Directed graphs for the analysis of rigidity and persistence in autonomous agent systems.

Julien M. Hendrickx[1], Brian D.O. Anderson[2], Jean-Charles Delvenne[1] and Vincent D. Blondel[1]

[1] *Department of Mathematical Engineering, Université catholique de Louvain, Avenue Georges Lemaitre 4, B-1348 Louvain-la-Neuve, Belgium;* `<last name>@inma.ucl.ac.be`
[2] *National ICT Australia Ltd. and the Australian National Universisty, 216 Northbourne avenue, Canberra ACT 2601 Australia;* `brian.anderson@nicta.com.au`

SUMMARY

We consider in this paper formations of autonomous agents moving in a 2-dimensional space. Each agent tries to maintain its distances toward a pre-specified group of other agents constant and the problem is to determine if one can guarantee that the distance between every pair of agents (even those not explicitly maintained) remains constant, resulting in the persistence of the formation shape. We provide here a theoretical framework for studying this problem. We describe the constraints on the distance between agents by a directed graph and define *persistent graphs*. A graph is persistent if the shapes of almost all corresponding agent formations persist. Although persistence is related to the classical notion of rigidity, these are two distinct notions. We derive various properties of persistent graphs, and give a combinatorial criterion to decide persistence. We also define minimal persistence (persistence with the least possible number of edges), and we apply our results to the interesting special case of cycle-free graphs.

KEY WORDS: Autonomous agents; Rigidity; Graph Theory

1. INTRODUCTION

From the recent increasing development of autonomous agent systems arise new questions in graph theory. Consider a formation of n agents able to move in a 2-dimensional space.

Contract/grant sponsor: The work of Brian D.O. Anderson is supported by an Australian Research Council Discovery Project Grant and by National ICT Australia, which is funded by the Australian Government's Department of Communications, Information Technology and the Arts and the Australian Research Council through the Backing Australia's Ability Initiative.

Contract/grant sponsor: The work of Julien M. Hendrickx, Vincent D. Blondel and Jean-Charles Delvenne is supported by the Belgian Programme on Interuniversity Attraction Poles initiated by the Belgian Federal Science Policy Office, and by the Concerted Research Action (ARC) "Large Graphs and Networks" of the French Community of Belgium. The scientific responsibility rests with its authors. Julien M. Hendrickx and Jean-Charles Delvenne are FNRS fellows (Belgian Fund for Scientific Research)

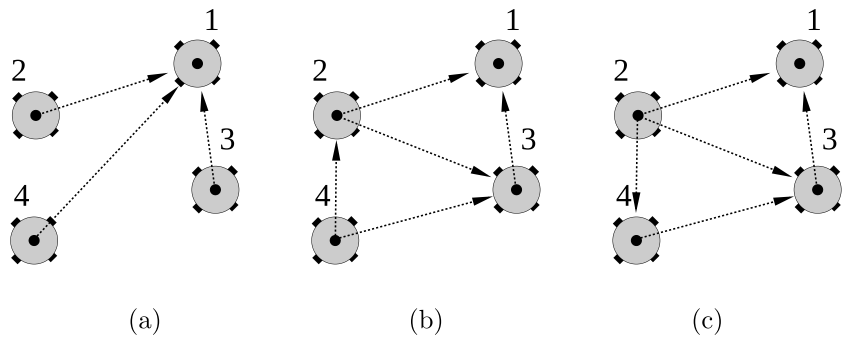


Figure 1. Examples of autonomous agent systems; each arrow represents a distance constraint. In (a) for example, agents 2, 3 and 4 try to maintain their respective distances toward agent 1 constant. We will show that only (b) is persistent.

With each agent is associated a set of neighbors, and with each neighbor a distance constraint which the agent must meet with respect to that neighbor. Thus if agent i has two neighbors j and k, agent i has to maintain distances d_{ij} from agent j and simultaneously d_{ik} from agent k. It is important to understand that this is a constraint for agent i but not for agent j of agent k, which will a priori not be required to do anything in order to maintain their respective distances from agent i constant. Moreover, as long as a particular agent satisfies all its distance constraints, no other hypothesis is made about its movement. Agent 4 in Figure 1(a) can thus move freely on a circle of radius d_{41} centered on agent 1. In relation to a particular formation, we are interested in knowing if one can guarantee that, provided that each agent is trying to satisfy all its distance constraints, the structure of the formation will be conserved. In other words, we want to know if the distance between every pair of agents (whether or not there is a distance constraint in either direction between the pair) will remain constant along any continuous move. As shown in Figure 1, this kind of systems can be represented by a directed graph: To each agent corresponds a vertex, and there is a directed edge from i to j if i has a constraint on the distance it must maintain from j. Note that double edges are allowed and represent a situation where both i and j have to maintain the distance between them constant.

This issue is evidently related to the notion of rigidity that has been used for decades in various domains like civil or mechanical engineering. The first works were done on particular concrete systems, but rigidity can actually be studied from a graph theory point of view. A framework is represented by a graph $G = (V, E)$, where V is the set of vertices representing the articulations, and E is the set of undirected edges representing the beams or any other type of links. Suppose now that we assign arbitrary positions in $\Re^2$ to all the vertices, and consider all the continuous moves such that the distance between the positions of any two vertices connected by an edge remains constant (This could be done in any other space, but in the sequel we will always work in $\Re^2$). The graph is called *rigid* if for almost all position assignments, every such move preserves the distance between the positions of any pair of vertices, as shown in the examples in Figures 2(a) and 2(b). In contrast, Figure 1(a) illustrates a non rigid graph.

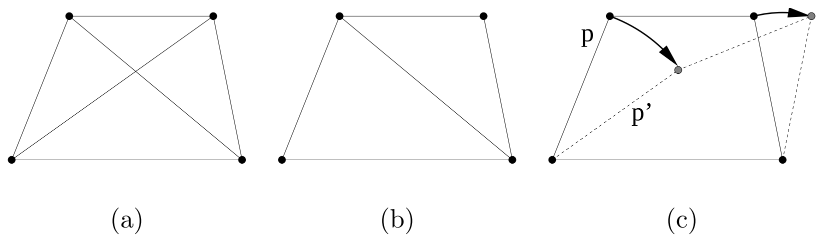


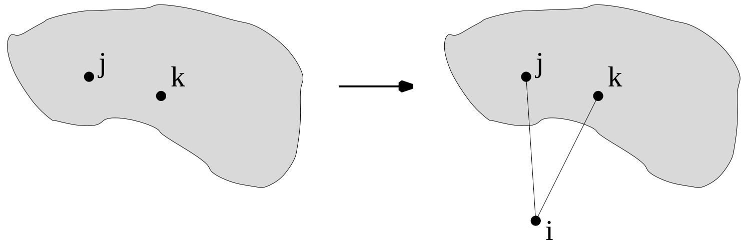
Figure 2. (a) is rigid, (b) is minimally rigid, and (c) is not rigid.

