A New Structural Rigidity for Geometric Constraint Systems

Christophe Jermann[1], Bertrand Neveu[2], and Gilles Trombettoni[2]

[1] Christophe.Jermann@epfl.ch
AI Lab, EPFL, 1015 Lausanne, Switzerland
[2] {Bertrand.Neveu, Gilles.Trombettoni}@sophia.inria.fr
COPRIN Team, INRIA-I3S-CERMICS,
2004 route des lucioles, BP 93, 06902 Sophia.Antipolis, France

Abstract. The structural rigidity property, a generalisation of Laman's theorem which characterises generically rigid bar frameworks in 2D, is generally considered a good heuristic to detect rigidities in geometric constraint satisfaction problems (GCSPs). In fact, the gap between rigidity and structural rigidity is significant and essentially resides in the fact that structural rigidity does not take geometric properties into account. In this article, we propose a thorough analysis of this gap. This results in a new characterisation of rigidity, the *extended structural rigidity*, based on a new geometric concept: the *degree of rigidity* (DOR). We present an algorithm for computing the DOR of a GCSP, and we prove some properties linked to this geometric concept. We also show that the extended structural rigidity is strictly superior to the structural rigidity and can thus be used advantageously in the algorithms designed to tackle the major issues related to rigidity.

1 Introduction

Geometric constraint satisfaction problems arise naturally in several areas, such as architecture, design of mechanisms and molecular biology. The rigidity concept plays an important for in many of these areas, for instance when one needs to decide whether a *geometric constraint satisfaction problem* (GCSP) is rigid or not, to detect over-rigid subparts which represent explanations for the absence of solutions, or to decompose a GCSP into several subGCSPs for the purpose of efficient solving.

Several solving methods [Kra92,BFH$^+$95,LM96,DMS97,HLS00,JTNR00] for GCSPs have to handle rigidity related problems. In particular, recursive rigidification techniques decompose a GCSP into a sequence of rigid subGCSPs to be solved separately and then assembled.

The techniques used so far for rigidity detection can be classified in two categories: pattern-based approaches [FH93,BFH$^+$95,Kra92] depend on a repertoire of rigid bodies of known *shape* which cannot cover all practical instances. Flow-based approaches [HLS97,LM96] use flow (or maximum matching) machinery to identify subGCSPs verifying a structural property: the *structural rigidity*. This property is based on a degree of freedom count.

The latter approaches are more general eventhough structural rigidity is only an approximation of rigidity. Heuristics have been proposed to enhance structural rigidity capabilities, none of which succeeded to fully cover the gap between structural rigidity and rigidity.

In this paper, we present a thorough analysis of this gap (Section 2). This analysis takes into account the different definitions and characterisations of rigidity that have been proposed in different communities: theory of mechanisms, structural topology and CAD. It allows us to identify precisely the causes of failure of the structural rigidity.

In section 3, we define a new geometric concept: the *degree of rigidity* (DOR) of a GCSP. We establish some interesting properties and propose an algorithm to compute the DOR of a GCSP. This new concept allows us to define a new characterisation of rigidity: the *extended structural rigidity* (Section 4).

We show that this new characterisation is strictly superior to the structural rigidity, even the heuristically enhanced one, but remains an approximation of rigidity. We show that this new characterisation, while remaining an approximation of rigidity, is strictly superior to the structural rigidity, even the heuristically enhanced one.

Finally, we briefly explain in section 5 how this new characterisation can be introduced in the algorithms designed by Hoffmann *et al.* [HLS97] to tackle with more reliability the major issues related to rigidity: deciding whether a GCSP is rigid or not, and identifying rigid and over-rigid sub-GCSPs.

2 Rigidity and Structural Rigidity

This section provides the necessary background definitions. We recall definitions of rigidity from the theory of mechanisms, the structural topology and the CAD communities. Then, we introduce the structural rigidity, a famous characterisation of rigidity based on an analysis of the degrees of freedom in a GCSP. We clarify the type of rigidity characterised by structural rigidity, and explain its limits.

2.1 Geometric Constraint Satisfaction Problems

Let us first define a GCSP (two examples are provided in figure 1).

Definition 1 Geometric Constraint Satisfaction Problems (GCSP)
A GCSP S is defined by a pair (O, C) (we note $S = (O, C)$) where:

- *O is a set of geometric objects,*
- *C is a set of geometric constraints binding objects in O.*

$S' = (O', C')$ is a **subGCSP** *of $S = (O, C)$ (noted $S' \subset S$) iff $O' \subset O$ and $C' = \{c \in C | c$ binds only objects in $O'\}$ (i.e. S' is induced by O').*

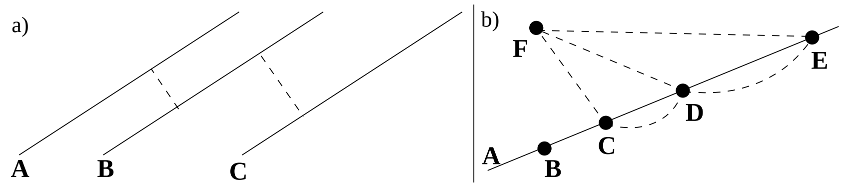

Fig. 1. a) A GCSP in 2D composed of 3 parallel lines lying at prescribed distance. **b)** A GCSP in 3D composed of one line (A) and 5 points (B, C, D, E and F); the constraints are 4 point-line incidences (B, C, D and E on A) and 5 point-point distances (CD, CF, DE, DF and EF).

Restrictions: We assume that geometric objects are indeformable (e.g., no circle with variable radius). Also geometric constraints must involve only positions and orientations of the objects and they must be independent from the global reference system (i.e., constraints only fix objects relatively one to another). These limitations make the structural characterisations of rigidity easier and are mandatory for geometric solving methods based on rigidity.

