Asociación Argentina
de Mecánica Computacional

Mecánica Computacional Vol XXVI, pp.3043-3055
Sergio A. Elaskar, Elvio A. Pilotta, Germán A. Torres (Eds.)
Córdoba, Argentina, Octubre 2007

KINEMATICS SYNTHESIS OF PLANAR MULTI-LOOP LINKAGE MECHANISMS FOR MULTIPLE TASKS PURPOSES

Martín A. Pucheta and Alberto Cardona

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

Keywords: Multi-loop linkage mechanisms, type synthesis, dimensional synthesis, multiple kinematic tasks.

Abstract. The essence of mechanism synthesis is to find the mechanism for a given motion or task. There are three customary tasks for kinematic synthesis: path following, rigid-body guidance, and function generation. However, two or more of these tasks may be required to be performed for different bodies of the same mechanism. In this paper, we present a fairly general method to solve the kinematics synthesis problem for multiple tasks. In order to apply analytical synthesis equations, the task is discretized by a number of prescribed displacements and orientations called "passing points". A Finite Element and Graph Theory representation of mechanisms is used to represent the prescribed motion constraints on the parts of the problem. Also, Graph and combinatorial algorithms are used to solve the type synthesis problem listing a discrete number of feasible non-isomorphic topologies. Then, for each feasible topology, a multi-objective optimization based on a Genetic Algorithm –zero order search– is run to find the initial dimensions and pivot positions of the unknown parts of the linkage.

Computer implementation of this method were programmed in *C++* language under the *Oofelie* environment and was presented in previous AMCA congresses. The aim of this paper is to incorporate new data and modify the existing algorithms to take into account multiple tasks. A double task which combines a path following and a rigid-body guidance problem will be presented throughout the paper.

1 INTRODUCTION

Mechanism synthesis is an inverse problem where the aim is to find the mechanism for a given motion. Specifically, we restrict ourselves to study the synthesis of planar single and multi-loop linkages mechanisms. Conceptual design of mechanisms has two main stages: (i) *Type Synthesis*, where the number, type and connectivity of links and joints are determined for the required degree-of-freedoms and structural constraints, and (ii) *Dimensional synthesis*, where the link lengths and pivot positions at the starting position are computed. In the pursuit of an optimal design, the user must evaluate both, type and dimensional synthesis. For this purpose, we have developed computational tools programming methods for type and dimensional synthesis of mechanisms.

Planar multi-loop linkage mechanisms are used to develop complex and non-linear motions. In classic literature (Sandor and Erdman, 1984; Erdman and Sandor, 1997), three well known customary tasks for kinematics synthesis were deeply studied:

- Path Following (PF), where the purpose of the synthesis is to determine the dimensions of a mechanism so that one point of the mechanism moves through a series of preset positions.
- Rigid-Body Guidance (RBG), the positions and also the orientations of a point in the mechanisms are prescribed.
- Function Generation (FG), the orientations of two members of the mechanism must follow a prescribed function or law.

In the practice of solving industrial problems, these tasks often appear in a combined way: two or more of these tasks may be required to be performed for different bodies of the same mechanism. The motivation and main objective of the present work is to complete those already developed methods to solve these multiple tasks, trying also to preserve the generality needed in Computer-Aided Synthesis Methods.

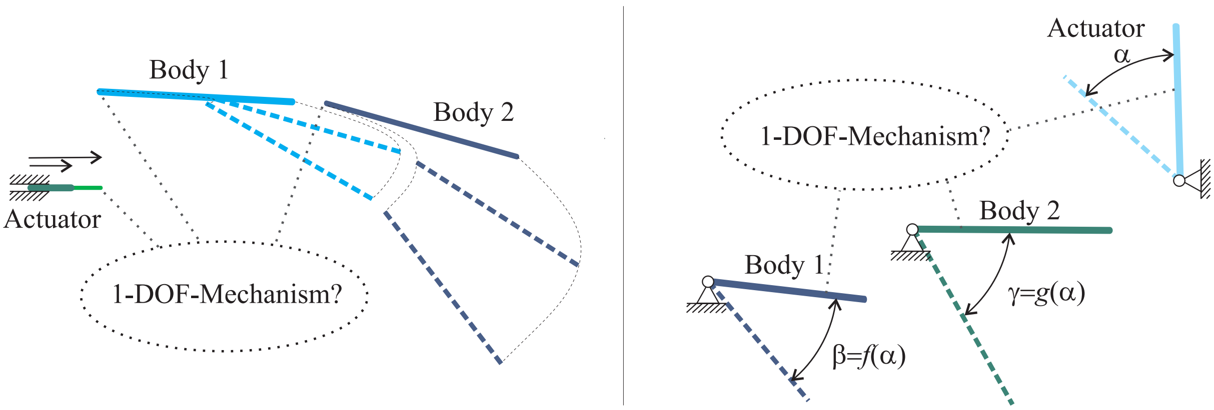

Figure 1: Multiple Rigid-Body Guidance (left) and Multiple Function Generation (right).