In 1970, Laman gave a necessary and sufficient condition (see Theorem 1) for a graph to be rigid in 2D [1]. In his works, he used the motion based approach as above: each edge represents a distance constraint between two vertices, and one wants to be sure that the structure cannot be deformed by a continuous move for which all the distance constraints are satisfied all the time. There is a dual equivalent approach, based on static equilibrium of forces [2]. The links are viewed as "force transmitters", and a structure is rigid if it can bear any equilibrium load, i.e., a collection of applied loads on each vertex such that the sum of all these applied loads is zero.

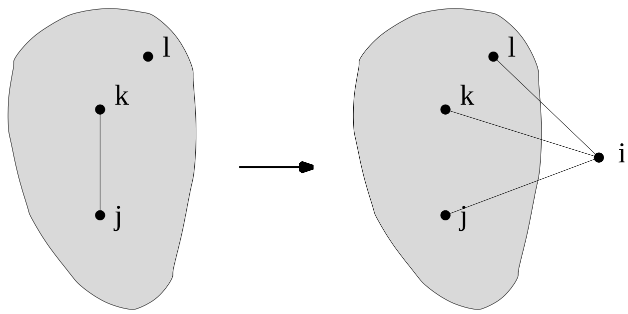
A graph is said to be *minimally rigid* if it is rigid and if there is no rigid graph having the same vertices but fewer edges. For example the graph in Figure 2(b) is minimally rigid while the one in Figure 2(a) is only rigid (we will discuss this notion more extensively in Section 4). This class of graphs is interesting to study, not only because it provides the least possible number of edges, but also because every rigid graph contains a minimally rigid graph. An extensive review of the state of the art regarding rigid graphs was provided in 1985 by Tay and Whiteley [3]. Among the main results, we will mention the following one about Henneberg sequences. A *Henneberg sequence* is a sequence of graphs beginning with the complete undirected graph on two vertices (i.e. a graph containing two vertices and a single edge joining them), and such that each graph can be obtained from the previous one by either a vertex addition or an edge splitting (see Figure 3) [4, 5]. One can show that, in a 2-dimensional space, every minimally rigid graph can be obtained as the result of a Henneberg sequence.

However, the *undirected* notion of rigidity does not suffice to characterize autonomous agent formations with *directed* distance constraints [4, 6]. Consider indeed the system represented in Figure 1(c). Although the underlying undirected graph is rigid, the structure of the formation may not be preserved: Agent 4 has an out-degree 1 and has thus only one distance constraint. If it moves on a circle of radius d_{43} centered on the position of agent 3, this constraint will remain satisfied. But, if agents 3 and 1 remain at the same position (and none of their constraints forces them to move) there will then generally be no position for agent 2 where it could satisfy its three distance constraints, which implies that the structure of the formation is in some way ill-posed.

In the control literature, the characterization of formations in which the structure will persist has started to be attempted using the notion of *rigidity of a directed graph* [4, 6]: a directed graph is called rigid if the structure of the corresponding formation is conserved along any



(a)



(b)

Figure 3. (a) *Vertex addition*: One adds a vertex and two incident edges. (b) *Edge splitting*: One replaces an edge (j, k) by a vertex i and three edges (i, j), (i, k) and (i, l) where l is another vertex of the original graph. Both operations preserve (minimal) rigidity [1, 3].

continuous move. Since this does not correspond to a simple transposition of the definition of rigidity for undirected graphs to directed graphs, we prefer here to call this notion *persistence* of a graph in order to avoid confusion. Some conjectures and results related to persistence do already appear in the literature, especially about minimally persistent graphs which are persistent graphs with a the least possible number of edges or sufficient conditions for a graph to be (minimally) persistent [4, 6]. In this paper, we propose a formal definition of persistence that provides a theoretical framework and allows us to prove the earlier partial results. We also derive some new properties of persistent graphs and give an operational criterion to determine if a graph is persistent.

The definition, which we provide in Section 2, has the following intuitive meaning: a graph is persistent if, provided that all the agents are trying to satisfy their distance constraints, the global structure of the agents formation is preserved. We will see that *rigidity of the underlying undirected graph is a necessary but not sufficient condition*. This will lead us to the notion of constraint consistence of graph, which is the additional condition for a rigid graph to be persistent. Intuitively, a graph is *constraint consistent* if every agent is able to satisfy all its distance constraints provided that all the others are trying to do so. We will then show that a graph is persistent if and only if it is rigid and constraint consistent. So, in Figure 1, as will subsequently become evident, (a) is not rigid and (c) is not constraint consistent. The only persistent graph is thus (b). Note that, although these notions are intuitively related to motion which is often large scale, we prefer to use equivalent and more convenient definitions based on relations between representations of the graph that are sufficiently close to each other.

In Section 3, we derive some of the main properties of persistent graphs and show the validity of the following criterion: A graph is persistent if and only if all the subgraphs obtained by removing edges leaving vertices with an out-degree larger than or equal to 3 so that their out-degree is 2 are rigid. This again explains why only (b) is persistent in Figure 1. We define then in Section 4 *minimal persistence* analogously to minimal rigidity. We discuss some differences and similarities between the two notions, and give a characterization of minimally persistent graphs. Finally, we turn our attention to cycle-free graphs in Section 5 and show some more powerful results that exist in this special case, such as a polynomial time criterion to decide persistence. No such polynomial criterion is indeed available in the generic case. A short version of the results present in this paper is available in [7].

2. PERSISTENCE FOR DIRECTED GRAPHS

A *representation* of a graph $G = (V, E)$ is a function $p : V \to \Re^2$. We say that $p(i) \in \Re^2$ is the *position* of the vertex i, and define the distance between two representations p_1 and p_2 of the same graph by

$$d(p_1, p_2) = \max_{i \in V} ||p_1(i) - p_2(i)|| \, .$$

Moreover, two representations p_1 and p_2 are *congruent* if the distance between the positions of every pair of vertices (connected by an edge or not) is the same in both of them: $||p_1(i) - p_1(j)|| = ||p_2(i) - p_2(j)||$ for all $i, j \in V$. Such representations can be obtained one from the other by a rotation, a translation and/or a reflection.

