Asociación Argentina de Mecánica Computacional

Mecánica Computacional Vol XXIX, págs. 6125-6135 (artículo completo)
Eduardo Dvorkin, Marcela Goldschmit, Mario Storti (Eds.)
Buenos Aires, Argentina, 15-18 Noviembre 2010

GRAPH LAYOUT ALGORITHMS FOR AUTOMATED SKETCHING OF LINKAGE MECHANISMS

Martín A. Pucheta[a], **Nicolás E. Ulrich**[b] **and Alberto Cardona**[a]

[a]*Centro Internacional de Métodos Computacionales en Ingeniería (CIMEC), INTEC (Universidad Nacional del Litoral-CONICET), Güemes 3450, S3000GLN Santa Fe, Argentina, e-mails: {mpucheta,acardona}@intec.unl.edu.ar, http://www.cimec.org.ar*

[b]*Facultad de Ingeniería y Ciencias Hídricas de la Universidad Nacional del Litoral, Centibecario , Ciudad Universitaria, Paraje "El Pozo", S3000 Santa Fe, Argentina, e-mail: nikolaseu@gmail.com, http://www.fich.unl.edu.ar*

Keywords: Graph layout, force-directed algorithm, combinatorics, linkage mechanisms, kinematic chains.

Abstract. The graph layout problem arises frequently in the conceptual stage of mechanism design, specially in the enumeration process where hundreds of topological solutions must be analyzed. Several combinatorial, heuristic (e.g. geometrical and genetic algorithms), and force-directed algorithms (based on spring models, self-organizing maps) have been used for layout of graphs of kinematic chains and linkage mechanisms. Two main objectives of graph layout are the avoidance of edge crossings and the aesthetics. Edge crossings cannot be always avoided by force-directed algorithms since they reach a minimum of the energy in dependence with the initial position of the vertices, often randomly generated. Since the number of vertices of graphs of mechanisms is low, combinatorial algorithms can be used to find an adequate initial position. In this paper, we present a combination of two algorithms: a combinatorial algorithm based on the exhaustive exploration of loop combinations with identification of all possible forms of drawing the graph as a base circle and successive arcs, followed by a force-directed algorithm based on spring repulsion and electrical attraction. The first algorithm is used to find an initial layout without edge crossings; then, the force-directed algorithm is used to achieve regularity on edges sizes and nodes distribution. Several available graphs of complex kinematic chains are used to validate the results. The layouts obtained have good quality in terms of minimization of edge crossings and maximization of aesthetic characteristics.

1 INTRODUCTION

The conceptual design is the earliest stage of linkage mechanism design where several alternatives are generated and evaluated in order to fulfill design requirements: functional, structural and design constraints. The best concepts are then selected for detailed design: detailed analysis and optimization, prototype construction and testing; the success of this second stage strongly depends on the quality of the selected concepts.

At conceptual design stage, many parts of the mechanisms have no dimensions and any potential mechanism solution has to be shown to the designer using a clear and comprehensive representation. The graph representation of mechanisms and Graph Theory algorithms can be used to computationally solve the enumeration of alternatives satisfying topological constraints of a given problem (Pucheta, 2008); often, the enumeration process leads to hundreds of topological solutions to be further analyzed either in visual form or automatically.

We have identified three needs of automated representation (Pucheta and Cardona, 2010b,a): (i) graph of the kinematic chain of a linkage mechanism, (ii) graph of a linkage mechanism, and (iii) graph of a linkage mechanism including parts to move with known coordinates. The first one can be solved using graph layout algorithms abundant from the computer science field (Fruchterman and Reingold, 1991; Hu, 2006; North, 2004), there exist two relevant approaches to solve the second one (Belfiore and Pennestri, 1994; Mauskar and Krishnamurty, 1996) but solutions for the third case are no reported.

The main objective of graph layout is to obtain a representation to ease the visualization and fast understanding of the graph. Two particular objectives are the avoidance of edge crossings and the aesthetics (good regularity on edges sizes and nodes distribution in the plane). A third objective useful for database representation and for repeated use of the topological structures, is to develop a *determinist algorithm*; any randomness in the algorithm must be eliminated in order to get the same layout for different executions of the algorithm over the same graph.

In the past, several graph layout algorithms have been used for layout of graphs and mechanisms and can be classified by their approach:

- Combinatorial algorithms which uses graph properties (Mauskar and Krishnamurty, 1996) and geometric constructions,
- Force-directed algorithms based on pseudo-physical models subject to attraction and repulsion forces (North, 2004; Fruchterman and Reingold, 1991; Hu, 2006), and
- Heuristic algorithms (e.g. genetic algorithms were recently proposed by Godoy (2007)),

nevertheless, none of these approaches are of general application.