Graph Theory proves to be a useful mathematical tool for modelling and computer implementation of discrete problems, thus it has been extensively used for solving the type synthesis of mechanisms in the last four decades (Mruthyunjaya, 2003). In a recent work of the authors, a method to find type synthesis solutions from the Finite Element Method (FEM) description of a prescribed task was proposed. The problem of finding and codifying all-non isomorphic solutions for a given kinematic problem was solved using Graph Theory and combinatorial algorithms (Pucheta and Cardona, 2007a). An important characteristic of the method is that both the *FEM* description of the parts entered by the user in the initial problem and its automatically made *Graph* conversion are retained and completed with new members –links and joints– present in the final solution. Then, the mechanism can be dimensionally sized by many techniques.

The dimensional synthesis of mechanisms is a highly non-linear problem. In most cases, for any given mechanism topology, it is difficult to find a good initial guess for starting a gradient-based optimization. The geometrical methods based on Precise Positions, also called Precision-Point Methods, are of great help to find initial dimensions. There exist well known procedures to synthesize single open-chains (SOCs) like dyads and triads passing through three, four and more positions (Erdman and Sandor, 1997; Sandor and Erdman, 1984). We have programmed dyads and triads passing through three and four positions, whose solutions are exact and fast to compute (consequently, do not need the use of iterative methods). Depending on the initial data of these SOCs, we may need to propose some free parameters. To apply this analytical method for solving a given type synthesized solution, i.e. a mechanism topology, it is necessary to decompose the closed chain into several SOCs. This decomposition is not unique and different decompositions might lead to a different number of free parameters. In a previous work we compute many decompositions using the minimal set of independent loops of the topology, and then we retain those decompositions which present a set of SOCs with a minimal number of free parameters (Pucheta and Cardona, 2006). We also take the convention of preferring those sets which solve the largest number of imposed motion constrains in a specified order, from the first to the last SOC.

Then, after retaining a decomposition we run a multi-objective optimization where we propose values for the free parameters to solve the set of SOCs which minimizes the size of the mechanism, and also we check some constraints useful for linkage design: minimal length of links, allowed space constraint, and a singularity constraint to avoid locking situations.

This sequence of procedures for type and initial dimensioning was successfully used to solve linkages with four-, six- and eight-bars rigid mechanisms. Although the present work deals with the design of rigid mechanisms, the solution of multiple tasks finds special application for the guidance of compliant members. This idea and the study of the Rigid-Body Replacement of binary links (Howell, 2001) are promising areas to obtain flexible mechanisms (Pucheta and Cardona, 2007a).

In Section 2 we review the available method for type synthesis of mechanisms and incorporate the proper changes in order to take into account multiple tasks. In Section 3 the decomposition method for multiple tasks is explained. In Section 5 we show the results for an example used throughout this paper.

2 TYPE SYNTHESIS REVIEW

Given structural requirements, the aim of Type Synthesis is to find the number of parts, number of links and joints, types of links and joints, and their connectivity. The work can be simplified if the potential mechanisms solutions are previously enumerated and stored in a data base. Then, it only remains to select those mechanisms for which some parts match with those given in the task. Since the order of the enumerated mechanisms could be of hundreds, even thousands, the visual inspection may easily lead us to neglect some feasible alternatives, for this reason we resorted to the aid of the computer.

The enumeration of mechanisms is actually a challenging area (Mruthyunjaya, 2003). The enumeration process is usually divided into: (i) the enumeration of basic kinematic chains (BKC) for a given number of degree-of-freedoms (Tsai, 2001), and (ii) the assignment of types of links and joints for each BKC, a process which is also called "Specialization of Mechanisms" (Yan and Hwang, 1991). We have followed this procedure to form different atlases of mechanisms, represented by graphs codified in an unequivocally way. Each kinematic chain is represented by the adjacency matrix of its graph $G(E, V)$, where each link is represented by a vertex $v_i \in V$ and each joint $e_{ij} \in E$ is the connection between vertices v_i and v_j. The specialization of the links and joint types is formulated as an assignment problem and a colored graph representation is used. The adjacency matrix of the colored graph has integer entries representing link types on the diagonal entries and joint types on the outer diagonal ones. We call it *Type Adjacency matrix* $\boldsymbol{T}$, and it is defined as follows:

$$T_{ii}(v_i) = \begin{cases} 0 & \text{if } v_i \text{ is the ground,} \\ 1 & \text{if } v_i \text{ is a rigid link,} \\ 2 & \text{if } v_i \text{ is a flexible link,} \end{cases} \quad T_{ij}(e_{ij}) = \begin{cases} 0 & \text{no connection,} \\ 1 & \text{if } e_{ij} \text{ is a revolute joint,} \\ 2 & \text{if } e_{ij} \text{ is a prismatic joint,} \\ 3 & \text{if } e_{ij} \text{ is a flexible joint,} \\ 4 & \text{if } e_{ij} \text{ is a clamped joint.} \end{cases}$$

Using this matricial representation and an isomorphism identifier based on the Degree Code we codified atlases with one and two degree-of-freedoms, rigid and flexible links, and revolute, prismatic, flexible, and clamped joints (Pucheta and Cardona, 2007b).

We also propose to represent the kinematic problem in hand by a graph that we called "Initial Graph". Since the kinematic problem and the atlases are represented by graphs, the type synthesis process consists in detecting and codifying all non-isomorphic subgraph occurrences of the Initial Graph inside a selected atlas.