A *distance set* $\bar{d}$ for G is a set of distances $d_{ij} \geq 0$, defined for all edges $(i, j) \in E$. A distance set is *realizable* if there exists a representation p of the graph for which $||p(i) - p(j)|| = d_{ij}$ for all $(i, j) \in E$. Such a representation is then called a *realization*. Intuitively, a distance set $\bar{d}$ is realizable if it is possible to draw the graph such that the distance between the positions of any pair of vertices i, j connected by an edge is d_{ij}. Note that each representation p of a graph induces a realizable distance set (defined by $d_{ij} = ||p(i) - p(j)||$ for all $(i, j) \in E$), of which it is a realization.

Definition 1. *A representation p is* rigid *if there exists $\epsilon > 0$ such that for all realizations p' of the distance set induced by p and satisfying $d(p, p') < \epsilon$ are congruent to p. A graph is* generically rigid *if almost all its representations are rigid.*

The reasons for which we only require almost all representations to be rigid instead of all of them are detailed in Remark 1. As an example of the application of this definition, Figure 2(c) shows a graph representation p and a realization p' of the induced distance set - the lengths of all edges are indeed the same in p and p' - which is not congruent to p. Since such realizations p' can be found arbitrarily close to p, this latter is not rigid. On the other hand, it is possible to prove that the representations in Figures 2(a) and (b) are rigid. Although this definition is given here with the intention of applying it to directed graphs, rigidity is essentially an undirected notion, or rather, the definition takes no account of whether edges are directed or not. We remark also that our definition of (generic) rigidity is slightly different from those usually given in the literature, but the following equivalence can be proved.

Theorem 1. *The following conditions are equivalent for a graph $G = (V, E)$*

- *G is generically rigid.*
- *There exists a representation p of G for which any continuous displacement of the positions (such that at all time the positions of the vertices remain a realization of the distance set induced by p) is a rigid motion, i.e., is such that all these realizations are congruent to each other. (This is equivalent to the usual definition of generic rigidity [8]).*
- ***Laman's criterion [1, 8]:** There is a subset $E' \subseteq E$ satisfying the following two conditions:*
  (1) $|E'| = 2|V| - 3$.
  (2) For all $E'' \subseteq E', E'' \neq \varnothing, |E''| \leq 2|V(E'')| - 3$, where $|V(E'')|$ is the number of vertices that are end-vertices of the edges in E''.

As mentioned above, rigidity is an undirected notion, and is therefore insufficient to characterize persistence. The rigidity of a representation implies that if an external observer (or some physical properties) makes sure that the distance between the positions of any pair of vertices connected by an edge remains constant, then all the sufficiently close realizations of the induced distance set are congruent to each other. But, in our system of autonomous agents, there is no such external observer. Each agent is only aware of its own distance constraints, and can "move freely" as long as these particular constraints are satisfied. Agents that only have one constraint can thus move along a circle centered on the position of the only other agent of which they are aware. So, it could happen that because one agent is moving on such a circle, it becomes impossible for another agent to satisfy all its constraints, especially if this last one has 3 or more constraints. Consider for example the rigid graph representation in Figure 1(c). Agent 4 can move freely as long as it remains a distance d_{43} of 3. But if it does and if agents 1 and 4 remain stationary, there is no possible position or trajectory for agent 2 which will allow agent 2 to continuously satisfy its three constraints. In order to have a more formal definition of persistence, we first need to characterize mathematically the fact that each agent is trying to keep the distances from its neighbors constant.

Let us thus fix a directed graph G, desired distances $d_{ij} > 0$ for $\forall (i,j) \in E$, and a representation p. We say that the edge $(i,j) \in E$ is *active* if $||p(i) - p(j)|| = d_{ij}$, i.e, if the corresponding distance constraint is satisfied. We also say that the position of the vertex $i \in V$ is *fitting* for the distance set $\bar{d}$ if it is not possible to increase the set of active edges leaving i by modifying the position of i while keeping the positions of the other vertices unchanged. More formally, given a representation p, the position of vertex i is *fitting* if there is no $p^* \in \Re^2$ for which the following strict inclusion holds:

$$\{(i,j) \in E : ||p(i) - p(j)|| = d_{ij}\} \subset \{(i,j) \in E : ||p^* - p(j)|| = d_{ij}\} \qquad (1)$$

This condition intuitively means that the agent i cannot move (other agents staying fixed) so as to satisfy additional distance constraints without breaking some that it already satisfies, as shown in the example in Figure 4. A representation of a graph is a *fitting representation* for a certain distance set $\bar{d}$ if all the vertices are at fitting positions for $\bar{d}$. Note that any realization is a fitting representation for its induced distance set. From an autonomous agent point of view, a fitting representation is a state of a formation where no agent move, others

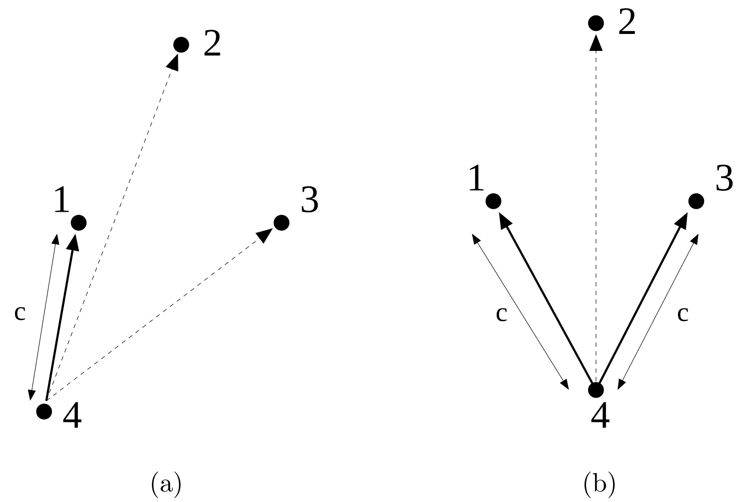


Figure 4. Suppose that $d_{41} = d_{42} = d_{43} = c$. The position of 4 in (a) is not fitting because it only makes $(4, 1)$ active while there exists a position that would make both $(4, 1)$ and $(4, 3)$ active. On the other hand, its position in (b) is fitting because no point can be at a distance $d_{42} = c$ of 2 in addition to being at a distance $d_{41} = d_{43} = c$ of 1 and 3.