Solution of a GCSP: According to these restrictions, a solution to a GCSP $S = (O, C)$ is composed of one position and one orientation for each object in O that satisfy all the constraints in C. For the solving purpose, a GCSP is translated into a system of equations: each object is represented by a set of unknowns (over the reals) which determine its position and orientation; each constraint becomes one or several equations on the unknowns of the objects it constrains.

2.2 Rigidity

The concept of rigidity has been studied in several scientific fields: theory of mechanisms, structural topology and CAD are the three main scientific communities which have studied this concept.

Theory of mechanisms

Theory of mechanisms [Ang82] is interested in the study of movements in mechanisms. Mechanisms are a specific subclass of GCSPs, composed of mechanical pieces linked by mechanical articulations. The mechanical pieces can be of various types, while the articulations belong to a reduced set of well-defined types of mechanical joints.

The *degree of mobility* of a mechanism represents the number of independent movements it admits: if it is 0, the mechanism is isostatic (or rigid); if it is less than 0, the mechanism is hyperstatic (or over-rigid); otherwise, the mechanism is under-rigid.

The degree of mobility can be computed in two ways:

1. A specific analysis (static, geometric, dynamic or cinematic) of a given *configuration* of the mechanism. A configuration of a mechanism is composed of one position, one orientation and one set of dimensions for each mechanical piece in the mechanism.
2. The computation of the Gruebler formula [Grü17].

The first approach provides only a local information about the rigidity of the mechanism since the computed degree of mobility depends on the configuration of the mechanism to be analysed. The second approach is more general, but can be mistaken in case the mechanism contains redundant articulations or singularities.

Structural topology

Structural topology [Whi87,Gra02] is interested in the study of *bar frameworks*, a subclass of GCSPs composed of points linked by distance constraints.

The rigidity of a bar framework is related to the kind of movements it admits; we distinguish two kind of movements: *displacements*, which corresponds to rigid-body movements (translations and rotations), and *deformations* (or *flexes*), which do not preserve the relative positions of the points in the bar framework.

Structural topology also proposes two approaches to study the rigidity of a bar framework:

1. A specific study of one configuration in the bar framework; a configuration of a bar framework is composed of one position for each point in the bar framework and the corresponding set of distance values for its bars.
2. A generic study that defines the rigidity of all the generic configurations of the bar framework; a configuration of a bar framework is generic if there is no algebraic dependency between the coordinates of the points.

The specific study is based on algebraic computations, rendered possible by the fact that bar frameworks embed only points and distances, i.e. objects and constraints of well-known form and properties.

The generic rigidity is related to algebraic independence and based on a count of the degrees of freedom in the bar framework. Its main characterisation was proposed by Laman:

Theorem 1 [Lam70]
A bar framework in 2D is generically rigid if it has exactly 3 *DOFs and all its sub-systems (see Definition 1) have at least* 3 *DOFs.*

Unfortunately, no similar characterisation was found for 3D bar frameworks: all the trials to extend known 2D characterisations have failed. Moreover, generic rigidity corresponds to rigidity only for non-redundant and non-singular bar-frameworks.

Thus, structural topology proposes the same choice as the theory of mechanisms: either a specific study of a given configuration, or a general but sometimes erroneous evaluation of the rigidity of all the configurations of a GCSP.

Computer-Aided Design
CAD is a more recent community and is interested in almost every kind of GCSPs. To our knowledge, no general framework for the study of the rigidity of any GCSP was ever proposed. We propose here the extension of the framework defined in structural topology.

Rigidity is related to the kind of movements admitted by a GCSP. As already said, we distinguish two kinds of movements: the displacements and the deformations. Below is a formal definition of movements borrowed from the structural topology community:

Definition 2 Movements of a GCSP
A **movement** M *of a GCSP is a set of continuous and differentiable functions* $m_i : [0,1] \rightarrow \mathbb{R}$*, one function for each parameter of each object in the GCSP. An evaluation for a given* $t \in [0,1]$ *is a configuration* $M(t)$ *of the GCSP that must satisfy all its constraints.*

A movement M *is a* **displacement** *if for any* $t \in [0,1]$*, there exists a rotation* R *and a translation* T *such that* $M(t) = T(R(M(0)))$*. Otherwise,* M *is a* **deformation**.

Then, we propose a formal and simple definition for the rigidity of a solution of a GCSP:

Definition 3 Rigidity of the solution of a GCSP
A solution C *of a GCSP* S *is* **rigid** *iff every movement* M *such that* $M(0) = C$ *is a displacement.*

Remark: We talk about solution of a GCSP instead of configuration. Indeed, a configuration, in the sense given in the theory of mechanisms and structural topology, generally comprises values for the parameters of the constraints. In the GCSPs we consider, the parameters of the constraints are fixed and only the parameters of the objects are unknown. A set of values for these parameters was called a solution of a GCSP in the beginning of section 2.

Like in theory of mechanisms and structural topology, this definition allows for a specific study of a given solution of a GCSP. However, another form of rigidity is of greater interest for the CAD community: the *a priori* rigidity of all the solutions of a GCSP. Indeed, this information is used by a lot of geometric decomposition methods which aim at aggregating rigid subparts of a GCSP to produce the solutions of the complete GCSP [Kra92,HLS00,JTNR00].

In the rest of this paper, we will use the following terminology:

- A GCSP is **over-rigid** if all its solutions are over-rigid;
- it is **rigid** if all its solutions are rigid;
- it is **under-rigid** if all its solutions are over-rigid.

In the example of figure 1-b, the subGCSP CDF is rigid since a triangle is indeformable and can be displaced anywhere in the Euclidean 3D space. The subGCSP AF is under-rigid: the point F moves independently from the line A

since they are not constrained. The subGCSP $ACDEF$ is over-rigid since, in the generic case, it has no solution; a possible explanation is that three spheres with prescribed radii and centres (C, D and E) aligned onto a single line (A) generically (i.e. if the radii are algebraically independent) do not have a common intersection (F).

