

Graph Geometry Solver based upon so-called
Spring Embedder

Some Pro's and Con's of the approach,
according to "the Tutorial":

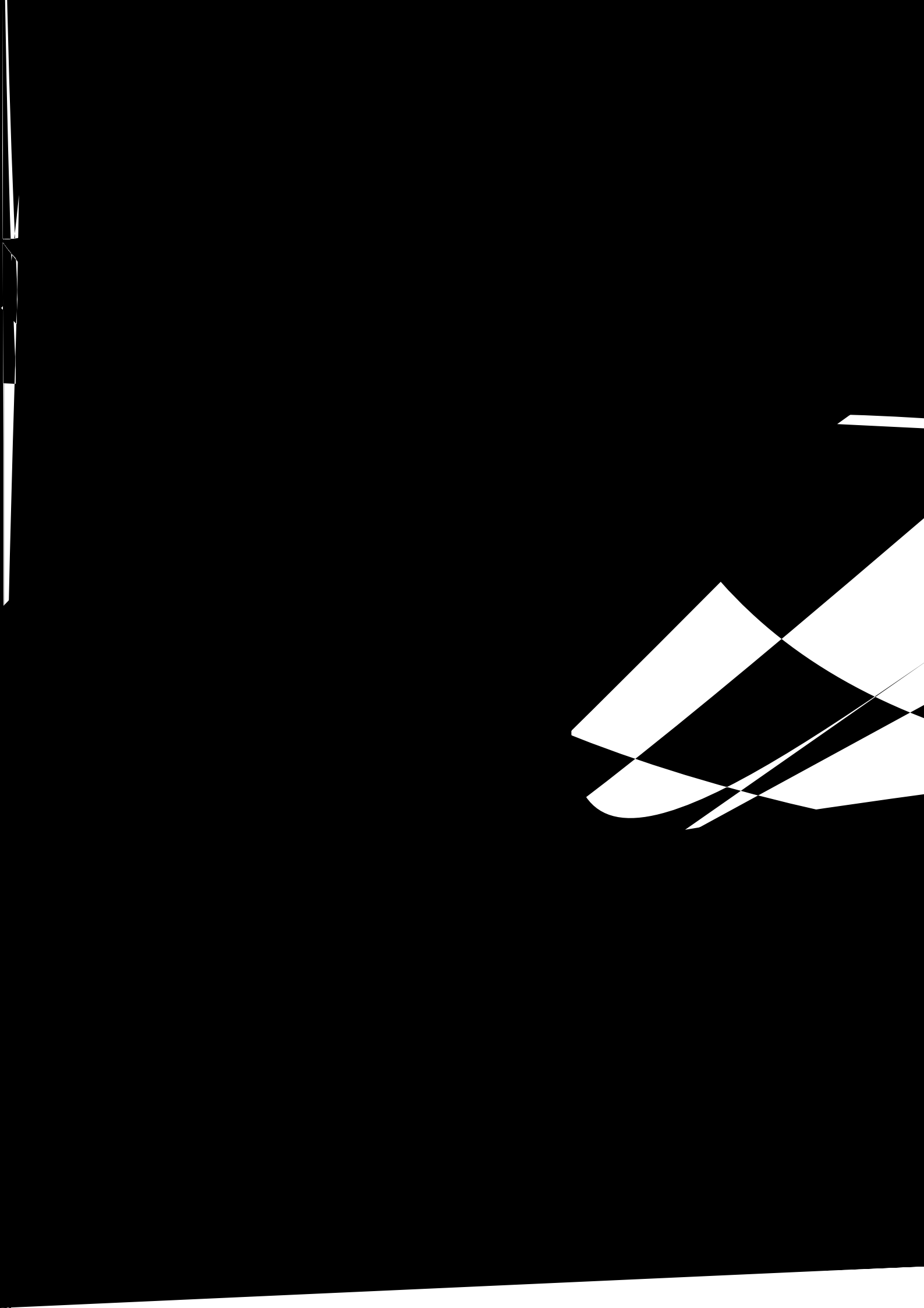
Pro's:

- relatively simple to implement
- heuristic improvements easily added
- often able to detect and display symmetries
- works well in practice for small graphs

Con's:

- limited constraint satisfaction capability
- few theoretical results on quality of drawings

HdB: the latter maybe Not true !



The background of the entire page is a complex, abstract geometric pattern. It consists of numerous overlapping triangles and lines in black and white, creating a sense of depth and movement. The pattern is dense and fills the entire frame, with some areas appearing more solid and others more transparent due to the layering of shapes.

ounda Conditions

Further improvement of graph layout by *permutation* of the boundary points.

Optimization

with Minimization of a Cost Function:

- Number of Coincident Vertices
- Number of Vertices "on" Edges
- Number of edges Crossing
- Number of unjoinable flows
- Amount of flow Directed Leftward
- Smallest Angle in the Graph

Given by comments in Console Panel of Demo:

- After each calculation of a new geometry
- After manipulation of layout by the user



ics
etry, wh
clipping

ation



Future Plan
Griding / Orthogonal analysis