staying fixed, so as to can improve the set of constraints that it satisfies. This notion of fitting representation is precisely what is needed to characterize "persistence". We can indeed extend the congruence requirement to all the (sufficiently close) representations fitting for a certain distance set instead of only the realizations of this distance set. In term of autonomous agents, instead of requiring the formation shape to be preserved only when all the distance constraints are satisfied, we can require it to be preserved as soon as no agent can improve its set of satisfied constraints, i.e., as soon as all agents are trying to satisfy all their constraints. We can thus now give a formal definition of persistence:

Definition 2. *A representation p is* persistent *if there exists $\epsilon > 0$ such that every representation p' fitting for the distance set induced by p and satisfying $d(p, p') < \epsilon$ is congruent to p. A graph is* generically persistent *if almost all its representations are persistent.*

This definition is similar to the one of rigidity, and it is thus natural to ask if there is a relation between the two notions. We will show that a generically persistent graph is always generically rigid, and give a necessary and sufficient condition for a generically rigid graph to be generically persistent. This condition is called the generic constraint consistence of a graph.

Definition 3. *A representation p is* constraint consistent *if there exists $\epsilon > 0$ such that any representation p' fitting for the distance set $\bar{d}$ induced by p and satisfying $d(p, p') < \epsilon$ is a realization of $\bar{d}$. A graph is* generically constraint consistent *if almost all its representations are constraint consistent.*

Intuitively, the constraint consistence of a representation means this. Suppose that for some representation, all constraints are fulfilled. Consider now a nearby representation that is fitting, i.e. one where each agent is at a fitting position, or satisfying as many distance constraints as it can, then in actual fact, every agent will be satisfying all its constraints. Consider the examples in Figure 5. Another example is provided in Figure 1, where (a) and (b) are constraint consistent while (c) is not. For suppose that p is a representation of 1(c) where all constraints are fulfilled, and p' is a nearby representation in which agent 1 and 3 have the same position and agent 4 has moved is such a way that it still satisfies its unique constraint; as already commented, there will be no new position possible for 2 which results in satisfying all three constraints; a fitting position for 2 is one in which only two constraints are satisfied. Therefore the same distance set as realized by p cannot be realized by p', and thus constraint consistence is lacking.

We have the following useful equivalences.

Theorem 2. *A representation is* persistent *if and only if it is* rigid *and* constraint consistent*. A graph is* generically persistent *if and only if it is* generically rigid *and* generically constraint consistent*.*

Proof: Observe first that we just have to prove this equivalence for a representation, since it will trivially imply the same equivalence for the graphs.

Let p be a rigid and constraint consistent representation, and p' a representation fitting for the distance set induced by p and satisfying $d(p,p') < \epsilon$ (where the ϵ is smaller than those coming from the application of the definitions of rigidity and constraint consistence to p). By the constraint consistence property, this fitting representation p' is necessarily a realization of the distance set induced by p. By rigidity, this implies that the representations p and p' are congruent. By Definition 2, p is therefore persistent.

Let us now consider a persistent representation of a graph G, the induced distance set $\bar{d}$, and the ϵ given by the definition of persistence. We are going to show that this ϵ is also appropriate for constraint consistence and rigidity. Because of the persistence of p, any representation p' fitting for $\bar{d}$ and satisfying $d(p,p') < \epsilon$ is congruent to p. It is thus by definition a realization of $\bar{d}$, and p is therefore constraint consistent. Now, if we consider a realization p' of $\bar{d}$ such that $d(p,p') < \epsilon$, it is by definition also a fitting representation for $\bar{d}$. The persistence of p implies then that it is congruent to p, which is therefore also rigid.

■

Remark 1. *In our definitions of generic rigidity, persistence and constraint consistence, a graph has a generic property if almost all its representations have the property. This "almost all" is loose phraseology meaning for all but members of a proper algebraic variety, i.e. for all but those points satisfying a set of nontrivial algebraic equalities. It indeed does not only excludes representations with several vertices having collinear or superposed positions, but also more delicate situations such as the one presented in Figure 6. One can see, using for example Laman's criterion (Theorem 1) that this graph is generically rigid. However, some of its representations are not rigid. Suppose indeed that in a certain representation p, the two triangles are congruent and the three transversal edges $(1,4)$, $(2,5)$ and $(3,6)$ are parallel and have the same length $d_{14} = d_{25} = d_{36}$. A representation p' obtained by translating one of the triangles in a way such that the distances along the transversal edges are preserved exists and will always be a realization of the distance set induced by p, but not congruent*

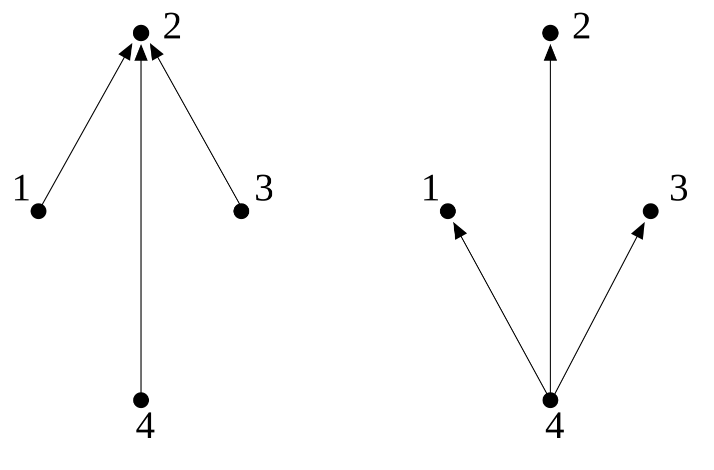


Figure 5. The graph represented in (a) is generically constraint consistent. Each of 1, 3 and 4 can indeed always satisfy its unique distance constraint. On the other hand, the one represented in (b) is not constraint consistent because there always exists a configuration of positions of 1, 2 and 3 such that 4 is unable to satisfy its three distance constraints.