The number of inputs desired for actuating the mechanism solution is managed by only selecting the atlas already available with the proper number of degree-of-freedoms. The number of outputs of the mechanism is described by the "Initial Graph" construction. This graph is built from the interpretation of the Finite Element description of the kinematics problem. Using the *Samcef Field* graphic interface (SAMTECH Group, 2007), the geometric description of the known mechanism's parts is entered in terms of nodes which are then used for defining bodies (Wire, Face, and Lumped Mass elements) and the allowed space. Next, the user defines elements for assembling bodies (Hinge, GroundHinge, GroundPrismatic, and Fixed connections), and finally defines the motion constraints prescribed at the synthesis problem, either by setting the motorization of joints or by defining constraints (Clamp, Prescribed Displacement, Prescribed Rotation) over nodes and bodies. This initial situation is analyzed and automatically converted into a graph following these rules:

Vertices: Free bodies with imposed movements will be isolated vertices of the initial graph. The remaining bodies, connected through joints, will be connected vertices of the graph. Conventionally, the ground link will be the vertex zero. Depending on the number of grounded bodies, this vertex may be binary, ternary, etc. The number of isolated fixations (represented by fixed nodes) is used to prescribe the degree of vertex zero (ground). For each isolated node with prescribed movements, we assign an isolated vertex in the graph, although this node is not attached to any element.

Edges: Joints will be edges of the initial graph connecting two of the previously defined vertices; all edges are assumed to be binary (isolated joints are not allowed).

The application of these rules for multiple tasks like (PF) and (RBG) produces isolated –disconnected– vertices on the Initial Graph. For instance, in Figure 2-a we show the requirements for a mixed problem where the trajectories for two isolated nodes are imposed (the node in the right also has imposed rotations), thus we obtained two vertices in the initial graph in correspondence with the floating links developing the tasks (Figure 2-b).

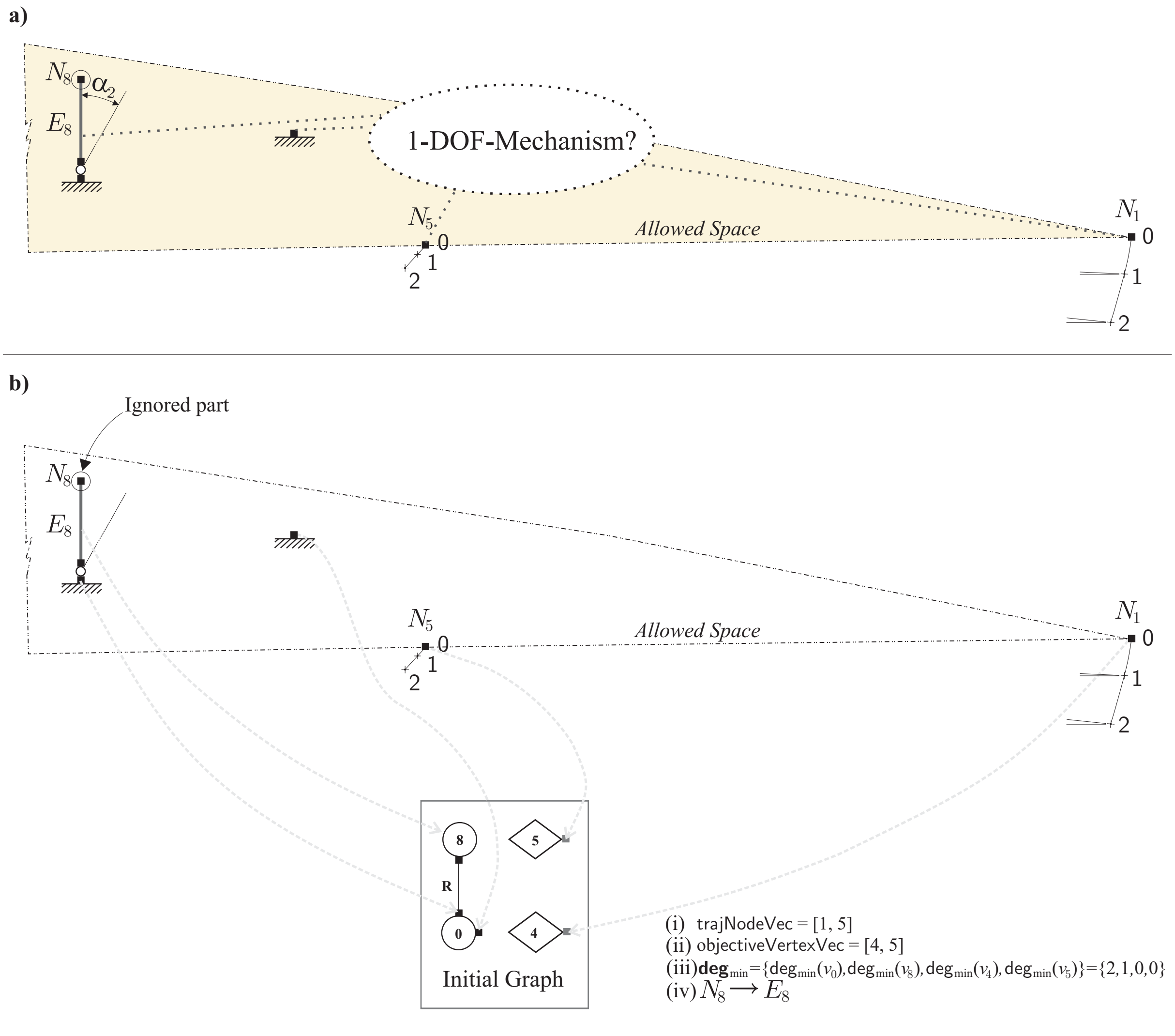

Figure 2: Translation from FEM (a) to graph (b) representation for a double task.