This definition of rigidity is similar to the generic rigidity in structural topology, or the Gruebler formula in the theory of mechanisms. This is the type of rigidity at which are aimed all the structural characterisations used in CAD. Whenever there is a risk of confusion, we will explicitly call *global rigidity* the rigidity of all the solutions of a GCSP.

2.3 Limits of the Global Rigidity

The global rigidity does not allow to classify every GCSP. Indeed, there exists GCSPs which have solutions with different rigidity. We say that these GCSPs are *non-globally characterisable.*

Consider the GCSP in Figure 2. It is composed of one line E and 4 points A, B, C and D. The constraints between these objects are: Incidence(A, E), Incidence(B, E), Incidence(C, E), Distance(A, B), Distance(A, D), Equidistance(A, B, B, C), Equidistance(A, D, D, C).

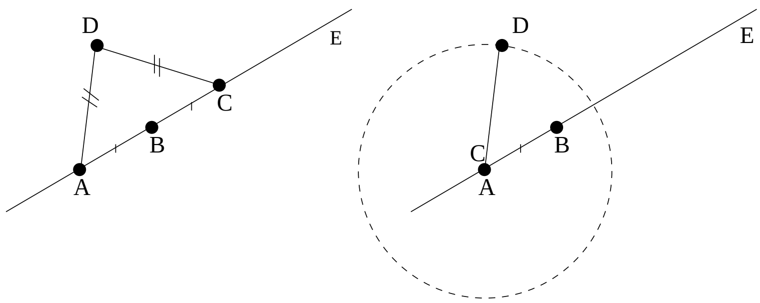

Fig. 2. Example of a GCSP with rigid and non-rigid solutions

According to the constraints of the GCSP, points A and C can either be coincident (Figure 2-a), or non-coincident (Figure 2-b). In case points A and C are coincident, point D can rotate independently of the other objects, introducing an under-rigidity in some solutions of the GCSP. In case A and C are non-coincident, point D is determined uniquely, i.e. some solutions of the GCSP are rigid.

This example illustrates the limits of any *a priori* characterisation of rigidity: they cannot characterise non-globally characterisable GCSPs.

To avoid this case, a common assumption is that GCSPs must be generic, i.e. contain no singular placements of their objects. This assumption is not reasonable: constraints like incidence and parallelism would be forbidden (because points are not generically incident to lines, lines are not generically parallel, ...), but these constraints are mandatory for designing many GCPSs in CAD.

We propose less restrictive restrictions:

Hypothesis 1 *Every GCSP is globally characterisable, i.e. all its solutions have the same rigidity.*

Hypothesis 2 *The valued constraints (distance, angle, ratio of distances, ...) are generic, i.e. only non-valued constraints (incidence, parallelism, equidistance, ...) introduce singularities in GCSPs.*

The first hypothesis is quite restrictive: it eliminates non-globally characterisable GCSPs like the one in Figure 2. On the contrary, the second assumption is not restrictive at all: singular valued constraints can always be formulated as non-valued constraints. For instance, a distance equal to zero can be replaced by an incidence constraint, an angle equal to zero by a parallelism one, *etc.*. Hence, this assumption only requires that given geometric properties, like incidences and parallelism, are stated using a specific constraint and not a singular valued constraint.

We assume that these hypotheses hold in the following sections and discuss the impact of removing them in section 6.

2.4 Characterisation of Global Rigidity

The characterisation principle is based on the following intuition:

- a GCSP which has less movements than displacements is over-rigid; moreover, a GCSP with an over-rigid subGCSP cannot be displaced also, and is then over-rigid itself;
- a non-over-rigid GCSP which has as many movements as displacements do not admit any deformation, and is then rigid;
- a non-over-rigid GCSP which has more movements than displacements admit deformations and is then under-rigid.

We note $M(S)$ (resp. $D(S)$) the number of movements (resp. displacements) of a GCSP S, i.e. the number of movements (resp. displacements) admitted by every solution of the GCSP. The principle of global rigidity characterisation can be formulated as follows:

- S is over-rigid iff $\exists S' \subseteq S$ such that $M(S') < D(S')$;
- S is rigid iff $M(S) = D(S)$ and S is not over-rigid;
- S is under-rigid iff $M(S) > D(S)$ and S is not over-rigid.

In the following section, we present the most famous *a priori* characterisation of rigidity: the *structural rigidity*.

2.5 Structural Rigidity

The structural rigidity is probably the most well-known and used characterisation of rigidity. It consists in computing approximations of the numbers of movements and displacements.

Approximation of the number of movements
To approximate the number of movements admitted by a GCSP, structural rigidity relies on a count of the degrees of freedom (DOF) in the GCSP. Intuitively, one DOF represents one independent movement in a GCSP. More formally:

Definition 4 Degree of freedom (DOF)
- Object o: DOF(o) is the number of independent parameters that must be set to determine the position and orientation of o.
- Constraint c: DOF(c) is the number of parameters the constraint allows to determine[3]*.*
- GCSP S: $DOF(S) = \sum_{o \in O} DOF(o) - \sum_{c \in C} DOF(c)$.

For instance, 2D lines have 2 DOFs, 3D points have 3 DOFs and 3D lines have 4 DOFs. A 2D line-line parallelism or distance removes 1 DOF, a 3D point-line incidence removes 2 DOFs and a point-point distance removes 1 DOF in any dimension. Hence, subGCSPs CDF, AF and $ACDEF$ in Figure 1-b have respectively 6, 7 and 5 DOFs.

