



HAL
open science

Defining tools to address over-constrained geometric problems

Moinet Mireille, Guillaume Mandil, Philippe Serré

► **To cite this version:**

Moinet Mireille, Guillaume Mandil, Philippe Serré. Defining tools to address over-constrained geometric problems. *Computer-Aided Design*, 2014, 48, pp.42-52. 10.1016/j.cad.2013.11.002 . hal-00946040

HAL Id: hal-00946040

<https://hal.science/hal-00946040v1>

Submitted on 10 Apr 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Defining tools to address over-constrained geometric problems in Computer Aided Design

Mireille MOINET^{a,*}, Guillaume MANDIL^{b,**}, Philippe SERRE^a

^a*Laboratoire d'Ingénierie des Systèmes Mécaniques et des MATériaux EA-2336*

SUPMECA

3 rue Fernand Hainaut

93407, Saint-Ouen Cedex

FRANCE

^b*Laboratoire des sciences pour la conception, l'optimisation et la production*

G-SCOP

46, avenue Félix Viallet

38031 Grenoble Cedex 1

FRANCE

Abstract

This paper proposes a new tool for decision support to address geometric over-constrained problems in Computer Aided Design (CAD). It concerns the declarative modelling of geometrical problems. The core of the coordinate free solver used to solve the Geometric Constraint Satisfaction Problem (GCSP) was developed previously by the authors. This research proposes a methodology based on Michelucci's witness method to determine whether the structure of the problem is over-constrained. In this case, the authors propose a tool for assisting the designer in solving the over-constrained problem by ensuring the consistency of the specifications. An application of the methodology and tool is presented in an academic example.

Keywords:

Geometric modelling, Non-Cartesian, Over-constrained, Geometric constraints, GCSP, Specification

*tel: +33 1 49 45 25 47 fax: +33 1 49 45 29 29

**tel: +33 4 76 57 47 76 fax: +33 4 76 57 46 95

Email addresses: mireille.moinet@supmeca.fr (Mireille MOINET),
guillaume.mandil@g-scop.eu (Guillaume MANDIL), philippe.serre@supmeca.fr
(Philippe SERRE)

1. Introduction

In Computer Aided Design (CAD), the model is the computer representation of the object being designed. This geometric model is often named the Digital Mock-Up (DMU) and is now the core of CAD systems. In this study, the geometric model is the reference model.

There are essentially two strategies for building a digital mock-up in CAD systems: the procedural approach and the declarative approach [1]. This paper focuses on declarative approaches because they are often used in the 2D sketcher and 3D assembly workbenches of CAD systems [2] and because the authors have already worked on them in [3, 4, 5, 6]. The declarative approach assumes that the designer first specifies a list of generic geometric objects and, second, a list of constraints between the objects defined previously. Then, a software application is used to solve all these constraints and build the virtual object. In order to obtain a valid object, it is necessary to ensure that all the specifications (generic objects and constraints) given by the user are consistent.

Moreover, the DMU is used in many simulations that cause the geometry to evolve due to changes made to the specifications required by the design team. It is therefore very important to maintain consistency in the statement of the problem. In general, problems that present inconsistencies are of two types: under-constrained problems and over-constrained problems. In this paper, we focus on over-constrained problems.

At present, when a designer comes across an over-constrained problem, no plans are available (at best a message is displayed on the screen). They must unravel it alone. Maintaining the consistency of the digital mock-up is even harder when several designers are involved in the design process. In this paper the authors propose to generate relations between the parameters of specifications in view to guiding users to ensure the consistency of the sets of parameters used. Therefore this paper will describe how to provide users with the elements necessary for maintaining the consistency of the geometry created. How is it possible to help designers to clearly specify their geometry? In the framework of collaboration, if a geometric problem is over-constrained, who arbitrates between the values of different specifications? This paper presents a solution for managing over-constrained geometric problems by giving users a tool for generating consistent sets of specifications.

The authors have already developed a conceptual model based on vectorizing the geometry for generic geometric objects. The associated solving strategy uses a coordinate-free representation. The major advantage of this approach is that it is unnecessary to take the cartesian reference frame into account for solving purposes. In 3D space, a geometric object is characterized by a Gram matrix that is positive semidefinite and has rank 3. Some elements of this matrix must have specific values imposed by the user, the others are unknowns. The geometric problem is solved by obtaining a Gram matrix \mathbf{H} that meets the above conditions. This entails a matrix completion problem [7, 8]. Indeed, in order to complete a partial matrix it is necessary to make specific choices of values for the unspecified entries.

The solution proposed by us is to find a transformation \mathbf{T} that changes an initial object into a final object. The \mathbf{G} Gram matrix characterizes the initial object. \mathbf{H} Gram matrix characterizes the final object and is defined by the fundamental relationship (1), as explained in greater detail in [9]:

$$\mathbf{H} = \mathbf{TGT}^t \tag{1}$$

More specifically, in our study, $\mathbf{T} = (\mathbf{I} + \mathbf{CX})$, where \mathbf{C} is the topological connection matrix, \mathbf{X} the vertex perturbation matrix and \mathbf{I} the Identity.

The authors make use of their previous work to address the problem of consistency. The proposed methodology and tool are applied to a case study shown in Figure 1. It is a 3D bar structure called "double banana". The lengths of the 18 bars are specified, it can be seen that this system is not rigid because each banana can rotate about the axis defined by the two end points A and B connecting them.

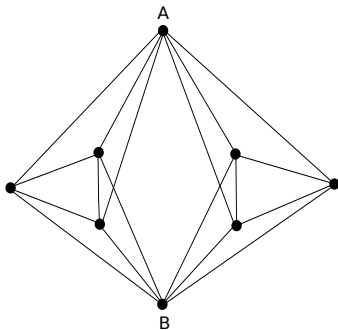


Figure 1: Geometry of the double banana.

We begin in section 2 by giving a series of tools used to describe the geo-

metric problem. Section 3 describes the coordinate free formulation applied to this GCSP ¹ and presents the solving method. In section 4, a method for analyzing whether the geometric problem is overconstrained or not, is described. Following this, if the problem is declared over-constrained, we propose a method for assisting a design team to seek a consistent set of constraints. Finally, section 5 uses the example of the double banana to illustrate the application of this methodology.

Remark: the Einstein notation or Einstein summation convention is employed in this paper. This notation implies the summation over a set of indexed terms in a formula, thus achieving notational brevity.

2. A coordinate free-model for representing the geometry

This section presents a non-cartesian model that characterizes geometric objects. We recall here the basics of the method as described in [5]. The characteristic of this approach is that it is unnecessary to perform cartesian reference frame, which is a real benefit for the sketching tasks of CAD designers. The principle is that any geometry can be represented as points and vectors. These are the central elements of our modelling. In the following, we describe three models that fully describe the design geometry.

2.1. The topological model

The geometry is reduced to a skeleton composed of points and line segments. An incidence matrix \mathbf{C} , establishes the relation between each point of an object and its edges. It is an $n \times m$ matrix, where n and m are the number of edges and vertices respectively, such that $C_i^j = -1$ if the edge e_j leaves vertex p_j , 1 if it enters vertex p_j and 0 otherwise. Figure 2 gives an example of an incidence matrix. Edges are oriented arbitrarily.

2.2. The geometrical model

The geometrical object is closely related to the topological model. Indeed, a vector is associated with each edge. Therefore each vector is an oriented bipoint. Thus the geometrical model is represented by a list of vectors.

