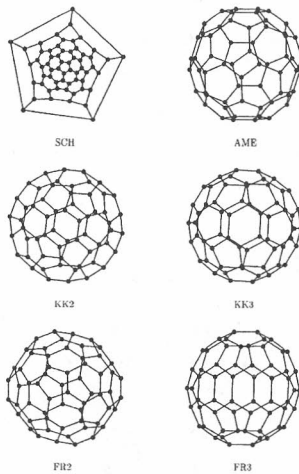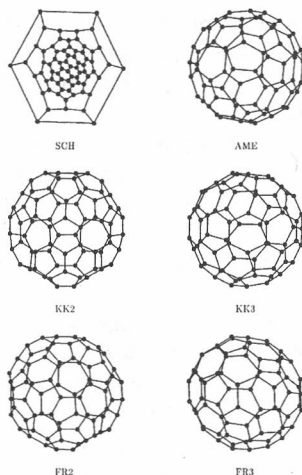

Figure 3. The Schlegel (SCH), adjacency matrix eigenvectors (AME), Kamada-Kawai 2D (KK2) and 3D (KK3), and Fruchterman-Reingold 2D (FR2) and 3D (FR3) NiceGraph drawings of the buckyball ($I_h : C_{60}$) graph.

Figure 4. The Schlegel (SCH), adjacency matrix eigenvectors (AME), Kamada-Kawai 2D (KK2) and 3D (KK3), and Fruchterman-Reingold 2D (FR2) and 3D (FR3) NiceGraph drawings of the C_{72} fullerene, a leapfrog of C_{24}.

The Schlegel diagram and the NiceGraph drawings of the C_{72} fullerene, a leap-frog[12] of the C_{24} cage, are shown in Figure 4.

The example depicted in Figure 5 represents Schlegel and NiceGraph configurations of the C_{28} cage (of T_d symmetry) which is the smallest fullerene so far[13] to form in substantial abundance.
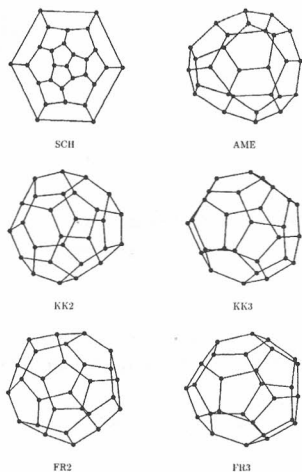

Figure 5. The Schlegel (SCH), adjacency matrix eigenvectors (AME), Kamada-Kawai 2D (KK2) and 3D (KK3), and Fruchterman-Reingold 2D (FR2) and 3D (FR3) NiceGraph drawings of the $T_d : C_{28}$ fullerene.

This cage behaves as a tetravalent species, trapping a tetravalent atom inside the cage to make endohedral fullerenes such as $Ti@C_{28}, U@C_{28}$, *etc*. Its tetravalence is exhibited also by reacting at the four tetrahedral vertices on the outside of the T_d: C_{28} cage to form *e.g.* $C_{28}H_4$.

Formally, the fullerenes are defined as 3-valent (3-regular) graphs having only pentagons and hexagons as faces. If the use of heptagons is allowed too, Euler's polyhedron formula gives:

$$h_5 = h_7 + 12(1 - g)$$

where h_5 and h_7 denote the number of pentagons and heptagons, respectively. The number of hexagons, h_6, is arbitrary. g denotes the genus of the surface at which G is embedded, *i.e.* g = 0, 1, and 2, for the sphere, torus and pretzel, respectively. The toroidal, pure carbon cages have recently received considerable attention.[15,16]

The above formula for h_7 = 0, g = 0 gives h_5 = 12, *i.e.* exactly 12 pentagons are needed to give a spherical fullerene. For a torus (g = 1), one has:

$$h_5 = h_7.$$

Two cases are possible. The first, in which $h_5 = h_7 = 0$, gives toroidal polyhexes, *i.e.* the toroidal fullerenes composed solely from hexagons. It is easy to see that: $3n = 2m = 6h_6$. The FR3 NiceGraph drawing of the toroidal polyhex with h_6 = 120 is depicted in Figure 6a.

The second case, $h_5 = h_7 \neq 0$, gives the azulenoid cages. The FR3 NiceGraph drawing of the torus with $h_5 = h_7 = 10$ (h_6 = 100) is depicted in Figure 6b.

(a)   (b)

Figure 6. Fruchterman-Reingold 3D (FR3) NiceGraph drawings of (a) the toroidal polyhex with 120 hexagons and (b) the azelunoid cage with 10 pentagons, 100 hexagons and 10 heptagons.

In the above cases, it is evident that the NiceGraph algorithms are capable of recognizing the genus of the surface in which the graph has to be embedded.

DISCUSSION

The algorithms presented in this paper give reasonably plausible geometries of fullerenes and other molecular graphs. The drawings obtained by the Fruchterman-Reingold algorithm are superior to those generated by the Kamada-Kawai algorithm.