We allow defining adjacency matrices with both null rows and null columns corresponding with a disconnected body. For example, the *Initial Graph* shown in Figure 2-b has a set of vertices $V = [v_0, v_1, v_2, v_3]$ labelled as $\mathcal{V}(V) = [0, 5, 8, 4]$. Its type adjacency matrix is:

$$\boldsymbol{T}_{\text{ini}} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

For tasks of PF and RBG it is undesirable to have floating links connected to the ground, thus we add a *distance constraint* limiting the distance between the ground v_0 to the floating links to a minimum value of 2.

Some results of the subgraph search for this example will be shown in Section 5, but we must remark that the subgraph search admits initial graphs with disconnected components. See for example, the first subgraph occurrence in Figure 3 of the following section. Since, the fulfillment of the *distance constraint* rejects the possibility of getting four-bars solutions, this first solution is a six-bars topology. The set of vertices $V_0 = [v_0, v_1, v_2, v_3, v_4, v_5]$ are labelled as $\mathcal{V}(V_0) = [0, 8, 11, 12, 5, 4]$. Its type adjacency matrix is:

$$\boldsymbol{T}_0 = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}.$$

3 TOPOLOGY DECOMPOSITION

The decomposition of the topology into SOCs is achieved using some background of Graph Theory. The number of independent closed loops, ν, is a property of the graph. Since all mechanisms have closed topology it is always implicit in their number of vertices v and edges e, then $\nu = v - e + 1$. The set of independent loops of minimal length allows to find the significant dimensions of links (Erdman and Sandor, 1997).

The decomposition method consists in the following steps (Pucheta and Cardona, 2006):

S1) **Topology decomposition:** The kinematic chain (closed-loops chain mechanism) is decomposed into a set of separated closed-loops[1] of minimal length.

S2) **SOCs decomposition:** For each set of closed-loops, each closed-loop is selected in a given order to be decomposed into SOCs, i.e. dyads, triads, quadriads, etc., using the node displacement constraints.

S3) **SOCs evaluation:** After analyzing data (geometry and synthesis data definitions), the SOCs solvability is evaluated in the resultant order.

S4) **Retained ordered SOCs:** The best valuated combination/s of open-chains is/are stored for dimensional synthesis.

This method is also suitable for multiple tasks, the main improvement is that we form a vector of trajectory nodes called trajNodeVec, and a vector objectiveVertexVec with their corresponding vertices in the graph (Figure 3-a). These data are used for the determination of the significant dimensions in the linkage. The computation of the *basis of minimum independent loops* is made using the graph, see Figure 3-b.

As it is shown in Figure 3-c, the significant dimensions are found by means of the extension of the loop which visits a given vertex in objectiveVertexVec[i] passing through the corresponding node in the trajNodeVec[i] with equal index of position i. These loops are marked in order to do a decomposition in both orientations.

[1] Known as *Cycles* in Graph Theory (Harary, 1969) or *Circuits* (Tsai, 2001).

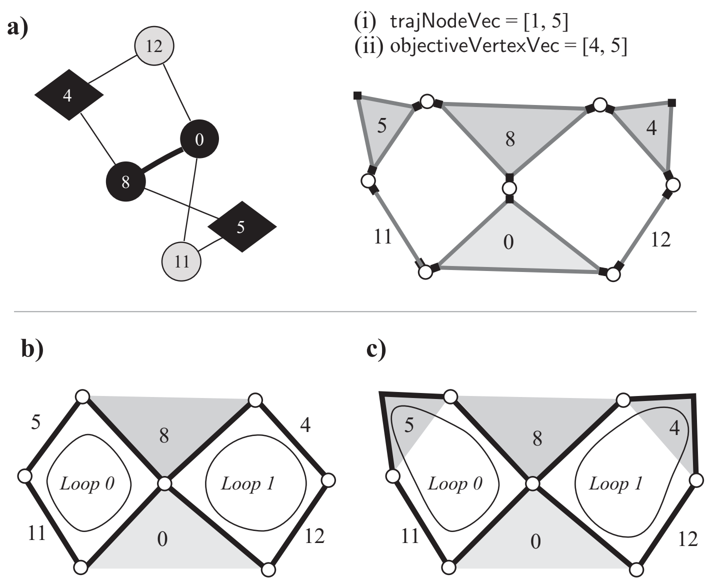

Figure 3: Minimal independent loops of the graphs and the additional vectors with information referred to nodes (i) and their associated vertices (ii).

4 INITIAL SIZING

The type synthesis stage finished with the decomposition into single-open chains. Then, a second stage of initial sizing for synthesis is launched for each feasible alternative. The design space is defined by the set of free parameters, if any. The user can change the values of their bounds, and a genetic algorithm is used to sweep the design space. The fitness function consists in the minimization of the size of the mechanisms together with three weighted constraints: minimal length of link dimensions, non-inversion of transmission angle, and allowed space violation (Pucheta and Cardona, 2005). Eventually, instead of considering non-inversion of transmission angle as a constraint, a full kinematics analysis is made for each individual to compute the fitness function.