In this paper, we present graph layout algorithms to automatically generate drawings of graphs and kinematic chains of linkage mechanisms. We use a combination of two algorithms: a combinatorial algorithm based on the exhaustive exploration of loop combinations with identification of all possible forms of drawing the graph as a base circle and successive arcs, followed by a force-directed algorithm based on spring repulsion and electrical attraction. The first algorithm is used to find an initial layout without edge crossings; then, the force-directed algorithm is used to achieve regularity on edges sizes and nodes distribution.

Several graphs of complex kinematic chains with one and two degrees-of-freedom are used to illustrate the algorithms, explain the methodology, and to validate the results.

2 GRAPH LAYOUT ALGORITHMS

In this section we review two existing graph layout algorithms and then propose an algorithm which takes the advantages of the first two.

2.1 Force-directed algorithms

The graph is modeled as a pseudo physical or mechanical system where vertices are connected by springs which produce attraction forces and there also exists repulsion forces between any pair of vertices. Note in Fig. 1 that the equilibrium between forces is achieved when the distance between vertices is k and also note that the contribution of repulsion forces when the distance equals $2k$ is low.

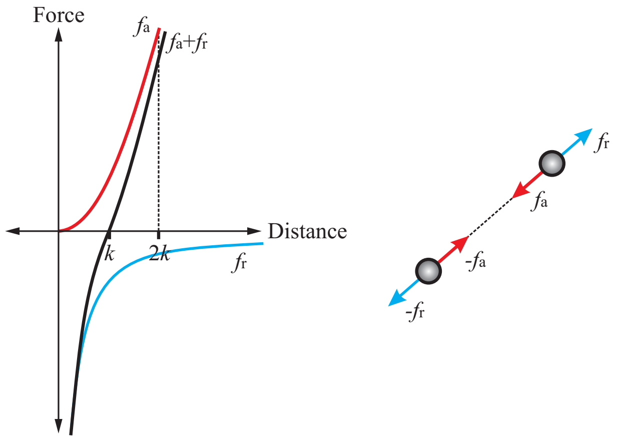

Figure 1: Repulsion and attraction force laws (Fruchterman and Reingold, 1991).

The force-directed algorithm develops several iterations until a stopping criteria over the energy is reached:

Step 1) Compute the direction of displacement for repulsion forces between any pair of nodes in the graph.

Step 2) Compute the direction of displacement for attraction forces.

Step 3) Using the direction of the resultant of repulsion and attraction forces apply a displacement to change the position of each vertex. The summation of each displacement is accumulated as the energy of the system.

In order to reduce the $\mathcal{O}(n^2)$ complexity of the repulsion forces, Fruchterman and Reingold (1991) proposed to assume the influence of vertices negligible if they are separated at a certain distance; in this way, repulsion forces at each vertex are computed only for vertices inside a given influence zone, for instance using a zone limited by a radius r equal to twice the desired length k of the edge.

The convergence rate can also be improved if the imposed amount of displacement is an adaptive function of the energy evolution: Hu (2006) proposed the use of little movements when the iterations begins and then to increase their magnitude if the energy is minimized a repeated number of iterations. This modification avoids oscillations around local minima and also speed up the convergence rate of the algorithm.

The main disadvantage of this method is that edge crossing cannot be always avoided because the energy (or objective function to minimize) does not take into account edge crossings; in terms of an optimization problem it converges to any local minima. Besides, the algorithm is strongly dependent of the initial state, often randomly generated. In order to overcome this disadvantage and eliminate the randomness, we propose to find an optimal initial state using a previous determinist algorithm. Due to the low number of vertices of the graphs of kinematic chains of the studied mechanisms, the combinatorial algorithms are a good choice for producing a right initial condition.

2.2 Combinatorial algorithm using cycles of the graph

We present an algorithm which is a simplification of that proposed by Mauskar and Krishnamurty (1996). Given the graph of the mechanism in the form of its adjacency matrix, the steps of the algorithm are summarized as follows:

Step 1) Compute the number ν of independent cycles of the graph and obtain a basis of independent cycles (Pucheta, 2008); using this basis compute all cycles of the graph without cut-vertices.

Step 2) From the list of all cycles take a combination of ν cycles and then take a permutation of them.