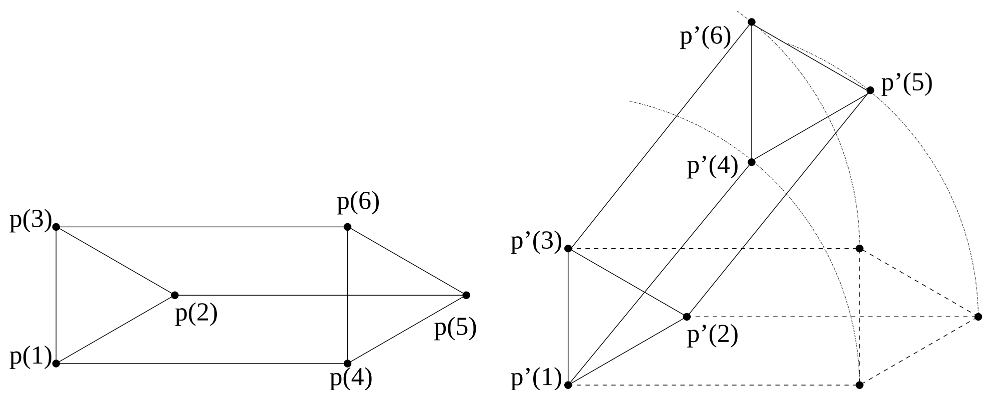


Figure 6. p is a representation of a generically rigid graph. However, (b) shows a representation p' of the same graph, fitting for the distance set induced by p, but which is not congruent to p.

to p. Such a representation is thus not rigid. For this reason, the properties of the graphs have to be considered as "generic". In the sequel, we however prefer to avoid further use of the words "generic", "generically", etc. Generically rigid (resp. generically persistent or generically constraint consistent) graphs will thus be called rigid (resp. persistent or constraint consistent) graphs.

3. CHARACTERIZATION OF PERSISTENT GRAPHS

In this section, we derive properties of persistent graphs and give a combinatorial criterion to decide persistence. We begin by giving a lower bound on the number of active edges, and a first sufficient condition for a graph to be constraint consistent. In the sequel, $d^-(i,G)$ and $d^+(i,G)$ designate respectively the in- and out-degree of the vertex i in the graph G. When no confusion is possible about the graph, we will use $d^-(i)$ and $d^+(i)$.

Lemma 1. *Let p be a representation of graph $G=(V,E)$, and i a vertex of this graph. If the position $p(i)$ is not collinear with two or more of its neighbors, then there exists $\epsilon > 0$ such that in every representation $p' \in B(p,\epsilon)$ (i.e., such that $d(p,p') < \epsilon$) fitting for the distance set induced by p, the number of active edges leaving i is at least $\min(2, d^+(i))$. Consequently, a graph in which all the vertices have an out-degree smaller than or equal to 2 is always constraint consistent.*

Proof: The proof of this lower bound is rather technical, and the reader may wish to skip the details at a first reading.

Let us consider a representation p' fitting for the distance set $\bar{d}$ induced by p. If the out-degree of i is 0 or 1, the set of possible positions that could make all the edges leaving i active is always non-empty (it is respectively $\Re^2$ or a circle). The position $p'(i)$ will then be fitting if and only if all the $d^+(i) = \min(2, d^+(i))$ edges are active.

If the out-degree of i is 2, we need the following result, which can be shown using simple geometric and continuity arguments:

Suppose there are given three non-collinear point $a,b,c \in \Re^2$ and d_{ab}, d_{ac}, d_{bc} the distances between each pair of points. There exists an $\epsilon(a,b,c) > 0$ such that for all $a', b' \in \Re^2$ satisfying $||a-a'||, ||b-b'|| < \epsilon(a,b,c)$, there exists $c' \in \Re^2$ such that $||b'-c'|| = d_{bc}, ||a'-c'|| = d_{ac}$. (Roughly speaking, if a, b and c are the agents meeting certain distance constraints, and if a and b move a small amount, then c can also be moved to ensure that again the distance constraints involving c are fulfilled.)

We can now show that

$$\epsilon_i = \min_{(i,j),(i,k)\in E} \epsilon(p(j), p(k), p(i)), \tag{2}$$

satisfies the required condition in the statement of Lemma 1. Let us indeed suppose that there is a representation $p' \in B(p,\epsilon)$ such that less than 2 active edges are leaving i, and take a set of two edges $(i,j),(i,k)$, containing the active edge leaving i if there is one. Observe that by hypothesis, $p(i)$, $p(j)$ and $p(k)$ are not collinear. By (2), there exists thus a point p^* such that $||p^* - p'(j)|| = d_{ij}$ and $||p^* - p'(k)|| = d_{ik}$, or equivalently a point p^* such that the strict inclusion (1) holds. The position $p'(i)$ and the representation p' are thus not fitting for $\bar{d}$, which contradicts our hypothesis. Hence we have proved the first part of the Lemma, that under the hypothesis given, the number of active edges leaving i as at least $\min(2, d^+(i))$

We now show the second part (about the constraint consistence) of the result. Observe first that in almost all representation, no vertex has a position collinear with two or more of its neighbors. Let us consider such a representation p of a graph G for which every vertex i has an out-degree $d^+(i) \leq 2$, and the induced distance set $\bar{d}$. If we take $\epsilon' < \epsilon_i, \forall i \in V$ where the ϵ_i comes from (2) for each vertex, then for any representation $p' \in B(p, \epsilon')$ fitting for $\bar{d}$, each vertex will be left by $\min(2, d^+(i)) = d^+(i)$ active edges, so that all the edges will be active. Every such p' is thus a realization of $\bar{d}$, and the representation p is thus constraint consistent. As we already mentioned, this can be done for almost all representations of G, which is therefore also constraint consistent.

■

The next proposition allows us to delete edges in a persistence graph and maintain persistence.

Proposition 1. *A persistent graph remains persistent after deletion of any edge* (i, j) *for which* $d^+(i) \geq 3$.

Proof: In the sequel, $G^* = (V, E^*)$ denotes the graph obtained by removing the edge (i, j) of $G = (V, E)$, which is persistent. Let us consider a realization p of G^* and the induced distance set $\bar{d}^*$. Observe that p can also be viewed as a representation of G, and the induced distance set is then $\bar{d} = \bar{d}^* \cup \{d_{ij}\}$. We assume here that no vertex has a position collinear with two or more of its neighbors (and thus that Lemma 1 can be applied), which is the case for almost all realizations.

We will first prove that any fitting representation of G^* for $\bar{d}^*$ sufficiently close to p is also a fitting representation of G for $\bar{d}$ (A). This will allow us to prove the persistence of G^* in a direct way *(B)*. Note that the proof of (A) is rather technical, and the reader may wish to skip it at a first reading.

