Flow Diagrams

The goal of this excercise is to generate visualizations of so-called Flow Diagrams. Flow diagrams, also called Sankey diagrams or Grasmann diagrams, are characterized by datasets of the following form:

```
Fuel Cell System
 pipe from & to Exergy kW
    2    5    3    480.00
    4    3    2    520.00
    7    2    7    150.00
    6    2    6    140.00
    5    2    1    210.00
    3    1    3    190.00
    1    4    1      8.00
```

Not a single indication of any kind concerning the geometry of the diagram. Actually, little specific is described by these data. This is an advantage and a disadvantage at the same time (according to the greatest philosopher of the last century, Johan Cruijff ;-).
The Advantage:
- *You can do everything you want.*
The Disadvantage:
- *You don't know what to do.*
Yet you have to do something specific, so specific that even a computer can understand .. So we decided to do something even a physicist can understand.

Spring Embedder

The Graph Geometry Solver is based upon the so-called Spring Embedder, which belongs to the class of Force-Directed Techniques, as quoted from the "Graph Drawing Tutorial":

`http://www.cs.brown.edu/people/rt/papers/gd-tutorial/gd-constraints.pdf`

Some Pro's and Con's of the approach are, according to this reference:
Pro's:
- relatively simple to implement
- heuristic improvements easily added
- often able to detect and display symmetries
- works well in practice for small graphs with regular structure
Con's:
- limited constraint satisfaction capability
- few theoretical results on the quality of the drawings produced

The last quote is repeated:

- few theoretical results on the quality of the drawings produced
But ... there may be evidence of the contrary !
Sample Theorems (HdB):
- The sum of the edges around a (non-fixed) vertex, when seen as vectors pointed toward other vertices, is exactly equal to zero.
- All faces associated with the graph are convex, with exception of the regions where fixed points are.

Our mathematical model is almost completely found on page 72 of the Tutorial.
Imagine all Edges in the graph being replaced by (equal) elastic strings which behave according to Hooke's law:

$$F = -c.u \quad \Longrightarrow \quad E = \frac{1}{2}c.u^2$$

Then it is required that the total elastic energy E of the system as a whole should reach a minimum:

$$\frac{1}{2}\sum_k c_k \left[(x_i - x_j)^2 + (y_i - y_j)^2\right] = \text{minimum}$$

Here (i, j) are numbers of the vertices which are connected by edge (k). The quantity c_k may be interpreted as the *elasticity* of edge (k). The best choice so far is to take all elaticities equal (to unity).

Systems of equations can be obtained by partial differentiation to x_i and y_i resulting in:

$$\sum_k c_k(x_i - x_j) = 0 \quad \text{and} \quad \sum_k c_k(y_i - y_j) = 0$$

Apart from boundary conditions.
The linear equations system may be assembled and solved by a common Finite Element technique. The important thing is that the equations for (x) and for (y) become mutually uncoupled. This means that values in the X direction and values in the Y direction can be solved quite independently of each other.

Boundary Conditions must be imposed in order to arrive at meaningful results. In a connected (undirected) graph, the number (E) of edges is related to the number (N) of vertices, as follows:

$$N - 1 <= E <= N(N - 1)/2$$

Since the equations for X and Y are uncoupled, it is clear that the number of Unknowns equals the number of vertices and the number of Equations equals the number of edges. Hence, strictly spoken only One boundary condition is really needed; resulting in a highly degenerated graph where all coordinates are the same. So there is a need for more ingenuity.

Intuitively, boundary points are ”more distant” from the inner parts of the domain. In a discrete graph, we can mimick this behaviour by calculating the shortest (topological) distances from a given vertex to all other vertices.
This is implemented by *Floyd’s Algorithm*.
Now sort the vertices with their ”far away” measures as a key. Boundary points are then selected according to the rule: best candidates have greatest overall distance.
Place boundary vertices at boundary of a regular convex polygon and continue to do so until the graph is stretched enough to prevent vertices and edges from being messed up altogether.