Approximation of the number of displacements
Structural rigidity proposes to approximate the number of movements admitted by a GCSP by the number of independent displacements allowed in a geometric d-space:

Property 1 *A geometric space in dimension d allows exactly $\frac{d(d+1)}{2}$ independent displacements: d independent translations and $\frac{d(d-1)}{2}$ independent rotations.*

Definition of structural rigidity (s_rigidity)
Following the standard characterisation scheme for rigidity proposed in section 2.4, structural rigidity can be defined as follows:

Definition 5 Structural Rigidity (s_rigidity)
- A GCSP $S = (O, C)$ is **over-s_rigid** *iff $\exists S' \subseteq S$, $DOF(S') < \frac{d(d+1)}{2}$;*
- it is **s_rigid** *iff $DOF(S) = \frac{d(d+1)}{2}$ and S is not over-s_rigid;*
- it is **under-s_rigid** *iff $DOF(S) > \frac{d(d+1)}{2}$ and S is not over-s_rigid.*

[3] In practice, we count the number of independent equations in the subsystem of equations representing the constraint. This approximation, which comes from the traditional assumption that one equation fixes one unknown over the reals, is not always correct: $x^2 + y^2 = 0$ is a single equation fixing two unknowns at a time. However, under Hypothesis 2, this singular case cannot occur.

Obviously, structural rigidity is equivalent to Laman's theorem (Theorem 1) for 2D bar frameworks. In the general case, s_rigidity is considered a *good approximation* of rigidity [Hen92,LM96,HLS97]. We show in the following section that it is not a so reliable characterisation.

Causes of failure

The count of degrees of freedom is a first possible cause of failure of the characterisation by s_rigidity. Indeed, as for Laman's theorem or Gruebler's formula, this count can be mistaken by the presence of redundancies or singularities[4].

For example, the subGCSP $ACDEF$ in Figure 1-b is generically over-rigid as we explained previously. However, assume that the distance between E and F is set in such a way that this GCSP has a solution (for instance the one displayed in the picture). Then, this subGCSP would be rigid, but would contain at the same time a singularity (one distance value is set such that the GCSP has a solution, i.e. this distance depends on the other objects/constraints of the GCSP) and a redundancy (one constraint can be removed, e.g. Distance(E, F)). And in this case, this subGCSP would have only 5 $(= 4+3+3+3+3-2-2-2-1-1-1-1-1)$ DOFs instead of the required $\frac{d(d+1)}{2} = 6$ in dimension $d = 3$. Thus, it would be detected over-s_rigid because of the redundancy and singularity it contains.

Detecting redundancies and singularities is a difficult problem: it is generally equivalent to deciding whether or not there are algebraic dependencies in an equation system, and how many unknowns can be fixed by each subset of equations.

The second source of error comes from the fact that s_rigidity considers that every subGCSP should admit all the displacements allowed in a geometric space. This assumption is valid only if the subGCSP *fills* every dimension of the geometric space.

For instance, the segment CD in Figure 1-b is a rigid subGCSP in 3D: the points cannot be moved separately, but can be rotated and translated altogether. However, this subGCSP has only 5 $(= 3 + 3 - 1)$ DOFs, which is less than the prescribed $\frac{d(d+1)}{2} = 6$ in dimension $d = 3$. Hence, it is detected over-s_rigid. This is because the rotation around the axis defined by the segment has no effect on it, which removes one independent rotation from its possible displacements.

This counter-example illustrates a very strong limit of structural rigidity since, according to Definition 5, this implies that any GCSP containing a distance constraint between two points in 3D will be considered over-s_rigid.

Heuristically enhanced structural rigidity

To try to overcome these problems, a common heuristic [Sit00] recommends to consider only *non-trivial* subGCSPs, i.e. subGCSPs composed of at least $d + 1$ objects in dimension d. This heuristic, while interesting for GCSPs composed of points only, does not fix the problem[5].

---

[4] Under Hypothesis 2, only redundancies can mistake the DOF count.

[5] In fact, this heuristic even causes new problems. For instance, the subGCSP $ABCD$ in Figure 1-b is non-trivial and is clearly under-rigid: point B can move independently

For instance, the subGCSP $ACDE$ in Figure 1-b. is non-trivial (it is composed of 4 objects in 3D) and is rigid (the three points can only be translated altogether along the line, and the line can be displaced anywhere). However, this subGCSP has only 5 $(= 4+3+3+3-2-2-2-1-1)$ DOFs, and is then over-s_rigid. A similar wrong behaviour occurs with the GCSP ABC of Figure 1-a: it is also non-trivial (3 objects in 2D) and rigid (3 parallel lines lying at prescribed distance in the plane), but has only 2 $(= 2+2+2-1-1-1-1)$ DOFs and is then over-s_rigid. This is because the number of displacements does not depend on the number of objects but on their geometric properties.

Another way to treat all these special cases is to add *ad-hoc* rules to characterise them correctly [Sit00]. However, these rules cannot handle every exception that can occur in practice, and renders the characterisation process not very general.

Conclusion on the structural rigidity
The structural rigidity suffers from the lack of reliability of the two approximations it uses and none of the heuristics proposed so far permit to overcome these problems.

In the following section, we introduce the *degree of rigidity* concept which represents an exact evaluation of the number of displacements allowed for a given GCSP, and requires to take into account the geometric properties of the GCSP. It is the basis for our new characterisation of rigidity presented in Section 4.

3 Degree of Rigidity

Intuitively, the degree of rigidity represents the number of DOFs a GCSP must have to be rigid. According to the standard characterisation scheme proposed in the previous section, this means that the degree of rigidity is an evaluation of the number of displacements allowed in a GCSP.

To define the degree of rigidity formally, we need to introduce the concept of *rigidification* of a set of objects:

Definition 6 Rigidification
A rigidification of a set of objects O is a non-redundant set of constraints R_O such that the resulting GCSP $S_R = (O, R_O)$ is rigid.

We also need to define the concept of geometric consistency (*g_ consistency*). Intuitively, two set of constraints are g_consistent if they do not imply contradictory *geometric properties*. By geometric property is understood any geometric relation between objects which can have an impact on its number of displacements. More formally:

of points C and D along line A, which introduces a deformation in the subGCSP. However, this subGCSP has 6 DOFs, which makes it s_rigid if we do not check the trivial subGCSPs it contains.