Step 3) Using the order given by the permutation, the vertices of the first cycle are drawn over a circle, the successive cycles are drawn as vertices distributed on arcs, each arc starts from a vertex in common with a previous cycle and ends in another vertex in common. Radius of arcs are drawn in increasing size, e.g. duplicating the previous radius of the previous cycle. Additionally arcs can be draw in two directions since the line connecting the starting and ending node of the arcs divide the space in two half spaces, so that all combinations of directions are explored (there is one choice for drawing the base circle and there are $2^{\nu-1}$ possibilities for drawing the arcs).

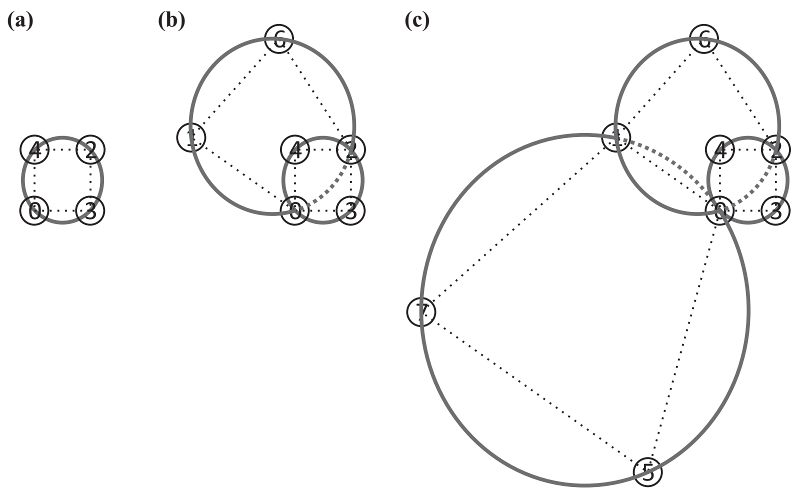

Figure 2: Algorithm based on loops.

Let us consider, for instance, a graph with three loops. Then the computation of all loops results in the list:

$$l_1 = \{0, 3, 2, 4\}$$
$$l_2 = \{0, 1, 6, 2, 3\}$$
$$l_3 = \{0, 1, 6, 2, 4\}$$
$$l_4 = \{0, 1, 7, 5\}$$
$$l_5 = \{0, 3, 2, 6, 1, 7, 5\}$$
$$l_6 = \{0, 4, 2, 6, 1, 7, 5\}$$

From this list an instance of the algorithm uses the following loops:

$$l_1 = \{0, 3, 2, 4\}$$
$$l_2 = \{\underline{0}, 1, 6, \underline{2}, 3\}$$
$$l_5 = \{\underline{0}, 3, 2, 6, \underline{1}, 7, 5\}$$

and the resulting graph layout is shown in Figure 2 with dashed lines. Using the loop l_1 a base circle is drawn (see Fig. 2b), then the algorithm searches two vertices in common with the following loop l_2, e.g. vertices 0 and 2 are selected as starting and ending point of the arc, since vertex 3 is already in a previous loop, it is ignored and the arc includes the list of vertices $\{2, 6, 1, 0\}$, see Fig. 2b. Then we process the following loop l_5, the first two vertices in common with the previous loop l_2 are vertices 0 and 1, since vertices $\{3, 2, 6\}$ were already included in the layout, the new arc will include vertices 7 and 5, Fig. 2c. By connecting vertices of the base circle and those of the successive arcs we obtain the graph layout.

2.3 Proposed algorithm

We use a combination of the two algorithms presented in the previous subsections: we firstly execute a combinatorial algorithm based on drawing loops followed by a force-directed algorithm based on spring repulsion and electrical attraction.

Since the existence of vertex-to-vertex repulsion forces does not ensure the convergence to situations without edges crossings with low energy, we have implemented a vertex-to-edge repulsion force of parabolic form (see Fig.3). The initial situation given in Fig. 4 is guided to the state shown in Fig. 5 if only the vertex-to-vertex repulsion forces are used.

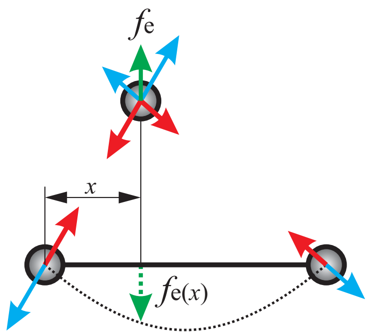

Figure 3: The verted-to-edge repulsion force is applied on the vertex in direction normal to the edge.