Given \mathcal{V} a set of n non-normed vectors as $\mathcal{V} = (\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n)$. The Gram matrix is the mathematical tool used to represent this vectorial model. Thus

¹Geometric Constraint Satisfaction Problem

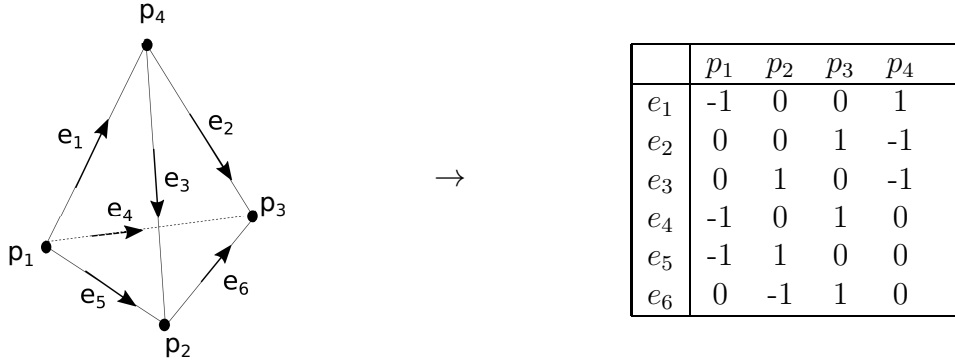


Figure 2: Geometric skeleton (left) and associated incidence \mathbf{C} matrix (right)

$\mathbf{G}(\mathcal{V})$ is defined by: $G_{ij}(\mathcal{V}) = \langle \mathbf{v}_i, \mathbf{v}_j \rangle$ where $\langle \mathbf{v}_i, \mathbf{v}_j \rangle$ is the scalar product between vectors \mathbf{v}_i and \mathbf{v}_j . This matrix is also noted \mathbf{G} in the following.

The Gram matrix of a set of vectors is constructed for each representation of a sketch. This Gram matrix fully defines the metrics of the object. It should be noted that this modelling is independent of the Cartesian coordinate system since all the vectors are defined in relation to one another: the vectors are not represented by their Cartesian co-ordinates but by their relative scalar products. An advantage of this approach is the possibility of ensuring specification consistency by verifying the mathematical properties of the Gram matrix (symmetrical, positive-semidefinite, rank, etc.). For example, by calculating specific determinants, it is possible to know whether or not there is a solution to the problem (see [10]).

2.3. The specification model

For our purpose, it is assumed that a geometric problem is defined by a skeleton, totally defined by the topological and geometrical models, and by a list of geometric constraints. In this study, we only focus on the length of the vectors, called L , or the angle between two vectors called α . All the specifications are stocked in \mathbf{S} . It is a partially filled Gram matrix.

Element S_{ii} is known if the user chooses a specific length for \mathbf{v}_i , as presented in equation 2.

$$L_i = \sqrt{S_{ii}} \quad (2)$$

Element S_{ij} is defined if the user specifies the angle between \mathbf{v}_i and \mathbf{v}_j . Thus,

$$\cos(\alpha_{ij}) = \frac{S_{ij}}{\sqrt{S_{ii}}\sqrt{S_{jj}}} \quad (3)$$

These three models fully characterize the GCSP.

3. A coordinate free formulation of geometric constraints

The previous section presented the geometric model based on a vectorial representation of the geometric entities and constraints. We now present the approach developed, based on the transformation of the initial geometry and using a perturbation matrix (see equation 1). This method is particularly appropriate for CAD as the users always start by "drawing" an initial shape that is gradually modified to obtain the desired object.

3.1. The perturbation of the initial object

To illustrate this method, let us take the trivial object presented in figure 3. For example, \mathbf{v}_1 has p_1 as origin and p_4 as end vertex. The $\mathbf{G}(\mathcal{V})$ matrix defines the initial object whereas $\mathbf{G}(\mathcal{V}')$ characterizes the final object. As the

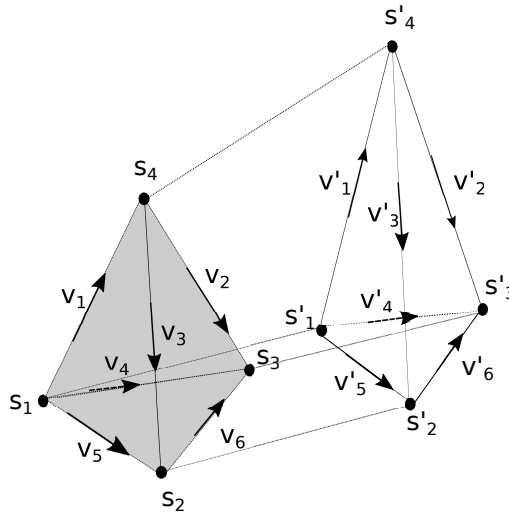


Figure 3: Vertex displacement

\mathbf{C} incidence matrix forms the relation between each point of the object and

vectors, we express the edge relation for the initial object in (4).

$$\mathbf{v}_i = C_i^j \mathbf{p}_j \quad (4)$$

The edge relation for the final object is found in (5).

$$\mathbf{v}'_i = C_i^j \mathbf{p}'_j \quad (5)$$

It should be noted that the two objects are represented by the same connection matrix \mathbf{C} because the topology is invariant during the transformation. The variation $\mathbf{p}'_i - \mathbf{p}_i$ corresponds to the vertex displacement. Here \mathbf{p}_i and \mathbf{p}'_i are vectors which are the positions of the vertex i before and after transformation. If \mathcal{V} is a set of vectors and \mathcal{V}' a set of vectors obtained after a transformation from an initial state to a final one, we express the relation between these two states. We define the vertex variation by (6):

$$\mathbf{p}'_i - \mathbf{p}_i = X_i^j \mathbf{v}_j \quad (6)$$

X_i^j are elements of matrix \mathbf{X} , which is called the perturbation matrix from the initial state of the geometry to the final one. It is an $n \times m$ matrix. The last three equations (4),(5) and (6), make it possible to obtain the relation between the initial vectors and the final one in (9) using formulas (7) and (8). Here, δ_i^k represents the Kronecker symbol which equals 1 if $i = k$ and, 0 otherwise.

$$\mathbf{v}'_i = C_i^j (X_j^k \mathbf{v}_k + \mathbf{p}_j) \quad (7)$$

$$\mathbf{v}'_i = \mathbf{v}_i + C_i^j X_j^k \mathbf{v}_k \quad (8)$$

$$\mathbf{v}'_i = (\delta_i^k + C_i^j X_j^k) \mathbf{v}_k \quad (9)$$

The aim is to obtain the transformation of an object from its initial shape to its final one. The Gram matrix $\mathbf{H} = \mathbf{G}(\mathcal{V}')$ characterizes the final object. Indeed,

$$H_{ij} = \langle \mathbf{v}'_i, \mathbf{v}'_j \rangle \quad (10)$$

$$H_{ij} = \langle (\delta_i^p + C_i^k X_k^p) \mathbf{v}_p, (\delta_j^q + C_j^k X_k^q) \mathbf{v}_q \rangle \quad (11)$$

$$H_{ij} = (\delta_i^p + C_i^k X_k^p) (\delta_j^q + C_j^k X_k^q) \langle \mathbf{v}_p, \mathbf{v}_q \rangle \quad (12)$$

$$H_{ij} = (\delta_i^p + C_i^k X_k^p) (\delta_j^q + C_j^k X_k^q) G_{pq} \quad (13)$$

This expression written in matrix form is:

$$\mathbf{H} = (\mathbf{I} + \mathbf{C} \mathbf{X}) \mathbf{G} (\mathbf{I} + \mathbf{C} \mathbf{X})^t \quad (14)$$

Note \mathbf{T} the vectorial transformation $\mathbf{I} + \mathbf{C} \mathbf{X}$. Due to the rank property: $rank(\mathbf{H}) \leq \min(rank(\mathbf{T}), rank(\mathbf{G}))$ \mathbf{H} is a matrix of rank r . Formula (14) above represents the variation of the geometric object. The elements of the perturbation matrix \mathbf{X} are the unknowns of the problem. So far, the Gram matrix \mathbf{G} has only been used to characterize the position and orientation of all the vectors of the initial sketch. Numerically, this matrix is totally known.