(A) There exists $\epsilon_t > 0$ *such that every representation* $p' \in B(p, \epsilon_t)$ *fitting for* $\bar{d}^*$ *is also fitting for* $\bar{d}$.

Let us consider a representation $p' \in B(p, \epsilon_t)$ (where ϵ_t remains to be determined) fitting for $\bar{d}^*$ and such that no vertex has a position collinear with two or more of its neighbors (which is always the case if ϵ_t is sufficiently small). For each $k \in V \setminus \{i\}$, the result is trivial since the conditions of fittingness are the same for $\bar{d}^*$ and $\bar{d}$. We have thus to show that (for a sufficiently small ϵ_t), no possible position of i would make the same edges active (with respect to $\bar{d}$) as $p'(i)$ does and some additional one(s). Let E_i^* denote the set of active edges in p' with respect to $\bar{d}^*$ leaving i, i.e., the set of constraints of G^* that are satisfied by i in the representation p', which is fitting for $\bar{d}^*$. Obviously, these constraints are still satisfied if we consider p' as a representation of $\bar{d}$. We can consider two separate cases:

If $|E_i^*| \geq 3$. Because of the non-collinearity condition, only one possible position can make all the edges of E_i^* active, i.e.,

$$\{x^* \in \Re^2 \text{ s.t. } ||p^* - p'(k)|| = d_{ik}, \forall (i, k) \in E_i^*\} = \{p'(i)\}.$$

There is thus a fortiori no possible position that would make active (with respect to $\bar{d}$) all the edges that $p'(i)$ does (including those of E_i^*) and some additional one(s). The position $p'(i)$ is thus fitting for $\bar{d}$.

Now, if $|E_i^*| < 3$, the out-degree of i and Lemma 1 implies that $|E_i^*| = 2$. Let us denote by l and m the two vertices such that $(i,l),(i,m) \in E_i^*$. There are only two points that can make both edges of E_i^* active and thereby assure fittingness of vertex i for $\bar{d}^*$:

$$\{x \in \Re^2 \text{ s.t. } ||x - p'(l)|| = d_{ik}, k = l, m\} = \{p'(i), s\,(p'(i), p'(l), p'(m))\},$$

where $s : \Re^2 \times \Re^2 \times \Re^2 \to \Re^2 : (a,b,c) \to s(a,b,c)$ is a function which maps the point a into its reflection to the line bc. So, if there exists a position for i that would make active (with respect to $\bar{d}$) some other edge in addition to those of E_i^*, it can only be $s\,(p'(i), p'(l), p'(m))$. Moreover, this additional edge cannot belong to E^* for otherwise $p'(i)$ would not be fitting with respect to d^*. It must therefore be (i,j), only edge of $E \setminus E^*$. To achieve this proof, we use the following geometrical result:

Let a,b,c,d be four non-collinear points of $\Re^2$. There exists an $\epsilon\,(a,b,c,d)$ such that for all a',b',c',d' located at a distance smaller than $\epsilon(a,b,c,d)$ from respectively a,b,c,d, $||s(a',b',c') - d'||_2 \neq ||a-d||_2$.

So, since by hypothesis $p(j)$ is not collinear with $p(l)$ and $p(m)$, if $\epsilon_t < \epsilon\,(p(i), p(l), p(m), p(j))$, $s\,(p'(i), p'(l), p'(m))$ cannot make (i,j) or any other edge active in addition to those of E_i^*, which implies that $p'(i)$ is fitting for $\bar{d}$.

(B) G^* is persistent

Observe first that the set of representations of G is identically equal to the set of representation of G^*. By hypothesis, almost all representations of G are persistent. Moreover, almost all of them do not have a vertex with a position collinear with two or more of its neighbors. Let us thus take a realization p satisfying these two last conditions, and show that it is also persistent as a representation of G^*:

Consider a representation $p' \in B(p,\epsilon_p) \cap B(p,\epsilon_t)$ fitting for the distance set $\bar{d}^*$ (induced by p as a representation of G^*), where ϵ_t is given by *(A)*, and ϵ_p comes from the definition of persistence applied to p as a realization of G. By *(A)*, it is also a fitting representation of G for the distance set $\bar{d}$ (induced by p as a representation of G). Since p is persistent as a representation of G and $d(p,p') < \epsilon_p$, we know that p' is congruent to p. And since this can be done for any $p' \in B(p,\epsilon_p) \cap B(p,\epsilon_t)$, p is persistent as a representation of G^*. Moreover, as explained above, this result can be applied to almost all representations of G^*, which is therefore persistent.

■

One can see that the reasoning of this last proof can also be applied to graphs that are only constraint consistent, which leads to the following result:

Proposition 2. *A constraint consistent graph remains constraint consistent after deletion of any edge (i,j) for which $d^+(i) \geq 3$.*

An interesting corollary of Proposition 1 concerns the total number of degrees of freedom in the graph. The *number of degrees of freedom* of a vertex is the maximal dimension, over all generic representations of the graph, of the set of possible fitting positions for this vertex. In a 2-dimensional space, the vertices with zero out-degrees have two degrees of freedom, the

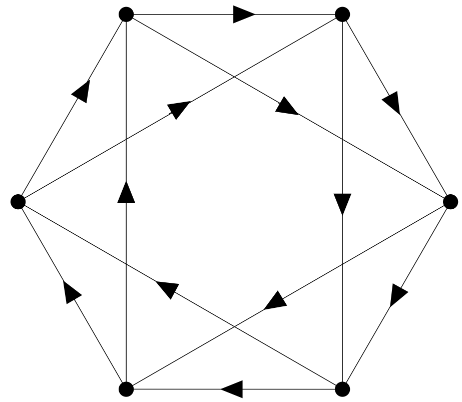


Figure 7. All the vertices of this rigid graph have an out-degree 2. By Lemma 1 it is thus constraint consistent and therefore persistent, but the number of degrees of freedom of each vertex is 0.

vertices with out-degrees 1 have one degree of freedom, and the others have none. Note that a vertex with no degree of freedom can have more than one possible fitting position. Observe indeed that there are in almost all situations two possible fitting positions for a vertex with out-degree 2. However, since this set contains a finite number of points, its dimension is still 0. The following result provides a natural bound on the sum of the degrees of freedom of individual vertices for a persistent graph.