The inclusion of the mentioned vertex-to-edge repulsion force avoids that any initial situation without edges crossings be modified in the final layout to one with edges crossings or more edges crossings than the original one. The application of this modification on the graph shown in Fig. 4 is now guided to the state shown in Fig. 6.

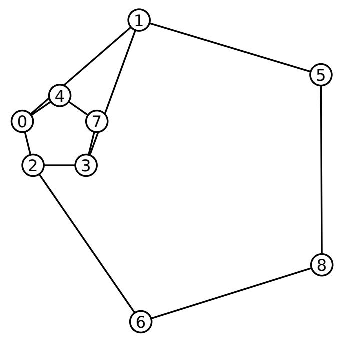

Figure 4: Results of loop-based layout algorithm executed for the graph of a 2-DOF kinematic chain.

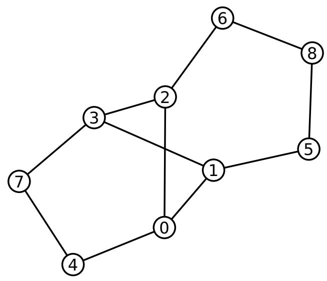

Figure 5: After loop-based layout (see Fig. 4), effect of force-directed layout with node to node repulsion and local attraction.

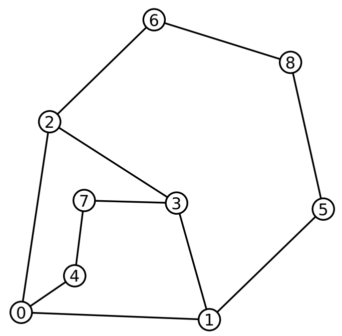

Figure 6: After loop-based layout (see Fig. 4), effect of force layout with node to node repulsion, local attraction and parabolic node to edge repulsion.

The main disadvantage of this algorithm is that it requires a selection criteria to retain the best layout among all the loop combinations used to generate the initial graph.

3 CONVERSION FROM GRAPH TO KINEMATIC CHAIN OF A MECHANISM

In this section we present a method to convert the graph of the mechanisms to its kinematic chain representation; this problem was called "Direct sketching" by Belfiore and Pennestri (1994). We also refer algorithms for corrections of new edge crossings occurrence after this conversion.

A basic procedure for direct sketching of the kinematic chain is merely geometric, starting from the graph (see Fig. 7a) it consist of locating a joint on the mid-point of each edge of the graph, see Fig. 7b. Then, joints are joined by lines if they are adjacent to a vertex of graph so that a vertex converted into a link represented by a line (if the vertex degree is 2) or by a polygon (if the vertex degree is higher than 2) formed by joints belonging to that link.

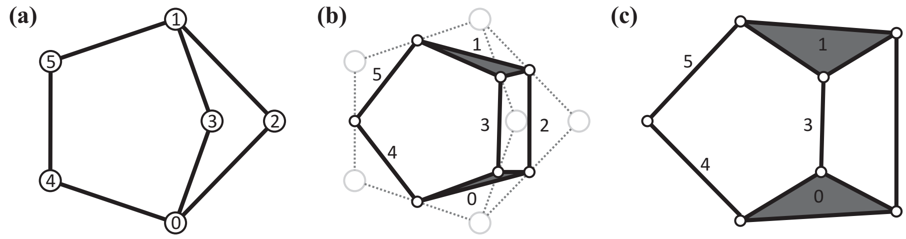

Figure 7: Direct sketching of the kinematic chain and refinement using a force-based algorithm.

After the geometric conversion, new link crossings can be originated. One way to overcome this problem was proposed by Belfiore and Pennestri (1994) were joints belonging to the convex hull of the graph are moved over lines perpendicular to the edges of the graph. Since this procedure does not avoid link crossings completely, several geometric algorithms for correction of this cases were proposed by (Ulrich, 2010). Ulrich's algorithm analyzes the penetration of each pair of conflictive links moving the joints to a right position. After this geometric corrections, we can execute the force-based algorithm in order to improve the aesthetics of the layout of the kinematic chains; some additional auxiliary edges (which are not drawn) connecting each node of the polygonal links are used to obtain more regular polygons, see 7c.

In Figs. 8, 9, and 10, we can see the direct sketching starting from the different layouts of the same graph, the last kinematic chain has the best aesthetics.