3.2. The specifications

The geometric constraints specified by the user lead to a set of equations that are presented in the following section. Here we itemize only two types of specifications (useful as examples) but many other specifications have been developed by the authors. The reader may refer to [5] and [11] for more details.

3.2.1. Length equations

By definition, an element of diagonal matrix G_{ii} (H_{ii} for the final one) represents the squared length of an initial vector (final resp.). The length specification of vector \mathbf{v}_i imposed by the designer is denoted L_i . Thus:

$$\sqrt{H_{ii}} = L_i \quad (15)$$

Given that each final matrix element is expressed as equation 13, and by developing this latter expression to the 1st order, we obtain (16):

$$H_{ij} \cong G_{ij} + (C_i^k X_k^p)G_{pj} + G_{ip}(C_j^k X_k^p) \quad (16)$$

In particular, for the formula of the diagonal elements (17) can be used.

$$H_{ii} \cong G_{ii} + 2(C_i^k X_k^p)G_{pi} \quad (17)$$

And by developing equation (15), it becomes equation (18). This equation is a set of linear equations in the unknowns X_k^p .

$$\frac{1}{\sqrt{G_{ii}}}(C_i^k X_k^p)G_{pi} \cong L_i - \sqrt{G_{ii}} \quad (18)$$

Terms L_i , G_{pi} , C_i^k and G_{ii} are known. The unknowns of this equation are elements of \mathbf{X} .

3.2.2. Line-line angle equations

By definition, element G_{ij} represents the scalar product between vectors \mathbf{v}_i and \mathbf{v}_j . The designer requires the cosine of the angle between vector \mathbf{v}_i and \mathbf{v}_j to satisfy specification α_{ij} , resulting in the equation (19) below:

$$H_{ij} = \sqrt{H_{ii}}\sqrt{H_{jj}} \cos \alpha_{ij} \quad (19)$$

Rewriting equation (19) to move all the terms of \mathbf{H} to the left of the equal sign gives:

$$H_{ij}(H_{ii})^{-1/2}(H_{jj})^{-1/2} = \cos \alpha_{ij} \quad (20)$$

By developing H_{ij} , H_{ii} and H_{jj} using equation (16) and by linearization to the 1st order, the expression of an angular specification is expressed as follows:

$$\left(G_{ip} - \frac{G_{ij}}{G_{jj}} G_{pj} \right) C_j^k X_k^p + \left(G_{pj} - \frac{G_{ij}}{G_{ii}} G_{pi} \right) C_i^k X_k^p \cong \sqrt{G_{ii}}\sqrt{G_{jj}} \cos \alpha_{ij} - G_{ij} \quad (21)$$

Terms G_{ip} , G_{pj} , G_{ii} , G_{ij} , G_{jj} , C_j^k , C_i^k and $\cos(\alpha_{ij})$ are known. The unknowns are elements of \mathbf{X} . Equation (21) is a set of linear equations in the unknowns X_k^p .

3.3. Geometric problem solving

In this section, we do not attempt to describe in detail the method used to solve the equations for the isoconstrained problem. This study was carried out by Moinet in [11] for a different set of equations but it remains well-adapted to the problem presented in this paper. The geometrical solver developed with *Matlab*[®] computes the \mathbf{X} perturbation matrix of equation (14) in order to obtain an object that conforms to the specifications given by the designers. The system to be solved consists of non-linear relations (15) and (19). The known terms are the elements of the initial Gram matrix \mathbf{G} , line-line angle specifications α_{ij} , and length specifications L_i . For solving purposes, the equations generated by the above mentioned solver are linearized to the 1st order. Then it is necessary to find the elements of perturbation matrix X_{ij} to solve the problem. The most frequently used method to solve this kind of system, $F(X) = 0$, is the Newton-Raphson method. However, the algorithm implemented is a variation of the Newton-Raphson method that has been described in [5]. Then equations (18) and (21), as

presented in previous paragraph, can be rewritten and stored in the linear form presented in (22).

$$\mathbf{Ax} = \mathbf{b} \tag{22}$$

In this equation, all the unknown elements from matrix \mathbf{X} are stored in the column vector noted \mathbf{x} . The elements of matrix \mathbf{A} and the second member \mathbf{b} (a column vector) are computed from \mathbf{S} , \mathbf{G} and \mathbf{C} . Note that there are as many elements in \mathbf{b} as the number of specifications given by the user. Indeed, these elements are extracted from matrix $\mathbf{S} - \mathbf{G}^{(i)}$ at each step i . In this case, \mathbf{S} is the specification given by the designer, whereas $\mathbf{G}^{(i)}$ is the Gram matrix at step i . The initial value, $\mathbf{G}^{(0)} = \mathbf{G}$, is given by the measure of the geometric parameter in the initial sketch. Then, the solving method can be summed up as in algorithm 1:

Algorithm 1: How to solve the GCSP problem**Input:**

\mathbf{G} is an initial object defined by a Gram matrix of size $n \times n$

\mathbf{C} is the connection matrix of size $n \times m$

\mathbf{S} are the specifications given by the user, defined by a Gram matrix of size $n \times n$ partially filled.

n : number of vectors

m : number of points in the sketch

d : number of specifications

Result:

A new Gram matrix \mathbf{G} of a rank equal to 3 that totally defines the final object and conforms to the list of specifications initialization;

$\mathbf{G}^{(0)} \leftarrow \mathbf{G}$;

$\mathbf{X} \leftarrow \emptyset$ (of size $m \times n$) ;

while *Not all the specifications are satisfied up to ϵ* **do**

Construct a system in the form $\mathbf{A}\mathbf{x} = \mathbf{b}$ using this relation:

$$\mathbf{C}\mathbf{X}\mathbf{G}^{(i)} + \mathbf{G}^{(i)}(\mathbf{C}\mathbf{X})^t = \mathbf{S} - \mathbf{G}^{(i)}$$

with:

\mathbf{A} matrix (of size $d \times (nm)$)

\mathbf{x} vector (of size $nm \times 1$); $\mathbf{x} = \text{vec}(\mathbf{X}^t)$

\mathbf{b} vector (of size $d \times 1$) equals to the d elements extracted from the column vector formed by stacking the lines of $\mathbf{S} - \mathbf{G}^{(i)}$ matrix

$\mathbf{x} \leftarrow \text{pinv}(\mathbf{A})\mathbf{b}$; pinv is the pseudo-inverse of *Matlab*[®]

Re-construct \mathbf{X} matrix

$$\mathbf{G}^{(i+1)} \leftarrow (\mathbf{I} + \mathbf{C}\mathbf{X})\mathbf{G}^{(i)}(\mathbf{I} + \mathbf{C}\mathbf{X})^t$$

end

This algorithm converges if the GCSP is underconstrained, isoconstrained or overconstrained with consistent specifications. Therefore in the next section the authors present a method that always ensures consistency of the GCSP.

An application of this algorithm is presented in section 5.

4. The Problem Analysis

When, for generic values of the GCSP, the number of solutions is finite and nonzero (resp. zero, infinity), the problem is said to be *well-* (resp. *over-, under-*) *constrained*.