Corollary 1. *The sum of the degrees of freedom of all the vertices of a persistent graph cannot exceed 3.*

Proof: Observe first that removing an edge leaving a vertex with an out-degree larger to or equal to 3 does not affect the number of degrees of freedom of this vertex. Let us now imagine that there exists a persistent graph $G = (V, E)$ for which this sum is larger than 3. Using recursively Proposition 1, we could obtain a persistent subgraph $G^* = (V, E^*)$ with the same number of degrees of freedom but without any vertex having an out-degree exceeding 2. In G^*, the number of degrees of freedom of a vertex i is thus $2 - d^+(i, G^*)$. So, if F is the sum of the degrees of freedom of all the vertices of the graph, we have $F = 2|V| - |E^*|$. $F > 3$ would then mean that $|E^*| < 2|V| - 3$, which by Theorem 1 is impossible for a persistent (and thus rigid) graph.

■

Note that the total of three degrees of freedom is an upper bound. There are persistent graphs which vertices do not have any degree of freedom, as shown in Figure 7.

We have shown in Proposition 1 that a persistent graph remains persistent after deletion of any edge (i, j) for which $d^+(i) \geq 3$. After successive deletions, we can thus reach in this way a persistent graph whose vertices all have an outgoing degree that is smaller than or equal to 2. In the next theorem we prove that a graph is persistent if and only if all the graphs obtained in this way are rigid.

Theorem 3. *A graph is persistent if and only if all those subgraphs are rigid which are* *obtained by removing outgoing edges from vertices with out-degree larger than 2 until all the vertices have an out-degree smaller than or equal to 2.*

Proof: Let us consider a graph $G = (V, E)$ and Σ the set of all the subgraphs S of G satisfying for every vertex $i \in V$, $d^+(i, S) = \min(d^+(i, G), 2)$. We prove separately the following two implications:

- *If G is persistent, any $S \in \Sigma$ is rigid.*
  Since it is possible to obtain S from G only by removing edges leaving vertices with an out-degree larger or equal to 3, Proposition 1 guarantees the persistence of G^* and thus its rigidity.

- *If every $S \in \Sigma$ is rigid, G is persistent.*
  Let us suppose that G is not persistent, and prove (to obtain a contradiction) that this implies that at least one graph of Σ is not rigid. We begin by showing this result for a particular representation of the graph, and then we generalize to the graph.

  Consider a representation p of G which is not persistent and to which Lemma 1 can be applied. We are going to show that p is not rigid for at least one $S \in \Sigma$. By the definition of persistence, for all $\epsilon > 0$, there exists a representation p' of G, fitting for the distance set $\bar{d}$ induced by p, and not congruent to p. Let us consider such a p' and build a subgraph $G^* = (V, E^*)$ of all the active edges in p'. Lemma 1 implies that $d^+(i, G^*) \geq \min(2, d^+(i, G))$ for all $i \in V$. Therefore, by removing some additional edges leaving vertices with an out-degree larger than 2, we can obtain a subgraph $S = (V, E^s)$, with $E^s \subseteq E^* \subset E$, and such that, for all $i \in V$, $d^+(i, G^*) = \min(2, d^+(i, G))$. Notice that we can regard S as being obtained by deletion of edges from G^*, or by deletion of edges from G.

  We denote now by $\bar{d}^s$ the subset of $\bar{d}$ corresponding to S. By construction, p and p' are realizations of $\bar{d}^s$, but p' is not congruent to p. So, for any value of ϵ, there is always a subgraph $S \in \Sigma$ such that there exists a realization (of the distance set induced by p as a representation of S) not congruent to p. The finite number of elements of Σ implies then that p is non-rigid for at least one $S \in \Sigma$.

  We now generalize this result to the graph G. By definition, G is persistent if almost all its representations are persistent. Since here G is not persistent, more than almost none of its representations are not persistent. Moreover, Lemma 1 can be applied to almost all the representations of a graph. It follows thus that more than almost none of the representations of G are at the same time not persistent and such that Lemma 1 can be applied. It is proved above that any such representation p fails to be rigid as a representation of at least one graph belonging to Σ. The finite number of such graphs implies then the existence of a graph $S \in \Sigma$ having more than almost none of its representations that are not rigid, i.e., a graph $S \in \Sigma$ which is not rigid.

■

This last result provides a non-polynomial time algorithm to check the persistence of a graph: it suffices to check the rigidity of all subgraphs obtained by deleting edges leaving vertices with

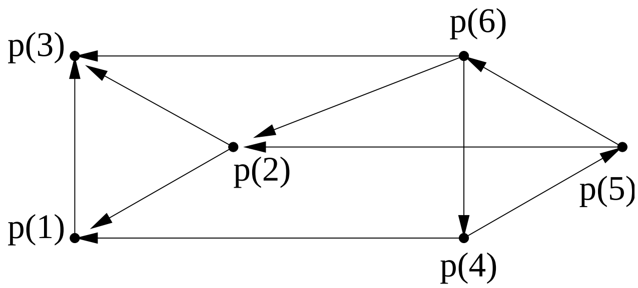


Figure 8. Example of non persistent representation of a persistent graph. Removing $(6, 2)$ (that leaves a vertex with out-degree 3) yields indeed the non-rigid representation of Figure 6.

out-degree larger or equal to 3 until all vertices have an out-degree smaller or equal to 2. An algorithm with a smaller complexity would be useful in the case of large graphs, especially if there is a high number of vertices with a high out-degree, but no such algorithm has been found yet and at the time of writing it is still unclear if the problem of determining if a directed graph is persistent can be solved in a polynomial time. However, polynomial time complexity results exist in some particular cases. In Section 5, we show that for cycle-free graphs, persistence can be checked in polynomial time, and in the next section we introduce the related notion of minimal persistence and prove a decision criterion that can also be checked in a polynomial time.

Remark 2. *The results of Proposition 1 and Theorem 3 concern the persistence of graphs. In the proofs of both of them, we use the fact that Lemma 1 can be applied to almost all representations. Actually, it is possible to prove similar results concerning the persistence of a representation, if one assumes that it satisfies the conditions of Lemma 1:*

(a) A persistent (resp. constraint consistent) representation of a graph G such that no vertex has a position collinear with two or more of its neighbors is also a persistent (resp. constraint consistent) representation of any graph G^* obtained by deletion of any edge (i, j) for which $d^+(i) \geq 3$.

(b) A representation such that no vertex has a position collinear with two or more of its neighbors is persistent for a graph if and only if it is rigid for all those subgraphs which are obtained by removing outgoing edges from vertices with out-degree larger than 2 until all the vertices have an out-degree smaller than or equal to 2.