A posteriori of the synthesis task, the user can verify the validity of the proposed solution by means of analysis, and he can even optimize the solution staying within the same topology.

5 RESULTS

The proposed test problem consists in finding a mechanism able to produce the deflection of two curved beams. The boundary conditions are shown in Figure 4. The topology can be considered as a compliant mechanism itself. If this mechanism is actuated by the prismatic joint, its full kinematics description at any point of the beams can be obtained.

In a previous work, (Pucheta and Cardona, 2007a), we presented solutions for this problem where we guided the tip of the beams (Rigid-Body guidance) considering the prismatic joint (shown on the left of Fig. 4) as passive. We propose to solve this compliant synthesis problem using synthesis of rigid mechanisms with multiple tasks. In the following subsections the synthesis stages are illustrated.

5.1 A mixed path following and rigid-body guidance example

In this paper, we desire to guide the tip but also another point, for example, we can choose the intermediate point, P_H, of the lower beam (Figure 5). The point is guided by a set of positions,

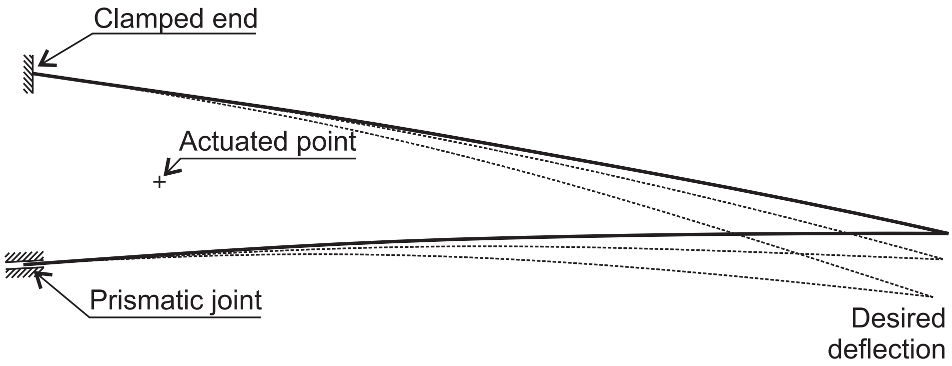

Figure 4: Boundary conditions for the deflection of two beams.

without prescription of orientations. This means that we must add a hinged connection between the mechanism and the lower beam. Therefore, we reformulate the problem as a multiple task: a path following task is required for node P_{H} and a rigid-body guidance for the tip P_{tip}.

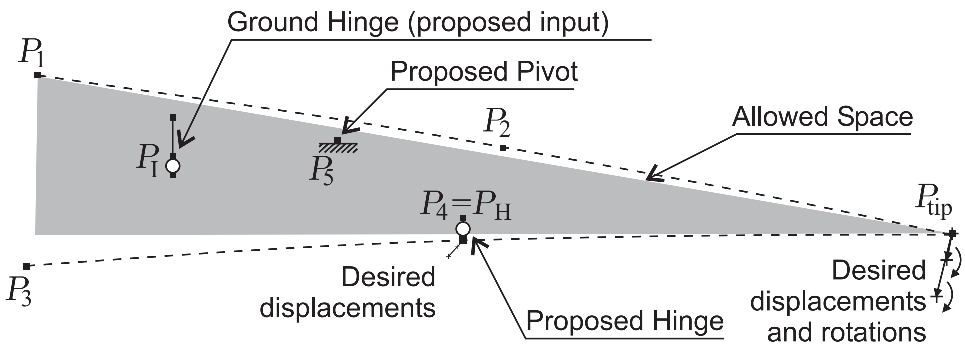

Figure 5: Deflection of two beams reformulated as a multiple tasks synthesis problem.

This problem also corresponds with Figure 2-a and the explanations given in the previous sections. Both movements must be coordinated with the rotation of a motorized crank. The mechanism solution must fit inside an allowed space.

Data for this problem are:

- Tip coordinates: $P_{\mathrm{tip}} = (1.0, 0.0)$
- Tip displacements:
  - ◇ $\delta_1^{\mathrm{tip}} = (-0.002089, -0.010085)$, and $\delta_2^{\mathrm{tip}} = (-0.005632, -0.024297)$.
- Tip rotations:
  - ◇ $\alpha_1^{\mathrm{tip}} = -2.4957°$, and $\alpha_2^{\mathrm{tip}} = -5.978°$.
- Guided hinge coordinates $P_{\mathrm{H}} = (0.79849, -0.00229)$
- Guided hinge displacements:
  - ◇ $\delta_1^{\mathrm{H}} = (-0.002325, -0.0025888)$, and $\delta_2^{\mathrm{H}} = (-0.005714, -0.0062605)$.
- Beams are defined by cubic-splines interpolation passing through the following points:
  - ◇ Upper beam: $P_1 = (0.65085, 0.06106)$, $P_2 = (0.85148, 0.03018)$, P_{tip}.

◇ Lower beam:$P_3 = (0.64915, -0.01086)$, $P_4 = (0.79849, -0.00229)$, P_{tip}.