Here, we only focus on the overconstrained problem and propose using the witness method of Michelucci et al. [12, 13, 14] to perform the problem analysis. The first part of this section will briefly introduce Michelucci's witness method. In the second part, we explain how to generate a useful witness using a coordinate free formulation for the geometric constraint solving problem. Then, the third part introduces the proposed method to analyse the chosen witness using the Rouché-Fontené theorem. The fourth part proposes a method to handle over-constrained problems.

4.1. The Witness method

Michelucci's witness method proposes studying the generic properties of a collection of objects by studying one of them that is called a "witness". Up to now this method has always been used in the context of constraints formulated in a Cartesian framework. In addition, the literature states that if there is a sketch with the same properties as the solution sought (coincidence, collinearity, coplanarity, etc.) then this is generally a witness. In the framework of this research, it is always assumed that the user has drawn such a sketch to describe a problem of geometrical constraints. This is why this method seems well-adapted to this task. It should be noted that the witness method has never been applied with non-Cartesian formulations. The authors therefore felt it pertinent to demonstrate that this method can be applied in this case.

4.2. Generation of a generic witness

Sometimes the sketch proposed by a designer is not totally representative of the shape of the desired solution. This occurs when certain geometric elements are drawn with specific properties (collinearity, coplanarities, etc.) without representing a real constraint. In this case the sketch cannot be used as a witness. This situation occurs when the sketch is not generic. A sketch is considered generic when it remains rigid before and after an infinitesimal perturbation. Likewise if the sketch was not rigid before the perturbation and continues not being rigid after it. Details on rigidity and generic frameworks can be found in [16]. This situation is illustrated in figure 4. The rigid sketch

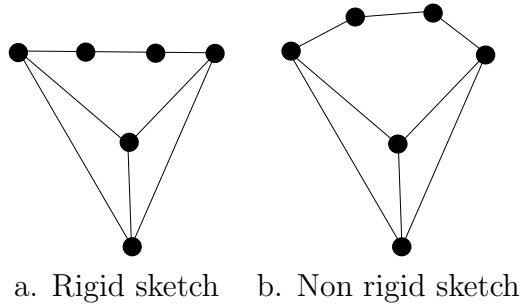


Figure 4: Example of rigid and non rigid sketches. [15]

(figure 4.a) is not generic: a small perturbation on the dimensions of the bars will result in a non rigid sketch like that shown in figure 4.b. On the other hand the sketch in figure 4.b is generic: if a small perturbation is introduced in the dimension of this sketch, it will remain non rigid. Usually, non generic sketches are constituted with line segments that are aligned as presented in figure 4.a. In the representation chosen (based on Gram matrix) for this paper, the vectors representing these line segments are collinear or coplanar. The collinearity of the vectors will lead to an artificial redundancy between the constraints associated with the collinear vectors. Consequently, before using the witness method with a coordinate free formulation of the GCSP, it is necessary to ensure that the sketch used as witness is generic. In order to ensure that the sketch is generic, the authors have used a naive technique that consists of the algorithm presented in figure 5. This algorithm has four steps:

1. The system of equations modelling the GCSP is based on the user's sketch. The rank of this system is calculated.
2. The starting point is a random perturbation of the sketch. Then a second linearised system is generated and its rank calculated.
3. If the rank of the second system is higher than that of the first, the first sketch is replaced by the second and the second step is reiterated.
4. Otherwise the first sketch is considered generic.

It is important to note that the probability of finding a perturbation at random that transforms a generic sketch into a non generic sketch is very low. This means that, apart from exceptional cases, the rank will not be reduced by the random perturbation. This subject will be discussed in section 6.

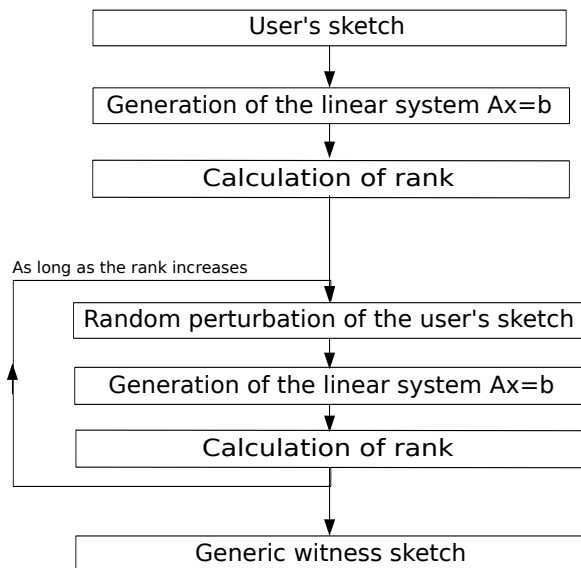


Figure 5: Naive algorithm for the generation of a generic sketch

In our case, the initial problem is represented by a matrix of incidence \mathbf{C} and a list of points with their initial positions (described by their Cartesian coordinates). Thus, to generate a random perturbation of the initial problem, it is advisable to conserve the matrix of incidence and randomly perturb the initial coordinates of all the points.

4.3. Analysing the witness

Once the witness has been generated, it can be analysed. In the second case of an over-constrained problem, the purpose of this analysis is to determine which constraints are redundant. Overconstrained problems that are locally underconstrained do not disturb the proposed methodology of this article. For example, system (23) holds this characteristic. Furthermore, the analysis proposed by the authors also permits determining, for each redundant constraint, the relation that links this constraint to those that are independent. The main steps of the method proposed are presented in figure 6. In this paragraph we assume that the system of equations is written in linear form: $\mathbf{Ax} = \mathbf{b}$ (cf. section 3.3). It should be noted that matrix \mathbf{A} has the dimension $d \times nm$. We also assume that this system is composed of r independent constraints (the matrix \mathbf{A} has rank r). As the GCSP is

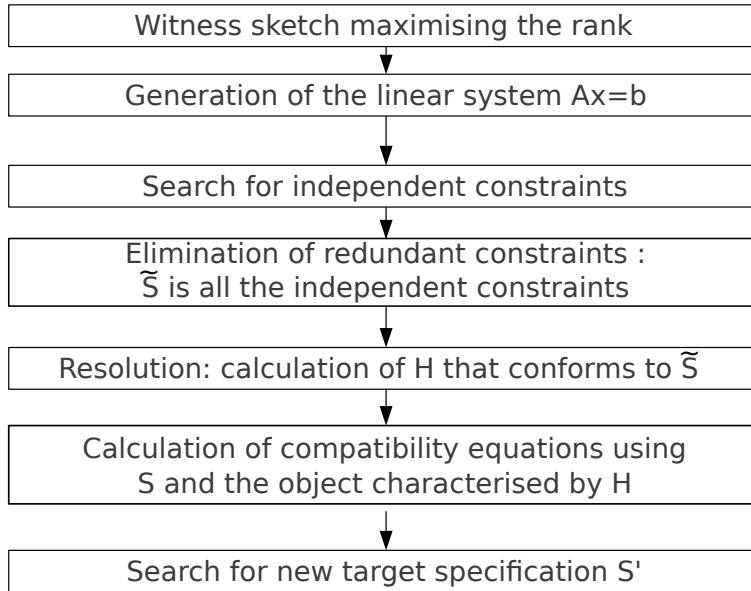


Figure 6: Method used to analyse the over-constrained problem

overconstrained, it is additionally assumed that $d > r$ and $nm \geq r$.