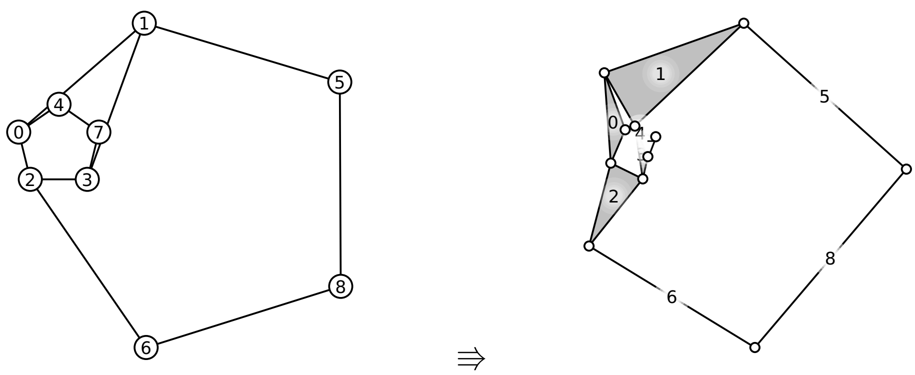

Figure 8: Conversions for graph of Fig. 4.

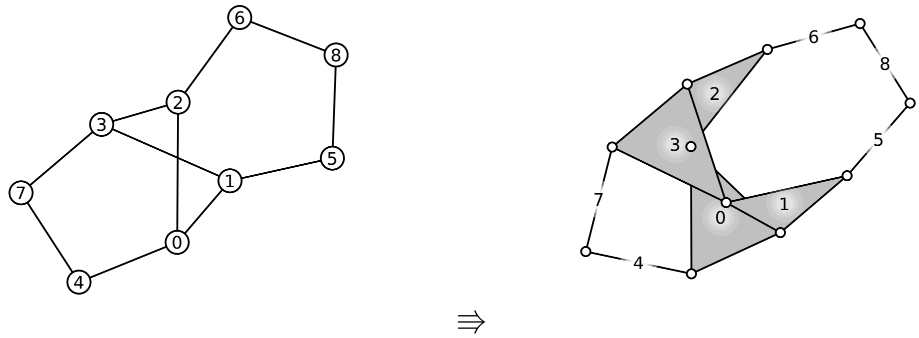

Figure 9: Conversions for graph of Fig. 5.

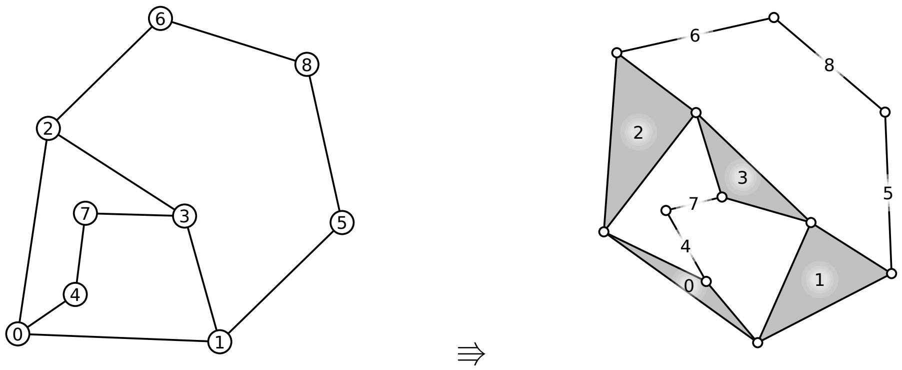

Figure 10: Conversions for graph of Fig. 6.

Figure 11 shows a graph layout of a complex 1-DOF linkage where link crossing cannot be avoided, this example was presented by Mauskar and Krishnamurty (1996, Fig.12, p.436).

In order to test these algorithms, a software prototype was written in C++ language and developed under the QT environment (Nokia, 2010) by the second author in the framework of a Fellow for research initiation (Cientibeca) of the *Universidad Nacional del Litoral*.