Definition 7 G_consistency
Let $C1$ and $C2$ be two set of constraints on the same set of objects O. $C1$ and $C2$ are g_ consistent iff $I_g(C1 \cup C2)$ is consistent, where I_g is an operator which produces all the geometric properties induced by a set of constraints.

It is now possible to propose a formal definition of the degree of rigidity:

Definition 8 Degree of Rigidity (DOR)
Let O' be a subset of objects in a GCSP $S = (O, C)$ and $R_{O'}$ be a rigidification of O' g_ consistent with C. Then, DOR($O', R_{O'}, S$) = DOF(($O', R_{O'}$)).

Thanks to Hypothesis 1, the DOR is unique and does not depend on the chosen rigidification as long as it is g_consistent with the constraints of the GCSP. Indeed, if different rigidifications were yielding to different DOR values, this would mean that the subGCSP S' induced by O' would have a varying number of displacements, i.e. would be non-globally characterisable which does not satisfy Hypothesis 1.

To define the degree of rigidity of a subset O' of objects in a GCSP S, we need to take into account all the objects and constraints in S because geometric properties can come from objects/constraints outside O'. For example, consider the subset of objects $O' = \{A, C\}$ in the GCSP of Figure 1-a; the parallelism of these lines has an impact on their number of displacements in 2D, and is thus a geometric property of interest for us. However, the parallelism is not directly stated as a constraint between A and C, but comes from the fact that both are parallel to B; this is why we need to consider the complete GCSP to be able to infer all the geometric properties on a subset of objects only.

Why the DOR is an exact evaluation of the number of displacements
The degree of rigidity is exactly equal to the number of displacements of S_R. Indeed, the DOF count is not mistaken because the rigidifications contain no redundancies nor singularities[6]. More important, if the DOR did not reflect the right number of displacements of the subGCSP S' induced by O', it would mean that $R_{O'}$ introduces a contradictory geometric property in the system, i.e. is not g_consistent with the constraints of the GCSP embedding O'. We refer the reader to [Jer02] for a more formal proof of these claims.

3.1 Computation of the DOR

The process for computing the DOR of a subset of objects O' in a GCSP $S = (O, C)$ could be described as follows:

1. Compute the set R of all the possible rigidifications of O'
2. Extract a rigidification $R_{O'} \in R$ which is g_consistent with C
3. DOR($O', R_{O'}, S$) = DOF(($O, R_{O'}$)).

[6] Other than the ones induced by the non-valued constraints in the rigidifications, see Hypothesis 2.

Computing all the rigidifications of a set of objects is not easy. Fortunately, the DOR computation process can be simplified thanks to the following proposition.

Proposition 1 *If O' and O'' are two subsets of objects in a GCSP S such that $O'' \subset O'$, then DOR($O'', R_{O''}, S$)$\leq$DOR($O', R_{O'}, S$).*

Basically, this means that the DOR increases with the number of objects. Indeed, keeping in mind that the DOR represents the number of displacements allowed in a subGCSP, and that we restricted the constraints to be only relative to the objects, it is clear that adding objects to a subGCSP cannot remove displacements. □

Then, we define the concept of *DOR-minimal subset of objects*:

Definition 9 DOR-minimal subset of objects
A subset O' of objects in a GCSP S is **DOR-minimal** *if it contains no proper subset of objects with the same DOR.*

It results from Proposition 1 that:

Proposition 2 *DOR($O', R_{O'}, S$) $= max_{O'' \in DM(O',S)}$DOR($O'', R_{O''}, S$), where DM(O', S) is the set of DOR-minimal subsets of objects in O'.*

Indeed, since the DOR only increases when objects are added, the DOR of any subset of objects is necessarily equal to the greatest of its DOR-minimal ones. □

Thus, computing the DOR of any set of objects in a GCSP amounts to computing the DOR of its DOR-minimal subsets. This is an important improvement because of the following proposition:

Proposition 3 *In dimension d, a DOR-minimal subset of objects contains at most $\frac{d(d-1)}{2} + 1$ geometric objects.*

Indeed, a single geometric object in dimension d allows at least d displacements. Moreover, in a DOR-minimal GCSP, each object must add at least 1 to the DOR, otherwise the object is unnecessary, and the subGCSP is not DOR-minimal. All these remarks imply that a subGCSP with n objects has a DOR greater or equal to $d+n-1$ (d for the first object, $n-1$ for the remaining ones). Finally, the DOR of a subGCSP is at most $\frac{d(d+1)}{2}$ since a geometric d-space allows at most this number of independent displacements (see Proposition 1). Thus, for a subGCSP with n objects, $d+n-1 \leq \frac{d(d+1)}{2}$, i.e. $n \leq \frac{d(d-1)}{2} + 1$.□

In fact, we have even proved by enumeration that for GCSPs in 2D (resp. 3D) composed only of points, lines (and planes in 3D) linked by distance, angle, parallelism and incidence constraints, this maximum number of objects is reduced to 2 (resp. 3). For GCSPs of these types, which can be used to model many real-world applications, this means that one only has to consider pairs (resp. triplets in 3D) of objects to compute the DOR of any set of objects.

DOR-minimal and rigidification computation
In a given class $\mathbb{S}$ of GCSPs with a fixed set $\mathbb{O}$ of types of objects and a fixed set $\mathbb{C}$ of types of constraints, we propose to use the following process for determining the set of DOR-minimal subsets (and their possible rigidifications) that can appear in any GCSP in $\mathbb{S}$:

1. Compute the set $DORmin_n$ of all the subsets of n objects of types in $\mathbb{O}$
2. For each $O \in DORmin_n$, determine the set R_O of all the possible rigidifications of O based on constraints of types in $\mathbb{C}$
3. For each pair $R \in R_O$:
   - if $\exists O' \subsetneq O$ with DOR($O', R', \mathbb{S}$)=DOR($O, R, \mathbb{S}$), then the pair (O, R) is not a potential DOR-minimal;
   - otherwise, (O, R) is added to the list of potential DOR-minimal.