- Ground Hinge actuator (input): $P_I = (0.7, 0.02)$.
  - ◇ Maximum displacement: $\alpha_2^I = -3°$.
- Boundary conditions:
  - ◇ Clamped constraint on proposed pivot: $P_5 = (0.76, 0.03)$.
  - ◇ Clamped constraint on node P_1 of the upper beam.
  - ◇ Grounded prismatic joint constraint on node P_3, axis direction $\hat{n} = (0.99835, 0.05729, 0)$.

5.2 Type synthesis outputs

The graphs for the first ten solutions are displayed in Figure 6. The algorithm for SOCs decomposition gives us the **Alternatives 0**, **3**, **4**, **7**, **8** and **9** as compatible to be solved analytically. Their sketches are shown in Figure 7.

5.3 Initial sizing using optimization

A multi-loop solution corresponding to **Alternative 7** is shown in Figure 8 where the beams were attached to the tip. Among the ten analyzed alternatives, this was the first one with a full satisfaction of constraints.

6 CONCLUSIONS

A method for solving the kinematic synthesis of combined tasks has been presented. A FEM representation of mechanisms and some Graph Theory theorems and algorithms were used in an integrated way to:

- Solve the type synthesis problem.
- Decompose a topology into single-open chains (SOCs).
- Compute the initial dimensions of the mechanism using analytical equations combined with a zero order search to sweep the free parameters.

A double task which combines a path following and a rigid-body guidance problem was presented throughout this paper. The presented example was only a first preliminar test for multiple task purposes.

In this paper, the tasks were defined on floating links or bodies attached to ground. This enables the user to solve many industrial problems. To improve this tool for conceptual design of mechanisms, further works will be focused on considering parts with known dimensions (sub-mechanisms).

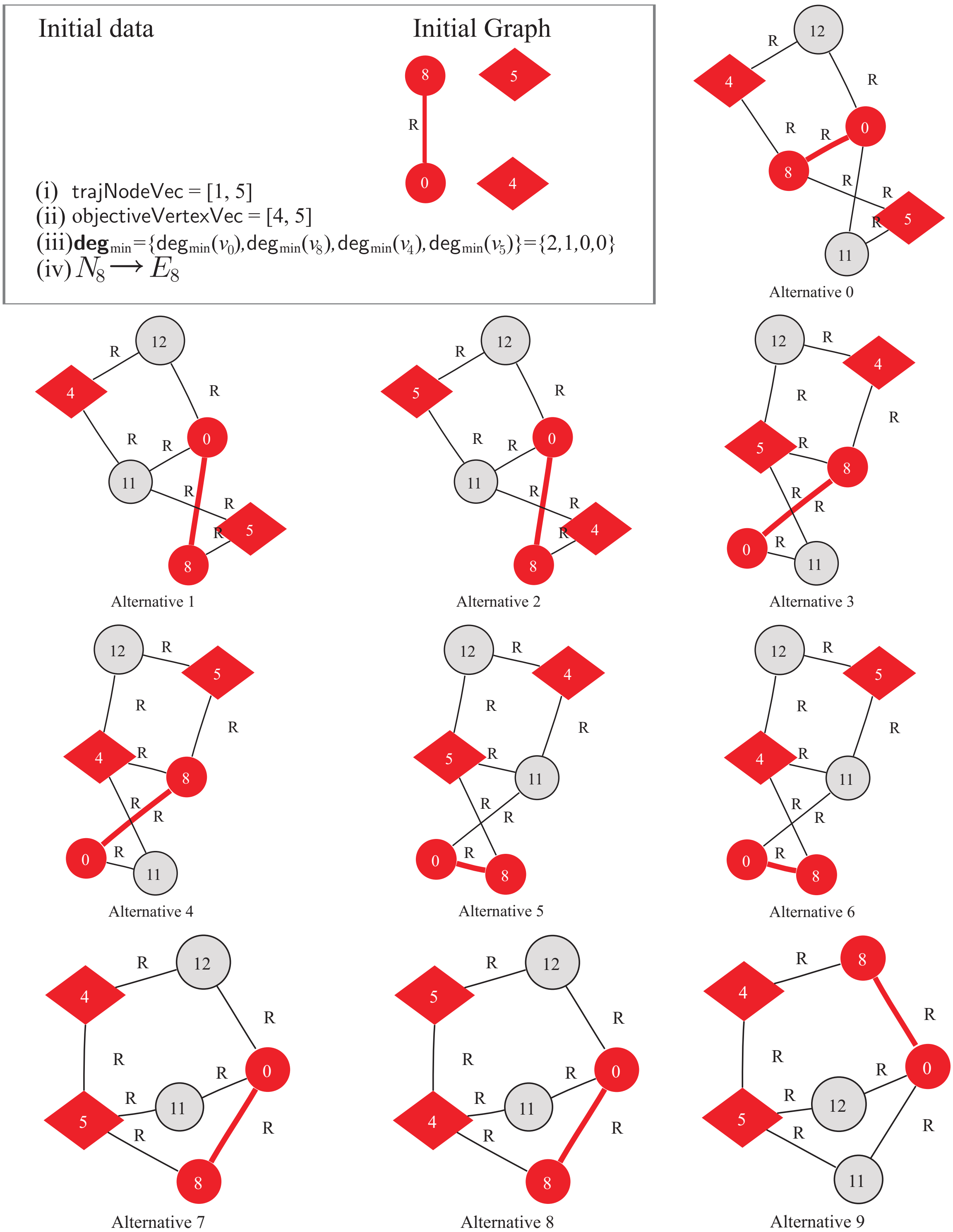

Figure 6: Data for the subgraph search and the first ten solutions of the type synthesis run.