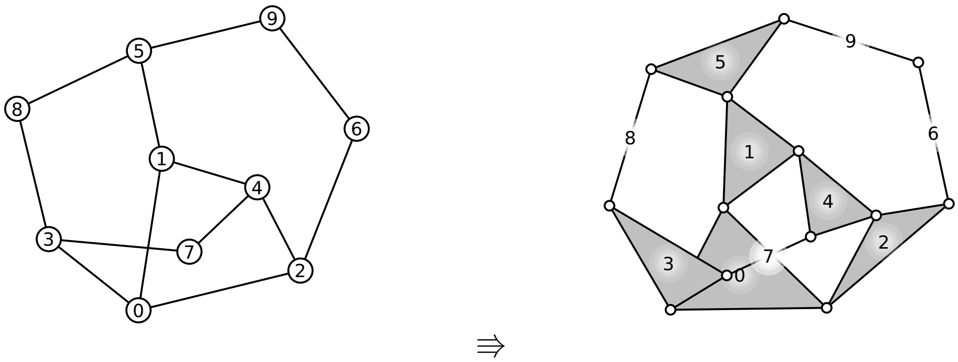

Figure 11: Graph of a kinematic chain which cannot be draw without link crossings.

4 RESULTS

The methodology for graph layout and direct sketching of kinematic chains was used over an atlas with one degree-of-freedom kinematic chains with up to three loops. The results are shown in Figs. 12 and 13. In comparison with the same layouts developed with existent software (North, 2004) our results are similar or better in most cases.

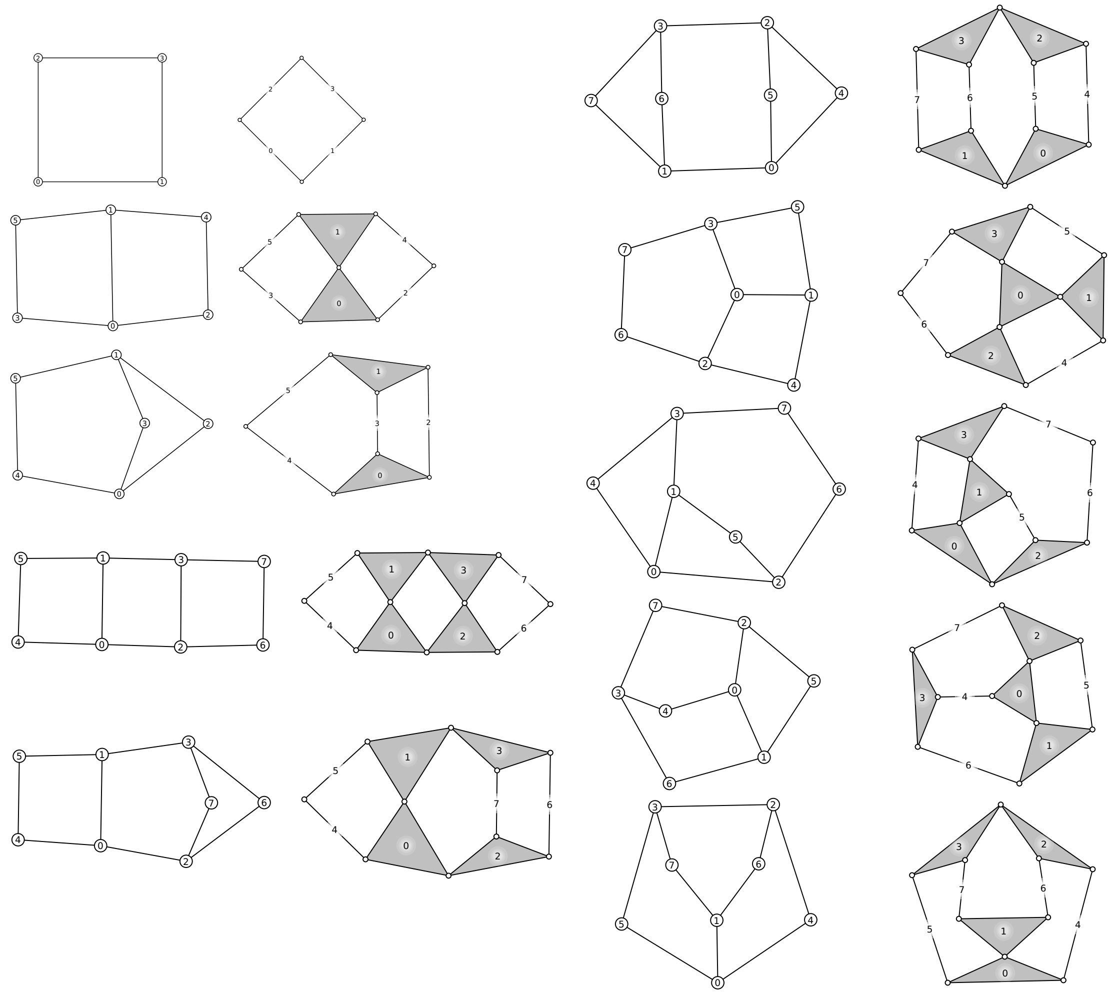

Figure 12: Atlas of 1DOF kinematic chains up to 8 links 10 joints (contd).