This process terminates because of the size limit given in Proposition 3, and because the types of objects and constraints are fixed. With this process, one can precompute once for all a table of all the possible pairs (set of objects, rigidification) which can be DOR-minimal in any GCSP of the considered class of GCSPs We have done this precomputation for the classes of GCSPs discussed above, and it resulted in a set of 3 (11 in 3D) subsets of objects, associated to at most 3 rigidifications each.

Checking the G_consistency
To check the g_consistency, one need an inference operator able to produce all the geometric properties induced by a set of constraints which influence the number of displacements of a given set of objects in a GCSP. It is known (see [Jer02] for more details) that the number of displacements only depends on the singularity or non-singularity of the relative positions of the objects. For instance, two parallel lines in 2D allow only 2 displacements, while two non-parallel lines allow 3. Because of Hypothesis 2, every geometric property can be stated by a non-valued constraint (which represent a singular relative placement) or its negation (which represent a non-singular relative placement).

Thus, one need to be able to infer new constraints from a given set of constraints. This amounts to geometric theorem proving: I_g (see Definition 7) is nothing but an automated geometric deduction process. For this purpose, one could use formal approaches [Wu86,Cho88,Wan02] or rule-based inference methods. We will not detail this step in this paper.

Avoiding formal theorem proving The use of several heuristics can avoid the need for theorem proving in practice. For instance, the candidate pairs (DOR-minimal subset, rigidification) can be sorted in decreasing order of their DOR, since the DOR of a set of objects is the maximum of the DORs of its DOR-minimal subsets (see Proposition 2). Also, the pairs for which the rigidification can be proved g_consistent trivially can be used first; this is the case in particular if a rigidification specifies geometric properties which are explicitly stated as constraints in the GCSP.

3.2 Example of DOR Computation

We illustrate the computation of the DOR on the GCSP depicted in Figure 1-a. First, we compute the DOR-minimal subsets of objects in this GCSP. As stated above, they are reduced to pairs of objects since we are in the class ({points, lines}, {distances, angles, incidences, parallelisms}) in 2D: $\{A,B\}$, $\{B,C\}$ and $\{A,C\}$ are the three DOR-minimal subsets of objects in this GCSP. We start by computing the DOR of $\{A,B\}$.

The possible rigidifications of $\{A,B\}$ in the considered class of GCSPs are:

1. Angle(A,B)=α ($\alpha \neq 0$)
2. Parallelism(A,B), Distance(A,B)=d ($d \neq 0$)
3. Incidence(A,B) (which means coincidence of the lines).

The DOR associated to these rigidifications are respectively:

1. DOF(($\{A,B\}$,{Angle(A,B)}))=$2+2-1$=**3**
2. DOF(($\{A,B\}$,{Parallelism(A,B), Distance(A,B)}))=$2+2-1-1$=**2**
3. DOF(($\{A,B\}$,{Incidence(A,B)}))=$2+2-2$=**2**

Now we have to determine which rigidifications are g_consistent with the constraints of the GCSP. It is clear that it cannot be the first one, since the GCSP explicitly contains a parallelism constraint between A and B. Now, depending on the value of the distance (null or not null) in the GCSP ABC, it could be the second or the third one[7]. In fact, we do not even care about the value of the distance because the rigidifications 2 and 3 result in the same DOR: this illustrates how we can skip theorem proving in practical cases.

The same process is repeated for $\{B,C\}$ and $\{A,C\}$ and results in DOR=2 for both also. According to Proposition 2, the DOR of the GCSP is then 2. This explains why, while rigid, this GCSP does not have the 3 DOFs required by the structural rigidity in 2D.

4 Extended Structural Rigidity

Given the concept of degree of rigidity, we propose the following characterisation of the rigidity of all the solutions of a GCSP:

Definition 10 Extended Structural Rigidity (es_rigidity)
- A GCSP $S=(O,C)$ is **over-es_rigid** *iff $\exists S'=(O',C') \subseteq S$, DOF($S'$) < DOR($O', R_{O'}, S$);*
- it is **es_rigid** *iff DOF(S) = DOR($O, R_{O'}, S$) and S is not over-es_rigid;*
- it is **under-es_rigid** *iff DOF(S) > DOR($O, R_{O'}, S$) and S is not over-es_rigid.*

[7] Note that according to the Hypothesis 2, we can ensure that the distance cannot be null.

This definition is very similar to the definition of structural rigidity (see Definition 5). In fact, the only difference is that we use the DOR instead of the *constant* $\frac{d(d+1)}{2}$.

Table 1 presents the answers given by the structural rigidity (S1), the heuristically enhanced structural rigidity (S2), i.e. where only non-trivial subGCSPs are considered, and the extended structural rigidity (S3) over some subGCSPs of Figure 1-b. Each line in this table presents one subGCSP.

	Rigidity	DOF	S1	S2	DOR	S3
CD	exact	5	over	trivial	5	exact
CDF	exact	6	over	trivial	6	exact
BCD	under	7	over	trivial	5	under
ACDE	exact	5	over	over	5	exact
ABCD	under	6	over	exact	5	under
ACDEF	over	5	over	over	6	over
ABCDE	under	6	over	over	5	under
ABCDEF	over	6	over	over	6	over

Table 1. Comparison table for rigidity and structural characterisations of rigidity on subGCSPs of the example presented in figure 1-b

This table illustrates the superiority of the es_rigidity over the s_rigidity, even with the commonly used heuristic; we also see that non-trivial subGCSPs having DOR < 6 ($\frac{d(d+1)}{2} = 6$ in dimension $d = 3$) can mislead the heuristically enhanced s_rigidity; the es_rigidity characterises correctly all subGCSPs in this example (as well as every subGCSP in Figure 1).