ACKNOWLEGMENTS

This work has received financial support from *Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET)*, *Agencia Nacional de Promoción Científica y Tecnológica (ANPCyT)*, *Universidad Nacional del Litoral (UNL)* from Argentina, and from the *European Community* through grant *SYNCOMECS (SYNthesis of COmpliant MEChanical Systems)* project UE FP6-2003-AERO-1-516183 http://www.syncomecs.org.

REFERENCES

Erdman A.G. and Sandor G.N. *Mechanism Design: Analysis and Synthesis*, volume 1. Prentice-Hall, New Jersey, 3rd edition, 1997.

Harary F. *Graph Theory*. Addison-Wesley Series in mathematics, 1969.

Howell L.L. *Compliant Mechanisms*. John Wiley & Sons, New York, 2001.

Mruthyunjaya T.S. Kinematic structure of mechanisms revisited. *Mechanism and Mechanism Theory*, 38(4):279–320, 2003.

Pucheta M. and Cardona A. Kinematics synthesis of compliant mechanisms using rigid-body replacement. In C.L. Bottasso, P. Masarati, and L. Trainelli, editors, *Multibody Dynamics 2007*, ECCOMAS Thematic Conference. Politecnico di Milano, Milano, Italy, 2007a.

Pucheta M.A. and Cardona A. Type synthesis and initial sizing of planar linkages using graph theory and classic genetic algorithms starting from parts prescribed by user. In *Multibody Dynamics 2005*, ECCOMAS Thematic Conference. Madrid, Spain, 2005.

Pucheta M.A. and Cardona A. A decomposition method for modular dimensional synthesis of planar multi-loop linkage mechanisms. In *Mecánica Computacional*, volume XXVII of *XV Congreso sobre Métodos Numéricos y sus Aplicaciones, ENIEF 2006*, pages 351–373. Santa Fe, Argentina, 2006.

Pucheta M.A. and Cardona A. An automated method for type synthesis of planar linkages based on a constrained subgraph isomorphism detection. *Multibody System Dynamics*, 2007b. (Accepted for publication).

SAMTECH Group. *SAMCEF Field v6.2, SAMCEF BOSS/Quatro v6.0*. 2007.

Sandor G.N. and Erdman A.G. *Advanced Mechanism Design: Analysis and Synthesis*, volume 2. Prentice-Hall, New Jersey, 1984.

Tsai L.W. *Mechanism Design: Enumeration of Kinematic Structures According to Function*. CRC Press, Boca Raton, 2001.

Yan H.S. and Hwang Y.W. The specialization of mechanisms. *Mechanism and Machine Theory*, 26:541–551, 1991.