In Fig. 14 we also show the results obtained for an atlas of two degree-of-freedom kinematic chains.

5 CONCLUSIONS AND FUTURE WORKS

In this work we have presented a combination of loop-based algorithms and force-directed algorithms with several improvements to layout graphs and sketch kinematic chains. The layouts obtained have good quality in terms of minimization of edge crossings and maximization of aesthetic characteristics, obtaining configurations without edge crossings wherever possible. However, the loop-based layout algorithm used to find an initial layout without edge crossings greatly increases its complexity as the number of loops in the graph grows. Future developments will try to reduce the complexity of the algorithm by using the knowledge of simplest

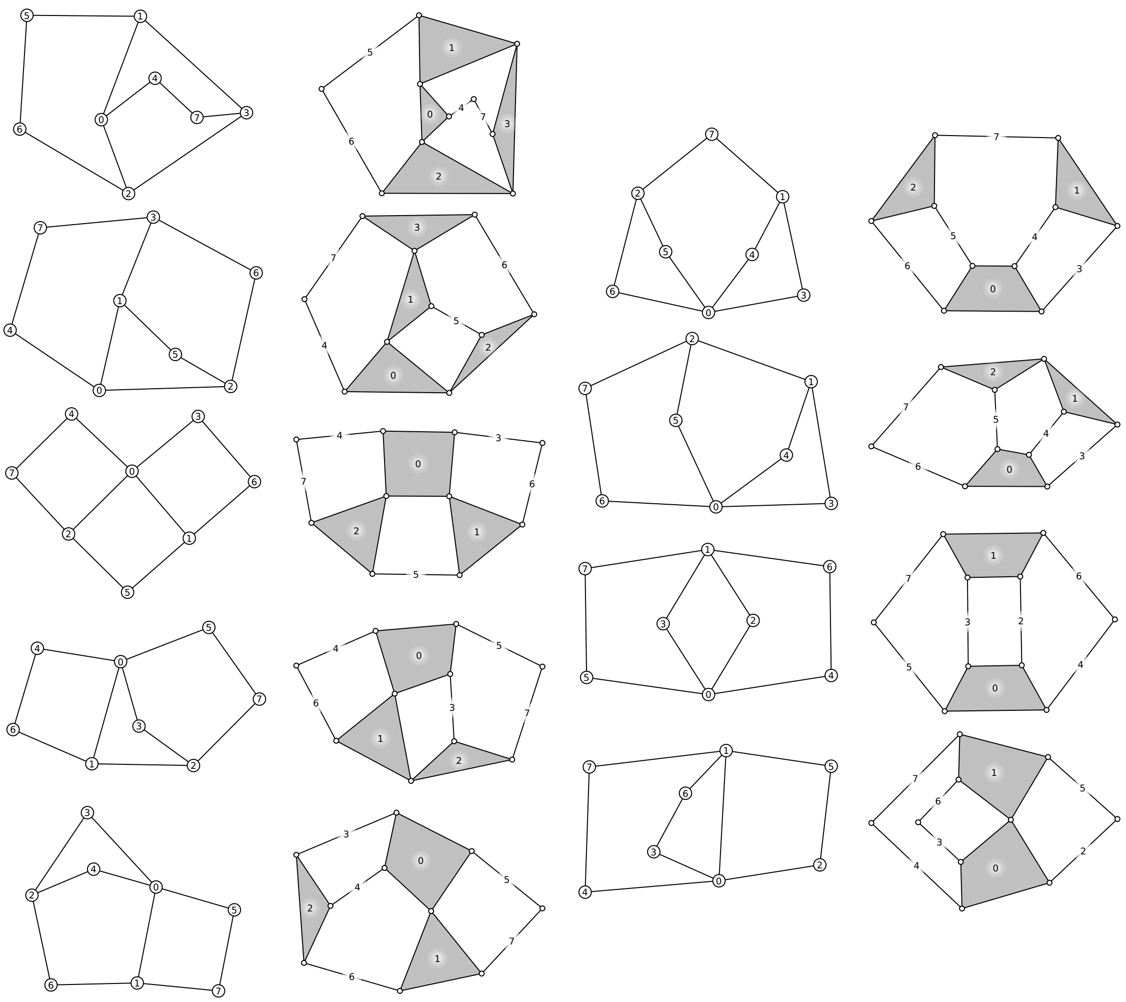

Figure 13: Atlas of 1DOF kinematic chains up to 8 links 10 joints (from Fig. 12).

structures subjacent in the graph. For instance, we think on using the contracted graph structure as proposed Belfiore and Pennestri (1994), and the proposed force-directed algorithm as a multilevel approach. We will also develop comparisons with other optimization algorithms, for instance, evolutive algorithms.