4.3.1. Search for redundant constraints

To find the redundant constraints, the authors propose using the Gauss elimination method [17] to obtain a triangular form of the matrix \mathbf{A} . This triangular form is used to identify the redundant constraints by searching the lines that contain only 0. These redundant constraints are withdrawn from the problem. The independent specifications are stored in $\tilde{\mathbf{S}}$, a partially filled Gram matrix. Comment: if a column of the matrix does not contain an elimination value, this means that the value of the unknown associated

with this column can be fixed freely.

$$\left\{ \begin{array}{l} x_1 + x_4 = b_1 \\ x_2 + x_4 = b_2 \\ x_3 + x_4 = b_3 \\ x_1 + x_2 + 2x_4 = b_4 \\ x_2 + x_3 + 2x_4 = b_5 \\ x_5 = b_6 \\ x_6 = b_7 \\ x_5 + x_6 = b_8 \end{array} \right. \quad (23)$$

For example, the system (23) can be represented by the matrix shown in (24) figure 7.a. This matrix can be given a triangular form (25) (in figure 7.c) by using the circled values and by performing the sequence of operations indicated in figure 7.b on its lines: Using this example, we can conclude

$\begin{bmatrix} \textcircled{1} & 0 & 0 & 1 & 0 & 0 \\ 0 & \textcircled{1} & 0 & 1 & 0 & 0 \\ 0 & 0 & \textcircled{1} & 1 & 0 & 0 \\ 1 & 1 & 0 & 2 & 0 & 0 \\ 0 & 1 & 1 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & \textcircled{1} & 0 \\ 0 & 0 & 0 & 0 & 0 & \textcircled{1} \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \quad (24)$	<ol style="list-style-type: none"> 1. $L_4 \leftarrow L_4 - L_1$ 2. $L_4 \leftarrow L_4 - L_2$ 3. $L_5 \leftarrow L_5 - L_2$ 4. $L_5 \leftarrow L_5 - L_3$ 5. $L_8 \leftarrow L_8 - L_6$ 6. $L_8 \leftarrow L_8 - L_7$ 	$\begin{bmatrix} \textcircled{1} & 0 & 0 & 1 & 0 & 0 \\ 0 & \textcircled{1} & 0 & 1 & 0 & 0 \\ 0 & 0 & \textcircled{1} & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \textcircled{1} & 0 \\ 0 & 0 & 0 & 0 & 0 & \textcircled{1} \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (25)$
a. Initial matrix	b. Operations on the lines	c. Triangular matrix

Figure 7: Illustration of the Gauss elimination method

that equations 4, 5 and 8 are redundant. We can also conclude that the 4th unknown can be chosen freely since the fourth column of the matrix (25) does not contain a Gauss value.

4.3.2. Determining the equations of compatibility

If the system $\mathbf{Ax} = \mathbf{b}$ is over-constrained the Rouché-Fontené theorem [17, 18] indicates that for a solution to exist, the second member \mathbf{b} must conform to $d - r$ conditions. In the following, these conditions will be called *equations of compatibility*. It is always possible to rewrite the system without

loss of generality so that the first r lines and the first r columns are independent. This amounts, on the one hand, to writing the redundant constraints on the last lines, and on the other hand, to rewriting the unknowns that can be fixed freely at the end (which corresponds to the last columns of the matrix). Thus the system $\mathbf{Ax} = \mathbf{b}$ will take the form presented in equation (26). Then, the expression of each compatibility equation is given by $h_q = 0$. The expression of h_q is given in equation (27) with $q \in \{1, \dots, d - r\}$.

$$\begin{bmatrix} A_1^1 & \dots & A_1^r & A_1^{r+1} & \dots & A_1^{nm} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ A_r^1 & \dots & A_r^r & A_r^{r+1} & \dots & A_r^{nm} \\ A_{r+1}^1 & \dots & A_{r+1}^r & A_{r+1}^{r+1} & \dots & A_{r+1}^{nm} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ A_d^1 & \dots & A_d^r & A_d^{r+1} & \dots & A_d^{nm} \end{bmatrix} \times \begin{bmatrix} x_1 \\ \vdots \\ x_r \\ x_{r+1} \\ \vdots \\ x_{nm} \end{bmatrix} = \begin{bmatrix} b_1 \\ \vdots \\ b_r \\ b_{r+1} \\ \vdots \\ b_d \end{bmatrix} \quad (26)$$

$$h_q = \begin{vmatrix} b_1 & A_1^1 & \dots & A_1^r \\ \vdots & \vdots & \ddots & \vdots \\ b_r & A_r^1 & \dots & A_r^r \\ b_{r+q} & A_{r+q}^1 & \dots & A_{r+q}^r \end{vmatrix} = 0 \quad (27)$$

It is important to note that in equation (27) variables A_i^j have known numerical values. On the contrary, the objective of this calculation is to obtain a linear relation that links variables b_i in the form presented in equation (28).

$$h_q = \sum_{i \in \{1, \dots, r, r+q\}} \gamma_q^i b_i \quad (28)$$

To explain this relation it is possible to expand the determinant of equation (27) by the first column. We therefore obtain the value of the coefficients γ_q^i by the formula (29).

$$\gamma_q^i = \begin{cases} (-1)^{i+1} \begin{vmatrix} A_1^1 & \dots & A_1^r \\ \vdots & \ddots & \vdots \\ A_{i-1}^1 & \dots & A_{i-1}^r \\ A_{i+1}^1 & \dots & A_{i+1}^r \\ \vdots & \ddots & \vdots \\ A_r^1 & \dots & A_r^r \\ A_{r+q}^1 & \dots & A_{r+q}^r \end{vmatrix} & \text{if } i \in \{1, \dots, r, r+q\}, \\ 0 & \text{otherwise} \end{cases} \quad (29)$$

4.4. Handling an over-constrained problem

In the case where the problem is over-constrained, i.e. there is no solution that conforms to all the specifications \mathbf{S} , the authors propose a method to help designers to find all the consistent specification values bringing them close to the initial expectations. The idea is to use the compatibility equations to change the independent specifications so that the redundant specifications draw close to the value desired. To do this the authors propose searching a set of new specifications stored in the Gram matrix \mathbf{S}' . The specification values must conform to the compatibility equation (30):

$$\mathbf{\Gamma}\mathbf{t} = 0 \quad (30)$$

We define \mathbf{t} , the vector of size $(d \times 1)$ equal to the d elements extracted from the column vector formed by stacking the lines of $(\mathbf{S}' - \mathbf{H})$. The expression of terms γ_q^i of matrix $\mathbf{\Gamma}$ is given by equation (29). In a general way, we note that matrix $\mathbf{\Gamma}$ has size $(d - r) \times d$. In most engineering cases, the number of compatibility equations is much lower than the number of independent specifications ($d - r < r$). When $d - r \geq r$ the designers must have imposed a very high number of redundant constraints ($d \geq 2r$). In this specific but improbable case, the designers would be asked to reformulate their problem. Furthermore, in the general case where $d - r < r$, matrix $\mathbf{\Gamma}$ will have rank $d - r$ since the compatibility equations are independent by construction. Consequently, by using system (30) it is possible to calculate $d - r$ values of t_i among d . If, without loss of generality, the columns of matrix $\mathbf{\Gamma}$ are re-ordered so that its first $d - r$ columns are independent, system (30) can be written in the form (31) and (32) by introducing two matrices $\mathbf{\Gamma}_1$ (of size $(d - r) \times (d - r)$) and $\mathbf{\Gamma}_2$ (of size $(d - r) \times r$). In this form matrix $\mathbf{\Gamma}_1$ is invertible and we can calculate the values t_1 to t_{d-r} using the formula (33).