Further improvement of the Graph’s layout is achieved by *permutations* of the boundary points in combination with minimization of a Cost Function.
The Cost Function is a superposition of the following components (so far):
- Number of Coincident Vertices
- Number of Vertices On Edges (careful : every edge is joined by vertices)
- Number of edges Crossing each other
- Number of vertices where flows are such that they canNot be Joined
- Number indicating the amount of flow
Directed Leftward
- Number attached to the Smallest Angle in the Graph
Given by comments in the Console Panel of the Demo:
- After each calculation of a new geometry
- After manipulation of layout by the user

Room for in- and outgoing Tubes

Lemma

If $A = sin(\alpha)$ and $B = sin(\beta)$ and $\phi = \alpha + \beta$ then:

$$sin^2(\phi) = A^2 + B^2 + 2.A.B.cos(\phi)$$

Proof

The following formula is considered to be well known:

$$cos(\alpha + \beta) = cos(\alpha).cos(\beta) - sin(\alpha).sin(\beta)$$

$$\Longrightarrow \quad cos(\alpha + \beta) = \sqrt{1 - sin^2(\alpha)}.\sqrt{1 - sin^2(\beta)} - sin(\alpha).sin(\beta)$$

$$\Longrightarrow \quad cos(\alpha + \beta) + sin(\alpha).sin(\beta) = \sqrt{1 - sin^2(\alpha)}.\sqrt{1 - sin^2(\beta)}$$

After sqaring both sides, and upon substitution of the above, we obtain:

$$[cos(\phi) + A.B]^2 = (1 - A^2).(1 - B^2)$$

$$\Longrightarrow \quad cos^2(\phi) + 2.A.B.cos(\phi) + A^2.B^2 = 1 - A^2 - B^2 + A^2.B^2$$

$$\Longrightarrow \quad A^2 + B^2 + 2.A.B.cos(\phi) = 1 - cos^2(\phi) = sin^2(\phi)$$

Herewith the Lemma is proven.

Application

Within our application, ϕ is the angle between two radii in a star around a vertex in a graph representing the Flow diagram. The magnitude of the flows will be represented by "tubes" or rectangles. The width of a rectangle (the thickness of a tube) will be forced to be proportional to (a square root function of) the flow going through it.
We seek to scale the thickness of the tubes in such a way that their walls do not intersect each other outside the circle which will be reserved for the star's central vertex. Let the radius of this "apparatus" be R and the thicknesses be given by D_a and D_b. Then we can substitute for each pair of adjacent edges:

$$R.sin(\alpha) = R.A = D_a \quad \text{and} \quad R.sin(\beta) = R.B = D_b$$

Now use the abovementioned Lemma:

$$(R.A)^2 + (R.B)^2 + 2.(R.A).(R.B).cos(\phi) = R^2.sin^2(\phi)$$

$$R^2.sin^2(\phi) = D_a^2 + D_b^2 + 2.cos(\phi).D_a.D_b$$

Conclusion:

$$R = \frac{\sqrt{D_a^2 + 2.cos(\phi).D_a.D_b + D_b^2}}{|sin(\phi)|}$$

A cosine-rule is recognized in the nominator. The angle between the lines connecting the central vertex to the verteces in a triangle attached to it is ϕ. Hence the algorithm for the nominator $\overline{AB}$ is given by:
- define a paralellogram with vectors directed along two neighbouring edges belonging to the same vertex;
- define the lengths of these vectors as D_a and D_b;
- calculate the length of the sum of these vectors, giving the number $\overline{AB}$, according to the cosine-rule.

Now all quantities are defined in the following quite simple formula:

$$R = \frac{\overline{AB}}{|sin(\phi)|}$$

This is called the "infamous Formula" in the computer program.
First, an overall scaling factor can be calculated with it, thus assuring that the width of the tubes does not exceed the greatest allowable size in a graph. The latter, in turn, is determined by the mutual distances between all of the geometrical elements: vertices and edges.
Second, radii R for each pair of edges emanating from a vertex are calculated. The maximum of these radii is appropriate for the "apparatus" which, in a flow diagram, is associated with a vertex, with tubes of varying thickness attached to it, thus assuring that these tubes do not intersect each other outside the apparatus.
What shall we do, however, when the denominator $|sin(\phi)|$ becomes zero ?
This can happen in two cases: $\phi = 0$ or $\phi = 180^o$. The first case is actually unavoidable. It occurs when the diagram is manipulated in such a way that two edges (almost) become alignant. Due to the overall scaling, this will lead to very thin tubes. The second case can be avoided, mathematically, by recognizing that any tube is limited in its size. It has no sense to let it prolongate outside the region which is occupied by the accompanying edge. A little geometry reveals that the singular situation is reached for obtuse angles ϕ , if $cos(\phi)$ becomes greater than minus the quotient D_a/D_b or D_b/D_a , whichever is the smallest of the two. When this is happening, we can replace the $sin(\phi)$ in the denominator by the other value. Having done this, only the special case remains where $D_a = D_b$ remains to be resolved.