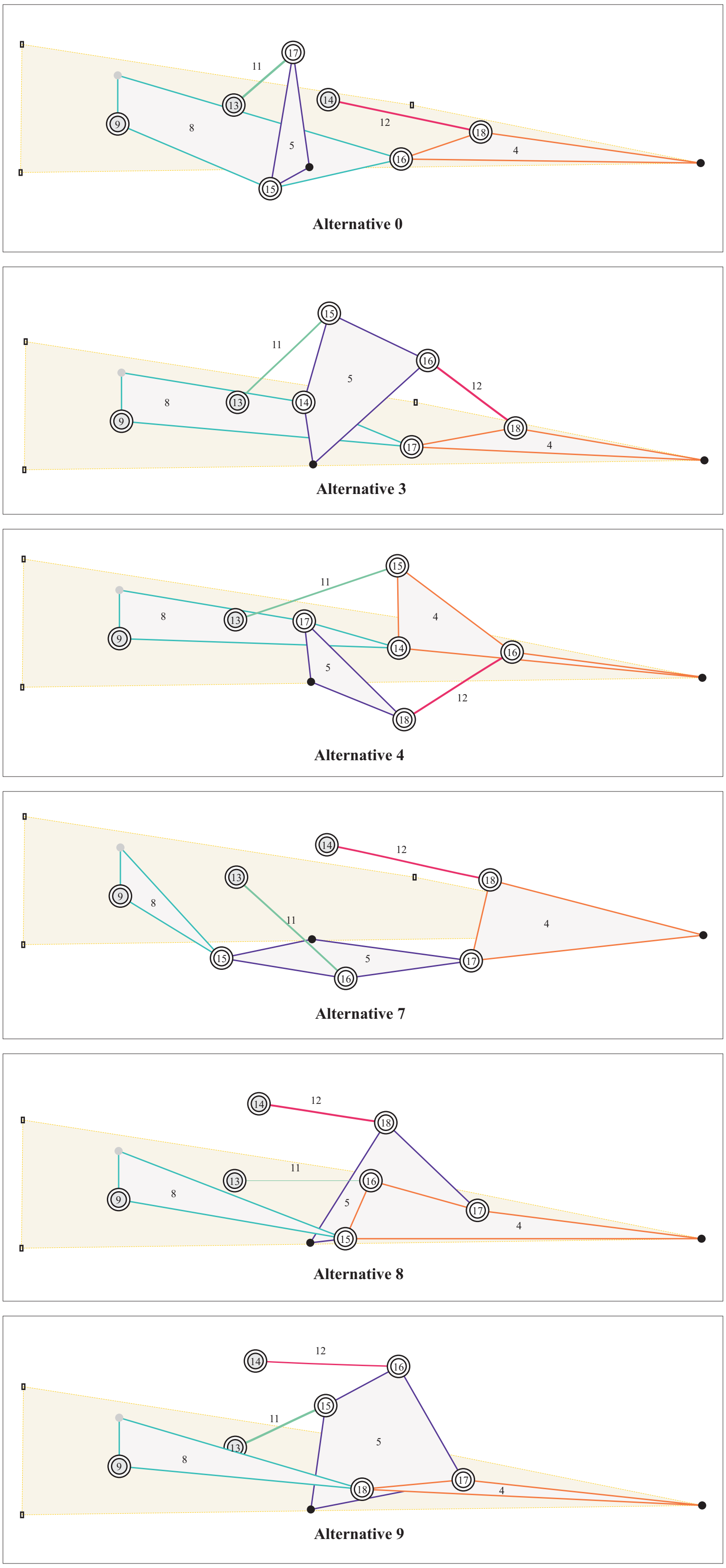

Figure 7: Sketches of the feasible decompositions.

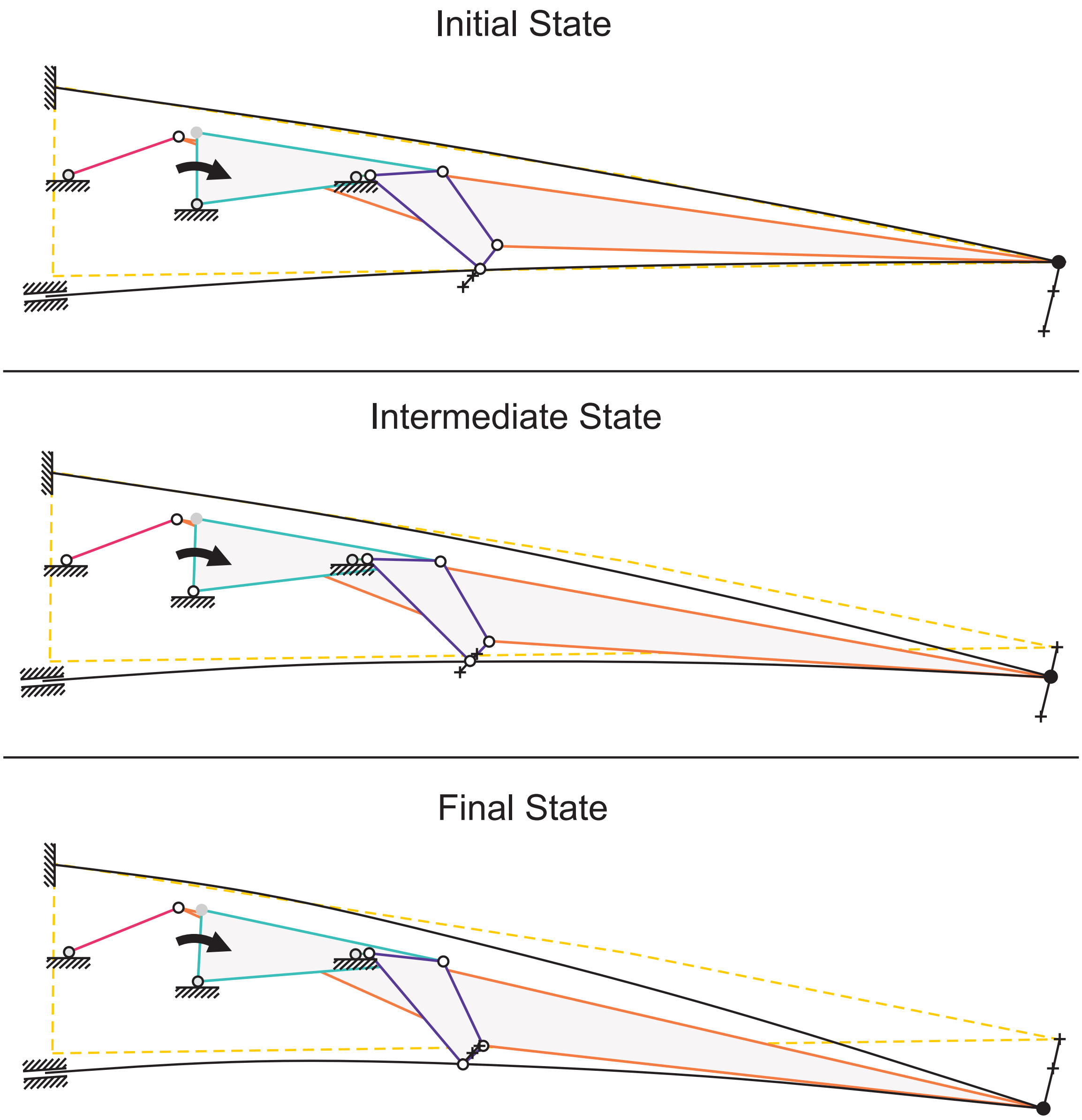

Figure 8: **Alternative 7** multi-loop linkage passing exactly through three positions prescribed for a combined task.