$$\begin{bmatrix} \gamma_1^1 & \dots & \gamma_1^{d-r} \\ \vdots & \ddots & \vdots \\ \gamma_{d-r}^1 & \dots & \gamma_{d-r}^{d-r} \end{bmatrix} \times \begin{bmatrix} t_1 \\ \vdots \\ t_{d-r} \end{bmatrix} + \begin{bmatrix} \gamma_1^{d-r+1} & \dots & \gamma_1^d \\ \vdots & \ddots & \vdots \\ \gamma_{d-r}^{d-r+1} & \dots & \gamma_{d-r}^d \end{bmatrix} \times \begin{bmatrix} t_{d-r+1} \\ \vdots \\ t_d \end{bmatrix} = 0 \quad (31)$$

$$\mathbf{\Gamma}_1 \times \begin{bmatrix} t_1 \\ \vdots \\ t_{d-r} \end{bmatrix} + \mathbf{\Gamma}_2 \times \begin{bmatrix} t_{d-r+1} \\ \vdots \\ t_d \end{bmatrix} = 0 \quad (32)$$

$$\begin{bmatrix} t_1 \\ \vdots \\ t_{d-r} \end{bmatrix} = -\mathbf{\Gamma}_1^{-1}\mathbf{\Gamma}_2 \times \begin{bmatrix} t_{d-r+1} \\ \vdots \\ t_d \end{bmatrix} \quad (33)$$

Now the above has been presented, the problem remaining to be solved is to partition \mathbf{t} so that the $\mathbf{\Gamma}_1$ matrix is full rank.

4.4.1. Partitioning the set of target specifications

The technique presented in section 4.3.1 allows determining the redundant constraints. It also provides a set of independent target specifications. However, in an engineering context, the designer may want to conserve the initial value for the redundant specification detected by the Gauss elimination method. In this case, it is necessary to find another partition for the target specifications. The following proposition does not claim to solve the problem of partitioning the set of specification parameters. The two ideas raised here are more the paths to be followed to give a clear framework for designers to choose a partition of t_i . Firstly, the designers may wish to assign priorities or importances to specifications: the solution must meet the important specifications as precisely as possible, whereas the accuracy for other specifications can be lower. In this case it is advisable to place the important specifications in subassembly (t_{d-r+1}, \dots, t_d) and the less important specifications in subassembly (t_1, \dots, t_{d-r}) which will be calculated using relation (33). Secondly, it could also be interesting to follow paths suggested in research works on robust design such as [19]. This study indicates criteria to help designers to divide their specifications into two parts to minimize the variation of values (t_1, \dots, t_{d-r}) that are calculated from (t_{d-r+1}, \dots, t_d) which is also our aim. Whatever the case, once the designers have chosen a partition, it is advisable that matrix $\mathbf{\Gamma}_1$ is invertible.

4.4.2. Discussion

On the basis of the compatibility equations (30) generated, the authors present two paths that should be combined carefully to permit a team of designers to solve an overconstrained problem. The paths proposed in this section to calculate the values of the target specifications \mathbf{t} using matrix $\mathbf{\Gamma}$ require relatively basic linear algebra functions. This means that the calculations necessary to obtain the target specifications \mathbf{t} will be fast. A simple tool for calculating the target specifications used in these two paths could be developed. Then it would be possible to use this tool during a design review

to generate several sets of target specifications. During the same meeting, the design team and experts could then decide on which target value seems most pertinent. However, it is important to note that the compatibility equations were generated using a linearised version of the problem. Consequently, these equations are only valid in the vicinity of the solution used to generate them.

5. Case study

This section presents a case study with the application of the method to a double banana configuration. The skeleton of the double banana is composed of 8 vertices and 18 edges. These 26 elements are declared in the text file presented in figure 8. The initial geometry is also declared in the same text file by the position of 8 points associated with each vertex. This over-constrained and under-constrained academic example is a good candidate for illustrating the proposed algorithm. The declaration of the geometrical

```

#!VERTICES 8
Point: Pt1 (0.0000000000 ; 0.0000000000 ; 0.0000000000)
Point: Pt2 (-75.0000000000 ; 0.0000000000 ; 0.0000000000)
Point: Pt3 (-10.2666666667 ; 23.8871420550 ; -0.0000000000)
Point: Pt4 (-19.3801129874 ; -2.4055069381 ; 19.5356521540)
Point: Pt5 (22.9570528818 ; 19.9141617064 ; 37.1537331437)
Point: Pt6 (-34.3090195745 ; 59.7807966350 ; 15.7003332989)
Point: Pt7 (-33.9051208042 ; 17.3142000308 ; 40.1426130337)
Point: Pt8 (-19.4098849778 ; 29.6808580743 ; 25.5496727141)

#!EDGES 18
Edge: A1 (-Pt2;Pt1)
Edge: A2 (-Pt2;Pt3)
Edge: A3 (-Pt2;Pt4)
Edge: A4 (-Pt5;Pt1)
Edge: A5 (-Pt5;Pt3)
Edge: A6 (-Pt5;Pt4)
Edge: A7 (-Pt5;Pt6)
Edge: A8 (-Pt5;Pt7)
Edge: A9 (-Pt5;Pt8)
Edge: A10 (-Pt2;Pt8)
Edge: A11 (-Pt2;Pt7)
Edge: A12 (-Pt2;Pt6)
Edge: A13 (-Pt6;Pt7)
Edge: A14 (-Pt8;Pt6)
Edge: A15 (-Pt7;Pt8)
Edge: A16 (-Pt3;Pt4)
Edge: A17 (-Pt1;Pt3)
Edge: A18 (-Pt4;Pt1)

#!LENGTH SPECIFICATIONS 18
Length: l1 (A1) = 75
Length: l2 (A2) = 69
Length: l3 (A3) = 59
Length: l4 (A4) = 48
Length: l5 (A5) = 50
Length: l6 (A6) = 51
Length: l7 (A7) = 73
Length: l8 (A8) = 57
Length: l9 (A9) = 45
Length: l10 (A10) = 68
Length: l11 (A11) = 60
Length: l12 (A12) = 74
Length: l13 (A13) = 49
Length: l14 (A14) = 35
Length: l15 (A15) = 24
Length: l16 (A16) = 34
Length: l17 (A17) = 26
Length: l18 (A18) = 32

```

Figure 8: Text file - example of the Double Banana geometry.

elements composing the double banana geometry is done as follows.

- First, the number indicated after the key word 'VERTICES' gives the total number of points in the geometry. Then each line that follows is used to declare each single point. This declaration starts with the keyword 'Point'. Then, the name of the point, such as $Pt1$, $Pt2$, \dots in Figure 8 and the initial coordinates of the point must be entered. This part provides the information needed to fill the initial Gram matrix, $\mathbf{G}^{(0)}$ presented in algorithm 1.
- Second, the number indicated after the key word 'EDGES' gives the total number of edges in the geometry. Then each line that follows is used to declare each single edge. This declaration starts with the keyword 'Edge'. Then, the name of the edge, such as $A1$, $A2$, \dots in Figure 8 must be entered. Afterwards, the name of the origin point of the current edge is preceded by a minus sign whereas no sign specifies the ending point. Consequently, all the edges are oriented. This part corresponds to providing the \mathbf{C} information presented in algorithm 1.
- Third, the number indicated after the key word 'LENGTH SPECIFICATIONS' gives the total number of length constraints declared in the GCSP. Each constraint is declared using the approach presented before. In this case, they are named $l1$ to $l18$ with their corresponding numerical values specified after the equal sign. This part gives the specifications contained in the partially filled Gram matrix \mathbf{S} .