However, the final NiceGraph configurations are not necessarilly »nice«, *i.e.* the final drawings do not necessarily exhibit the geometrical symmetry indicated by the automorphism group of a graph. This is especially true of graphs with a large number of vertices. One possible way to remedy this deficiency would be to make the threshold values in computations smaller, but this would increase computation time. Another approach, which is now under development, is to use the knowledge of the automorphism group of a graph in building up the proper geometrical symmetry of the final drawing.[14]

All algorithms presented here are written in Turbo Pascal and implemented in a system for manipulating discrete mathematical structures Vega.[17]

The NiceGraph program in its present form does not discriminate between the individuality of atoms and chemical bonds. However, the results for the $C_{28}H_4$ cage show that its plausible geometry is nevertheless achieved. The example of the biphenylene molecule is instructive too. Its KK NiceGraph drawings, where carbon and hydrogen atoms are treated on an equal footing, rotate one ring with respect to another but by an incorrect angle of $90°$.

Future developments of the program in which the individuality of various chemical bonds will be reflected by appropriate changes in graph-theoretical distances are highly desirable. It would be especially interesting to study the changes in the geometry of fullerenes in which a number of carbons is substituted by boron, nitrogen, and other atoms.

A systematic comparison of geometries obtained by the mathematical models like the NiceGraph, the eigenvectors of the adjacency matrix, and more realistic, physically based models like molecular mechanics, semiempirical and *ab initio* quantum-chemical models as well as with the experimental data is under way by the present authors.[18]

CONCLUSIONS

The NiceGraph model, proposed here for applications in chemistry, is easy to apply. The only input data are the connectivity tables of a molecule and the starting coordinates of atoms are taken at random. The final NiceGraph configuration is aesthetically acceptable and represents a plausible starting geometry for applying some more realistic model, like molecular mechanics. The present model is especially recommendable when a large number of isomers have to be searched, as it is the case of fullerenes.

REFERENCES

1. *Graph Drawing '93, Proceedings of the ALCOM International Workshop on Graph Drawing*, G. Di Battista, P. Eades, H. de Fraysseix, P. Rosenstiehl, and R. Tamassia (Eds.), Sevres, France, 1993.
2. T. Fruchterman and E. Reingold, *Software-Practice and Experience* **21** (1991) p. 1129.
3. T. Kamada and S. Kawai, *Information Processing Letters* **31** (1989).

4. G. Di Battista, P. Eades, R. Tamassia, and I. G. Tollis, *Algorithms for Drawing Graphs: An Annotated Bibliography*, (1993). Available *via* anonymous ftp from wilma.cs.brown.edu,files pub/paper/compgeo/gdbiblio.tex.z and pub/paper/compgeo/gdbiblio.ps.z.
5. P. Eades, *Congressus Numerantium* **42** (1984) 149.
6. See *e.g.*: H. W. Kroto, W. Allaf, and S. P. Balm, *Chem. Rev.* **91** (1991) 1213; H. Kroto, *Fullerene Science and Technology* **2** (1994) 333.
7. D. E. Manolopoulos and P. W. Fowler, *J. Chem. Phys.* **96** (1992) 7603.
8. J. Shawe-Taylor and T. Pisanski, *Characterizing graph drawing with eigenvectors* Technical Report CSD-TR-93-20 Royal Holloway, University of London, Department of Computer Science, Egham, Surrey TW20 OEX, England, submitted.
9. C. D. Godsil, *Algebraic Combinatorics*, Chapman and Hall, 1993.
10. P. W. Fowler, T. Pisanski, and J. Shawe-Taylor in: *Graph Drawing, Lecture Notes in Computer Science,* Springer Verlag, Berlin, R. Tamassia and I. G. Tollis (Eds.), **894** (1995) 282.
11. T. Pisanski, A. Žitnik, A. Graovac, and A. Baumgartner, *J. Chem. Inf. Comp. Sci.* **34** (1994) 1090.
12. P. Fowler and T. Pisanski, *J. Chem. Soc. Faraday Trans.* **90** (1994) 2865.
13. Ting Guo *et al.*, *Science* **257** (1992) 1661.
14. M. Kaufman, M. Sc. Thesis, University of Ljubljana (1994); M. Kaufman *et al.* in preparation.
15. E. C. Kirby, *Fullerene Science and Technology* **2** (1994) 395.
16. B,. Borštnik and D. Lukman, *Chem. Phys. Letters* **228** (1994) 312.
17. T. Pisanski *et al.*, Vega: *System for manipulating discrete mathematical structures*, Available *via* anonymous ftp from ftp.mat.uni-lj.si.
18. B. Borštnik, A. Graovac, M. Kaufman, D. Lukman, T. Pisanski, B. Plestenjak, and A. Žitnik, in preparation.

SAŽETAK

NiceGraph program i njegove primjene u kemiji

Tomaž Pisanski, Bor Plestenjak i Ante Graovac

Problem risanja grafova postaje uzbudljiva tema matematike i računarskih znanosti. U ovom radu se po prvi puta u kemiji primjenjuju dva postojeća algoritma za risanje grafova, naime, algoritmi Kamada-Kawaija i Fruchterman-Reingolda, i to u njihovu izvornom dvodimenzijskom (2D), i u od nas poobćenom trodimenzijskom (3D) obliku. U radu je dalje testiran već poznati algoritam zasnovan na vlastitim vektorima matrice susjedstva.

Sva tri algoritma u njihovim 2D i 3D verzijama su testirana na seriji molekula, posebno na fullerenima i toroidnim čisto ugljikovim kavezima. Dobivene konformacije predstavljaju prilično dobre početne geometrije za točnije račune, s time da su crteži dobijeni Fruchterman-Reingoldovim algoritmom superiorni onima dobijenim algoritmom Kamada-Kawai.

Sve molekulne grafove prikazali smo također tzv. Schlegelovim dijagramima za čije smo generiranje razvili vlastiti algoritam.