Here is a formal proof by disjunction that the es_rigidity is strictly superior (in the sense it characterises correctly strictly more GCSPs) to the s_rigidity, even heuristically enhanced:

- If the count of DOFs is false, then both es_rigidity and s_rigidity fail.
- If the count of DOFs is correct, then:
  - if the number of displacements (=DOR) is equal to $\frac{d(d+1)}{2}$ for every subGCSP then both s_rigidity and es_rigidity characterise correctly the GCSP.
  - if the number of displacements (=DOR) is not equal to $\frac{d(d+1)}{2}$ for at least one subGCSP, then s_rigidity (even with the discussed heuristic) fails while es_rigidity characterises correctly the GCSP.

□

Thus, every method based on previous definitions of structural rigidity should consider using our new characterisation of rigidity: it will certainly turn out into a significant gain in reliability and generality.

4.1 Limits of the Es_rigidity

Es_rigidity uses an exact evaluation of the number of displacements but still relies on a DOF count to approximate the number of movements in a GCSP.

We have explained in section 2.5 that the DOF count can be false in case of redundancies, and in case of singularities if Hypothesis 2 does not hold. Thus it remains an heuristic even if Hypothesis 1 holds.

5 Tackling the Main Issues Related to Rigidity with Es_rigidity

In this section, we briefly explain how s_rigidity can be replaced by es_rigidity in the algorithms designed by Hoffmann *et al.* [HLS97] to tackle the major issues related to the rigidity concept, i.e. deciding if a GCSP is rigid, identifying rigid and over-rigid subGCSPs and minimising their size.

The structural rigidity identification algorithms proposed by Hoffmann *et al.* use flow machinery in a bipartite network derived from the GCSP: the objects-constraints network (see Figure 3 of an example).

Definition 11 Objects-Constraints Network
To a GCSP $S = (O, C)$ *corresponds one objects-constraints network* $G = (V, E, w)$ *such that:*
- $s \in V$ *is the source and* $t \in V$ *is the sink*
- *Each object* $o \in O$ *becomes an object-node* $v_o \in V$
- *Each constraint* $c \in C$ *becomes a constraint-node* $v_c \in V$
- *For each object* $o \in O$ *there is an arc* $v_o \to t$ *in* E*, with capacity* $w(v_o \to t)$=*DOF(o)*
- *For each constraint* $c \in C$*, there is an arc* $s \to v_c$ *in* E*, with capacity* $w(s \to v_c)$=*DOF(c)*
- *For each object* $o \in O$ *constrained by* $c \in C$ *there is an arc* $v_c \to v_o$ *of capacity* $w(v_c \to v_o) = \infty$ *in* E.

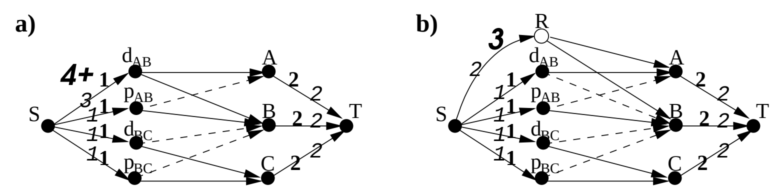

Fig. 3. The objects-constraint network corresponding to the GCPS in Figure 1-a. **a)** Application of Hoffmann *et al.*'s function `Distribute`. **b)** Application of our new function `Distribute`.

A flow distribution in this network represents a distribution of the constraints DOFs onto the objects DOFs. Hence, a maximum flow represents an optimal distribution of the DOFs in the GCSP.

Hoffmann *et al.*'s algorithms

Hoffmann *et al.*'s algorithms are all based on the same flow-distribution function called `Distribute` which:

1. overloads the capacity of one arc from the source to a constraint c, by $\frac{d(d+1)}{2}+1$.
2. distributes a maximum flow in the overloaded network.

This process is illustrated in Figure 3-a, where the first constraint is overloaded by 4 (in 2D, $\frac{d(d+1)}{2}+1=4$) and a maximum flow distribution is depicted.

If the overloaded arc is not saturated (i.e. the distributed flow is less than the capacity of the arc), Hoffmann *et al.* proved that there exists a subGCSP S' having less than $\frac{d(d+1)}{2}+1$ DOFs, i.e. a well- or over-s_rigid subGCSP. This subGCSP is induced by the set of objects traversed during the last search for an augmenting path in the maximum flow computation [HLS97].

Then, Hoffmann *et al.* propose an algorithm for finding well- or over-s_rigid subGCSPs in a GCSP, which simply applies the `Distribute` function to each constraint in an incrementally constructed objects-constraints network [HLS97]. Other algorithms for identifying only over-s_rigid subGCSPs and for minimising well- or over-s_rigid subGCSPs are proposed in [HLS97].

Modifications of Hoffmann *et al.*'s algorithms

We propose to modify the `Distribute` function to derive a new family of similar algorithms based on es_rigidity to tackle the major issues related to rigidity in a more reliable way.

To achieve this improvement, we propose two modifications:

1. the overload is not applied on an existing constraint but is distributed through a dedicated node R added to the objects-constraints network; this node, which can be seen as a virtual constraint, can be linked to any subset of objects in the network.
2. the overload $\frac{d(d+1)}{2}+1$ is replaced by DOR(O', S)+1 where O' is the subset of objects to which the virtual constraint R is attached.

Hence, if a maximum flow distribution cannot saturate the arc $s \rightarrow R$, this means that there exists a subGCSP with less DOFs than the DOR of its objects, i.e. a well- or over-es_rigid GCSP.

This new process is illustrated in Figure 3-b, where the virtual constraint R appears and is associated to a capacity of 3 since it is linked to A and B, two parallel lines having DOR=2. Note that both the value of the overload, and the way it is distributed have changed on this example.

Thanks to the properties of the DOR concept we have proven in section 3, it is not needed to apply the new `Distribute` function to any subset of objects but only to the DOR-minimal subsets. Thus, we can derive new polynomial algorithms similar to Hoffmann *et al.*'s ones for detecting well- or over-es_rigid subGCSP, for detecting over-es_rigid ones only, and for minimising them.