ACKNOWLEGMENTS

This work has received financial support from the following Argentinean institutions: *Consejo Nacional de Investigaciones Científicas y Técnicas* (CONICET) PIP2009 112-200801-02473 and *Universidad Nacional del Litoral (UNL)* CAI+D 2009 PI65-330 *"Computational Synthesis of Mechanisms"* project and *Fellow for research initiation (Cientibeca)* of UNL for the second author.

REFERENCES

Belfiore N.P. and Pennestri E. Automatic sketching of planar kinematic chains. *Mechanism and Machine Theory*, 29:177–193, 1994.

Fruchterman T.M.J. and Reingold E.M. Graph drawing by force-directed placement. *Software*

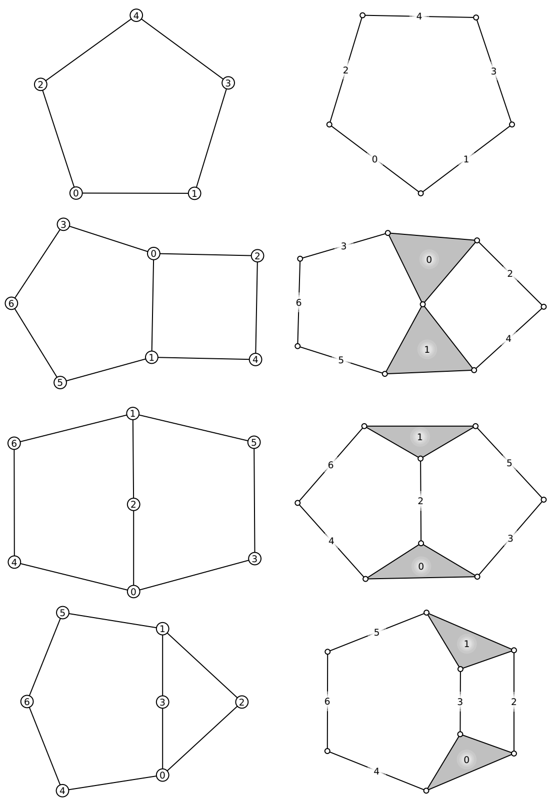

Figure 14: Atlas of 2DOFs kinematic chains up to 7 links 8 joints.

- *Practice and Experience*, 21(11):1129–1164, 1991.

Godoy R.J. Diseño de un framework de algoritmos evolutivos y aplicación al problema de representación de grafos (in spanish). In *Proceedings of XIII Congreso Argentino de Ciencias de la Computación CACIC 2007*, pages 1–9. Resistencia, Chaco, Arentina, 2007.

Hu Y. Efficient, high-quality force-directed graph drawing. *The Mathematica Journal*, 10(1):37–71, 2006.

Mauskar S. and Krishnamurty S. A loop configuration approach to automatic sketching of mechanisms. *Mechanism and Machine Theory*, 31:423–437, 1996.

Nokia. QT - A cross-platform application and UI framework. 2010. Web page:http://qt.nokia.com/.

North S.C. Drawing graphs with NEATO. 2004. AT&T Research. Web page: http://www.graphviz.org/.

Pucheta M. *Computational methods for design and synthesis of planar mechanisms*. Ph.D. thesis, Universidad Nacional del Litoral, Santa Fe, Argentina, 2008.

Pucheta M.A. and Cardona A. Automated type and modular dimensional synthesis of planar linkages. In *ASME IDETC/CIE 2010*. Montreal, Quebec, Canada, 2010a. Paper DETC2010-28540.

Pucheta M.A. and Cardona A. Design of bistable compliant mechanisms using precision-position and rigid-body replacement methods. *Mechanism and Machine Theory*, 45(2):304–326, 2010b.

Ulrich N.E. Algoritmo para la esquematización automática de grafos de mecanismos. 2010. A ser presentado en XIV Encuentro de jóvenes investigadores de la UNL – V Encuentro de jóvenes investigadores de universidades de Santa Fe.