The figure in Fig.9 is drawn using vertices coordinates (listed in Fig.8); they do not fulfill length specifications (also in fig.8). The resolution of the GCSP is assigned to the solver developed by the authors [5]. The result expected is a geometry that conforms to all the 18 specifications given. As this case study is over-constrained, the solver is unable to find a geometry that satisfies the 18 specifications. Consequently, this research proposes to add a numerical problem analysis before solving the GCSP. The result of this analysis gives a set of compatibility equations required to find a new set of consistent specifications stored in \mathbf{S}' .

5.1. Analysis of the Numerical Problem

The numerical analysis of the problem is performed by adapting the witness method to the case of the double banana structure. The first part is to find a generic sketch. A linear system is generated from the initial configuration. Its rank is calculated as 17. The points of the initial sketch are

From this point we can first observe that the 18th row is full of 0, meaning that the 18th specification is redundant. Secondly, it can be seen that the columns numbered $\{12, 18, 20, 21, \dots, 144\}$ do not contain any elimination value which means that the value of unknowns $\{12, 18, 20, 21, \dots, 144\}$ can be chosen freely. This second feature is not considered in the method presented.

$$\begin{bmatrix}
 \textcircled{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.54309 & 0 & 0 & 0 & 0 & 0 & -0.63836 & 0 & -0.87313 & \dots & 0.904 \\
 0 & \textcircled{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.5846 & 0 & 0 & 0 & 0 & 0 & 0.74544 & 0 & 1.8806 & \dots & -1.309 \\
 0 & 0 & \textcircled{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1.1119 & 0 & 0 & 0 & 0 & 0 & 0.4288 & 0 & 0 & \dots & -1.2757 \\
 0 & 0 & 0 & \textcircled{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2.0694 & 0 & 4.8563 & \dots & -2.699 \\
 0 & 0 & 0 & 0 & \textcircled{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1.1408 & 0 & -4.7575 & \dots & -0.82269 \\
 0 & 0 & 0 & 0 & 0 & \textcircled{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1.0271 & 0 & 0 & \dots & 3.0168 \\
 0 & 0 & 0 & 0 & 0 & 0 & \textcircled{1} & 0 & 0 & 0 & 0 & 0 & -2.0841 & 0 & 0 & 0 & 0 & 0 & -0.87694 & 0 & -0.7622 & \dots & 1.786 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & \textcircled{1} & 0 & 0 & 0 & 0 & 1.5431 & 0 & 0 & 0 & 0 & 0 & 0.53983 & 0 & 0.97193 & \dots & -1.4089 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \textcircled{1} & 0 & 0 & 0 & 0.78228 & 0 & 0 & 0 & 0 & 0 & 0.75185 & 0 & 0 & \dots & -2.3414 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \textcircled{1} & 0 & 0 & 0.84097 & 0 & 0 & 0 & 0 & 0 & 0.32545 & 0 & 0.67516 & \dots & -0.8287 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \textcircled{1} & 0 & -0.0076529 & 0 & 0 & 0 & 0 & 0 & -0.096649 & 0 & 0.16532 & \dots & -0.060778 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \textcircled{1} & 0 & 0 & 0 & 0 & 0 & 0 & -1.5895 & 0 & -3.1882 & \dots & 1.8921 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \textcircled{1} & 0 & 0 & 0 & 0 & 0.70222 & 0 & 3.9125 & \dots & 0.50639 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \textcircled{1} & 0 & 0 & 0 & 1.3099 & 0 & 0 & \dots & -3.9679 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \textcircled{1} & 0 & 0 & 0.86343 & 0 & 0.66885 & \dots & -0.67273 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \textcircled{1} & 0 & -0.28511 & 0 & -1.7211 & \dots & -0.2056 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \textcircled{1} & 0 & 1.3945 & \dots & -0.46582 \\
 0 & \dots & 0
 \end{bmatrix}
 \tag{35}$$

Consequently to have a well-posed problem, the proposed solution is to release the 18th constraint. The new reduced problem is therefore composed of 17 specifications which are stored in $\tilde{\mathbf{S}}$. This new well-posed problem, is sent to the solver. A solution is found in 4 iterations. In figure 10, we can see all the configurations of the objects obtained during the solving. An object

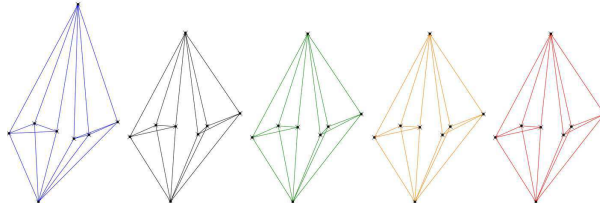


Figure 10: Four iterations of the double banana (left initial, right final object).

can be drawn at each step of the algorithm. Convergence is reached very quickly. Indeed, the configuration desired is almost obtained after only one iteration. It is necessary to wait for the 4th stage for an object to reach standard convergence ($\epsilon = 1e^{-6}$ in algorithm 1).

The solution computed is a set of points called \mathbf{P}' , with the same topology as before. The geometrical solution is illustrated by figure 11. The analysis of this solution gives the length parameters presented in table 1. It can be seen

that the 18th constraint that has been removed is equal to $L_{18} = 27.622761$ instead of 32 (the designer’s original goal). This solution is not the exact desired solution. It is probably just one solution close to the designer’s aims. The geometrical solver also gives the compatibility equations (in this example, only one equation describes the dependence between the 18 parameters) (there are only 17 independent parameters). The following section describes the generation of this compatibility equation.

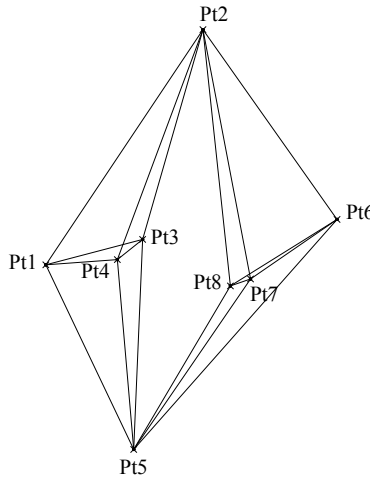


Figure 11: Example of the Double Banana configuration with $(\mathbf{P}', \mathbf{C}, \tilde{\mathbf{S}})$.

5.2. Compatibility equation generation

As we found that there was only one redundant constraint, there is therefore only one compatibility equation of the form (28) to be calculated. Thus we perform the calculation of γ_q^i presented in (29). Note that $q = 1$ in this case.

$$\sum_{i=1}^{18} \gamma_1^i b_i = 0 \quad (36)$$

Using equation (36), each element b_i can be calculated by a linear combination of the 17 others.