More details, as well as correctness and completeness proofs and a discussion on the complexity of these algorithms, can be found in [JNT03].

6 The Multiple Rigidity Case

In case Hypothesis 1 does not hold, we have to take into account the fact that GCSPs can be non-globally characterisable.

In this case, the DOR computation process we have presented is still valid: the only change is that the rigidifications which are g_consistent with the constraints of the GCSP can yield to a set of DORs instead of a single DOR value.

However, the geometric decomposition of non-globally characterisable GCSPs would result in as many decompositions as there are possible cases of rigidity for their subGCSPs, i.e. a potentially exponential number of decompositions. Thus, even if we cannot guarantee Hypothesis 1, it is advisable to continue using an *a priori* global rigidity of all the solutions of the GCSPs, knowing well that it is only a heuristic.

Even if we do not take into account the multiple DORs in non-globally characterisable GCSPs, the es_rigidity remains better than the s_rigidity.

7 Conclusion

We have provided a thorough analysis of the gap between structural rigidity and rigidity: it lies in the approximation of both the number of movements and the number of displacements of the considered GCSP. The causes of failure mainly reside in the fact that the structural rigidity does not take into account the geometric properties of the GCSPs.

We have introduced the degree of rigidity concept for the exact computation of the number of displacements allowed in a given GCSP. We have explicited an algorithm for computing the DOR, which involves geometric theorem proving in the general case. We have shown that this can be avoided in practical cases.

We have then explained why any *a priori* characterisation of the rigidity of all the solutions of a GCSP cannot be complete and correct. This is related to the fact that some GCSPs have at the same time rigid and non-rigid solutions. This has an impact on the DOR computation, since this results in multiple DOR values for a given set of objects.

However, under the assumption that all the solutions of a GCSP have the same rigidity state, we have proposed a new *a priori* characterisation of rigidity: the extended structural rigidity. It uses the DOR as an evaluation of the number of displacements allowed for a GCSP. We have shown that this characterisation, while still being an approximation of rigidity, is strictly superior to the classical or heuristically enhanced structural rigidity.

We have shown (see [JNT03] for more details) that algorithms based on the structural rigidity can be modified to deal with the extended structural rigidity. This allows us to address the major issues related to rigidity in a more reliable and general way.

Hence, all geometric methods based on the structural rigidity characterisation can now be upgraded to use the extended structural rigidity. We expect this will result in much more reliable algorithms in practice, in particular for the decomposition of GCSPs.

References

[Ang82] J. Angeles. *Spatial Kinematic Chains.* Springer-Verlag, Berlin, 1982.

[BFH+95] W. Bouma, I. Fudos, C.M. Hoffmann, J. Cai, and R. Paige. Geometric constraint solver. *Computer Aided Design*, 27(6):487–501, 1995.

[Cho88] S.C. Chou. *Mechanical Theorem Proving.* Reidel Publishing Co., 1988.

[DMS97] J.-F. Dufourd, P. Mathis, and P. Schreck. Formal resolution of geometrical constraint systems by assembling. In C. Hoffmann and W. Bronswort, editors, *Proc. Fourth Symposium on Solid Modeling and Applications*, pages 271–284, 1997.

[FH93] I. Fudos and C.M. Hoffmann. Correctness proof of a geometric constraint solver. Technical Report TR-CSD-93-076, Purdue University, West Lafayette, Indiana, 1993.

[Grü17] Grübler. *Getriebelehre.* Springer, Berlin, 1917.

[Gra02] J. Graver. *Counting on Frameworks: Mathematics to Aid the Design of Rigid Structures.* Number 25 in Dolciani Mathematical Expositions. Mathematical Association of America, 2002.

[Hen92] B. Hendrickson. Conditions for unique realizations. *SIAM j Computing*, 21(1):65–84, 1992.

[HLS97] C.M. Hoffmann, A. Lomonosov, and M. Sitharam. Finding solvable subsets of constraint graphs. In *Principles and Practice of Constraint Programming CP'97*, pages 463–477, 1997.

[HLS00] C.M. Hoffmann, A. Lomonosov, and M. Sitharam. Decomposition plans for geometric constraint systems. In *Proc. J. Symbolic Computation 2000*, 2000.

[Jer02] C. Jermann. *Résolution de contraintes géométriques par rigidification récursive et propagation d'intervalles.* Thèse de doctorat en informatique, Université de Nice Sophia-Antipolis, 2002.

[JNT03] C. Jermann, B. Neveu, and G. Trombettoni. Algorithms for identifying rigid subsystems in geometric constraint systems. In *18^{th} International Joint Conference in Artificial Intelligence, IJCAI-03*, 2003.

[JTNR00] C. Jermann, G. Trombettoni, B. Neveu, and M. Rueher. A constraint programming approach for solving rigid geometric systems. In *Principles and Practice of Constraint Programming, CP 2000*, volume 1894 of *LNCS*, pages 233–248, 2000.

[Kra92] G. Kramer. *Solving Geometric Constraint Systems.* MIT Press, 1992.

[Lam70] G. Laman. On graphs and rigidity of plane skeletal structures. *J. Eng. Math.*, 4:331–340, 1970.

[LM96] R.S. Latham and A.E. Middleditch. Connectivity analysis: A tool for processing geometric constraints. *Computer Aided Design*, 28(11):917–928, 1996.

[Sit00] M. Sitharam. Personal communication on the minimal dense algorithm. University of Florida at Gainesville, 2000.

[Wan02] D. Wang. Geother 1.1: Handling and proving geometric theorems automatically. In *Proceedings of the $4^t h$ International Workshop on Automated Deduction in Geometry*, 2002.

[Whi87] W. Whiteley. Applications of the geometry of rigid structures. In Henry Crapo, editor, *Computer Aided Geometric Reasoning*, pages 219–254. INRIA, 1987.

[Wu86] W. Wu. Basic principles of mechanical theorem proving in elementary geometries. *J. Automated Reasoning*, 2:221–254, 1986.