5.3. Handling the overconstrained problem

For example, let us assume that the value for L_{18} calculated by the solver is too far from the designer’s aims and that the value of L_9 may be less accurate than its primary specification. The other values are correct. Therefore

Length	Final measure	Initial value
L1	75.000000	94.382202
L2	69.000000	88.865066
L3	59.000000	82.048766
L4	48.000000	48.774994
L5	50.000000	50.418251
L6	51.000000	51.951901
L7	73.000000	78.689262
L8	57.000000	57.323643
L9	45.000000	45.86938
L10	68.000000	90.87904
L11	59.000000	83.600239
L12	74.000000	101.641527
L13	49.000000	59.380131
L14	35.000000	44.271887
L15	23.000000	24.779023
L16	34.000000	34.655447
L17	26.000000	26.248809
L18	27.622761	27.622761

Table 1: Lengths measured

the user wishes to know the new value of L_9 which conforms to the consistency of the problem. They can also change more than one value. For the compatibility equation, the coefficients of γ_1^i are the influence factors of each specification i which ensures the consistency between the 18 length specifications. As we want to specify L_{18} and let L_9 free, we apply equation (32) with $t_i = 0$ for $i \in \{1, \dots, 8, 10, \dots, 17\}$. Then,

$$\gamma_1^9 t_9 + \gamma_1^{18} t_{18} = 0 \quad (37)$$

$$t_9 = -\frac{\gamma_1^{18}}{\gamma_1^9} t_{18} \quad (38)$$

As $t_9 = L'_9 - L_9$ and L_9 is equal to $\sqrt{H_{99}}$, we have:

$$L'_9 = t_9 + \sqrt{H_{99}} \quad (39)$$

Parameters	Values
γ_1^1	0.008951
γ_1^2	-0.008535
γ_1^3	-0.013538
γ_1^4	0.006108
γ_1^5	-0.006594
γ_1^6	-0.012478
γ_1^7	-0.020714
γ_1^8	-0.008851
γ_1^9	0.039190
γ_1^{10}	0.022958
γ_1^{11}	-0.003612
γ_1^{12}	-0.008140
γ_1^{13}	-0.006937
γ_1^{14}	0.027790
γ_1^{15}	0.010428
γ_1^{16}	0.008528
γ_1^{17}	-0.003392
γ_1^{18}	-0.006685

Table 2: coefficients of the compatibility equations

Length L'_9 can be easily calculated thanks to relation (40).

$$L'_9 = -\frac{\gamma_1^{18}}{\gamma_1^9} t_{18} + \sqrt{H_{99}} \quad (40)$$

With $L_9 = 45.000000$, $t_{18} = 32 - 27.622761$, $\gamma_1^9 = 0.039190$ and $\gamma_1^{18} = -0.006685$.

After calculus, we find $L'_9 = 45.746657$.

6. Discussion

The geometric problem is described by a coordinate-free formulation. To solve the GCSP, the authors obtained the transformation between an initial configuration and a final one that conformed to the users' specifications. This transformation was totally defined when the \mathbf{X} matrix was calculated. The present size of matrix \mathbf{X} is $m \times n$, consequently the number of unknowns is

nm. As $3n - 6$ unknowns are sufficient to position n points in 3D, the algebraic problem is underconstrained, so values of some unknowns can be fixed to zero. This strategy was tested on several examples but was not presented in this paper.

The authors presented a method to analyze the GCSP. The analysis was performed before the solving process, something not done at present at each step of the resolution. An improvement is possible: to detect particular properties of geometric elements during the solving process. It will be interesting to analyse the algebraic system in it in order to know whether the final sketch is generic. If the final sketch is not generic, the proposed method cannot be applied.

When problems are over-constrained, a search is made to identify the redundant constraints by using the Gaussian elimination method. The independent specifications set depend on the order of the rows in the matrix \mathbf{A} . Different strategies could be studied to sort the rows in \mathbf{A} differently.

Finally, the double-banana problem is an academic example with only one overconstraint. It was chosen for pedagogical reasons. Further applications must be performed to demonstrate the efficiency of this on industrial problems with several overconstraints.

7. Conclusion

In this paper we used a non-cartesian formulation for GCSP. We wrote constraint equations for point-point distance and line-line angles. The efficiency of the witness method for handling problems defined by a coordinate free formulation was demonstrated and a solution for identifying redundancy between constraints in over-constrained cases was presented. A set of compatibility equations was generated after performing a prior analysis. Finally, strategies and tools were proposed to assist a team of designers to find a set of consistent specifications. The complete process was illustrated using the double banana geometrical configuration.

In the design environment it is increasingly difficult to specify a geometric problem correctly in a declarative way since designers often work together. The proposed tools could be used in a collaborative environment, where it is quite difficult to arbitrate between the different goals of design engineers.

References

- [1] B. Bettig, C. M. Hoffmann, Geometric Constraint Solving in Parametric Computer-Aided Design, *Journal of computing and information science in engineering* 11 (2011).
- [2] K. Haller, A. Lee-St.John, M. Sitharam, I. Streinu, N. White, Body-and-cad geometric constraint systems, *Computational Geometry* 45 (2012) 385 – 405. *Geometric Constraints and Reasoning*.
- [3] A. Clément, A. Rivière, P. Serré, A Declarative Approach for Geometry and Topology : a New Paradigm for CAD/CAM Systems, in: ed. Kluwer Academic (Ed.), *International Conference on Integrated Design and Manufacturing in Mechanical Engineering*, pp. 199–206.
- [4] P. Serré, Cohérence de la spécification d’un objet de l’espace euclidien à n dimensions, Ph.D. thesis, École Centrale Paris, 2000.
- [5] M. Moinet, Descriptions non cartésiennes et résolution de problèmes géométriques sous contraintes, Ph.D. thesis, École Centrale Paris, 2008.
- [6] G. Mandil, Modèle de représentation géométrique intégrant les états physiques du produit, Ph.D. thesis, École Centrale Paris & Université de Sherbrooke, 2011.
- [7] J. McTigue, R. Quinlan, Partial matrices whose completions all have the same rank, *Linear algebra and its applications* 438 (2013) 348–360.
- [8] M. Laurent, Matrix completion problems, in: C. Floudas, P. Pardalos (Eds.), *Encyclopedia of Optimization*, Springer US, 2001, pp. 1311–1400.
- [9] M. Moinet, P. Serré, A. Rivière, A. Clément, A new approach to transform a constrained geometric object, in: *Proceedings of the 20th CIRP Design Conference*, Nantes, France.
- [10] P. Serré, A. Ortuzar, A. Rivière, Non-cartesian modelling for analysis of the consistency of a geometric specification for conceptual design, *International Journal of Computational Geometry & Applications* 16 (2006) 549–565.

- [11] M. Moinet, P. Serré, Declarative modelling and transformation of a constrained geometric object, *Int. J. of Product Development*, 13 (2011) 344 – 361.
- [12] D. Michelucci, S. Foufou, Geometric constraint solving: The witness configuration method, *Computer-Aided Design* 38 (2006) 284 – 299.
- [13] D. Michelucci, S. Foufou, L. Lamarque, P. Schreck, Geometric constraints solving: some tracks, in: *SPM '06: Proceedings of the 2006 ACM symposium on Solid and physical modeling*, ACM, New York, NY, USA, Cardiff, Wales, United Kingdom, 2006, pp. 185–196.
- [14] S. Foufou, D. Michelucci, Interrogating witnesses for geometric constraint solving, *Information and Computation* 216 (2012) 24 – 38. Special Issue: 8th Conference on Real Numbers and Computers.
- [15] S. E. Thierry, P. Schreck, D. Michelucci, C. Fünfzig, J.-D. Génevaux, Extensions of the witness method to characterize under-, over- and well-constrained geometric constraint systems, *Computer-Aided Design* 43 (2011) 1234 – 1249. *Solid and Physical Modeling 2010*.
- [16] J. Graver., B. Servatius, H. Servatius, *Combinatorial Rigidity*. Graduate Studies in Mathematics, American Mathematical Society, 1993.
- [17] S. Balac, F. Sturm, *Algèbre et analyse: Cours de mathématiques de première année avec exercices corrigés*, Collection des sciences appliquées de l'INSA de Lyon, Presses Polytechniques et Universitaires Romandes, 2003.
- [18] R. Riedel, Condition nécessaire et suffisante de compatibilité d'un système linéaire de n équations à p inconnues - Lien avec le Théorème de Rouché-Fontené - Démonstration nouvelle du Théorème de Rouché-Fontené, Technical Report, 2011.
- [19] S. Caro, F. Bennis, P. Wenger, Tolerance synthesis of mechanisms: a robust design approach, *Journal of Mechanical Design* 127 (2005) 86–94.