Note that the proofs are almost identical to those of respectively Proposition 1 and Theorem 3. An example of application of this last result to a representation is shown in Figure 8. One can see that p *is a representation of a persistent graph. However, if both triangles are congruent and if the three transversal edges* $(4, 1)$, $(5, 2)$ *and* $(6, 3)$ *are parallel and have the same length* $d_{14} = d_{25} = d_{36}$, *p is not persistent. One can indeed see that* 6 *has an out-degree 3 and that the representation obtained after deletion of* $(6, 2)$ *is not rigid, as shown in Figure 6.*

4. MINIMAL PERSISTENCE

In this section we define the notion of minimal persistence, analogously to minimal rigidity. We then discuss the main properties of minimally persistent graphs, and show some similarities and difference between minimal persistence and minimal rigidity.

But first, we recall a few facts about minimal rigidity. One way to define the concept is to say that a graph is *minimally rigid* if it is rigid and if there exists no rigid graph with the same number of vertices and a smaller number of edges. Another way is to say that a graph is *minimally rigid* if it is rigid and if no single edge can be removed without losing rigidity. These two definitions are provably equivalent and lead to the following criterion: A graph $G = (V, E)$ is minimally rigid if it is rigid and if $|E| = 2|V| - 3$ (with an exception if $|V| = 1$). Moreover, a necessary and sufficient condition for a graph to be rigid is the presence of a minimally rigid (edge) subgraph. This can be seen using for example Laman's criterion (Theorem 1).

A *Henneberg sequence* is a sequence of graphs $G_2, G_3, \ldots, G_{|V|}$ such that G_2 is the complete (undirected) graph with two vertices, and G_{i+1} $(i \geq 2)$ can be obtained from G_i by performing either a vertex addition or an edge splitting (see [4, 5]). These operations are defined in Figure 3, and one can show that they preserve minimal rigidity. Moreover, every minimally rigid graph can be obtained as the result of a Henneberg sequence [3].

We now define minimal persistence as follows:

Definition 4. *A persistent graph is* minimally persistent *if it is persistent and if no edge can be removed without losing persistence.*

A first important and surprising difference with the concept of minimal rigidity is that a graph having a minimally persistent (edge) subgraph is not necessarily persistent, as shown in the example in Figure 9. More generally, unlike the case of rigidity, it is possible to obtain a non-persistent graph by *adding* edges to a persistent graph. From this observation arises the question: does there exist a minimally persistent graph from which one could obtain a persistent graph after deletion of *more than* just one edge? We will see that Proposition 3 provides a negative answer since it states that the number of edges of a minimally persistent graph is uniquely determined by the number of its vertices. A first necessary condition for a persistent graph to be minimally persistent is immediate from Proposition 1: the absence of vertex with an out-degree exceeding 2. On the other hand, a sufficient condition is minimal rigidity: suppose indeed that one removes an edge of a persistent minimally rigid graph; then the obtained graph would by definition not be rigid and therefore not persistent. We will see in the sequel that this condition is also necessary, i.e. that any minimally persistent graph is minimally rigid.

As explained above, every minimally rigid graph can be obtained from an initial seed of two vertices and one edge by a sequence of vertex additions and edge splittings. We define here the directed version of these operations as in [4] by giving a direction to the added arrows in a way such that the out-degrees of the already existing vertices are not affected, as represented in Figure 10. To perform a *(directed) vertex addition* on a graph $G = (V, E)$, one adds a vertex and two edges from this vertex to different vertices of V. The *(directed) edge splitting* consists

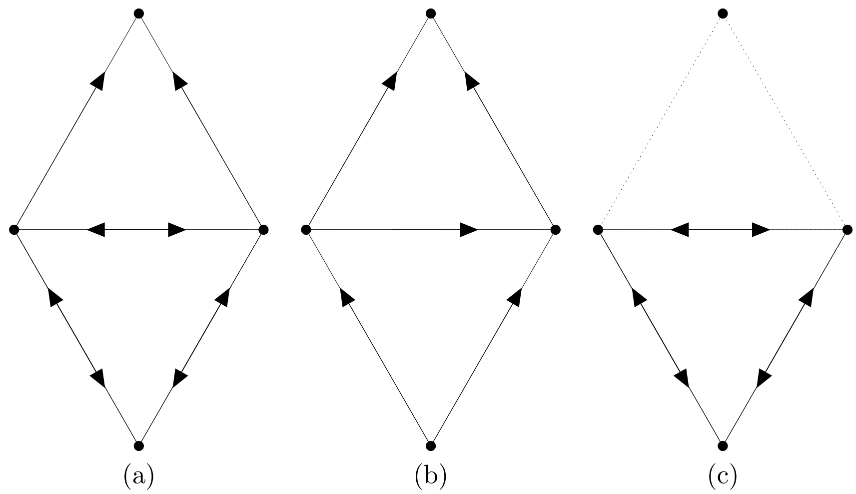
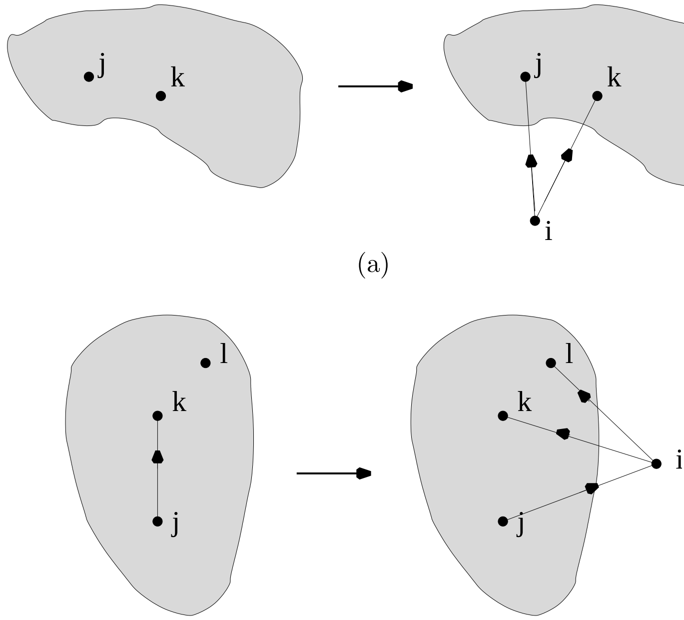


Figure 9. The graph represented in (a) has a minimally persistent subgraph (b). However, by Theorem 3, it is not persistent because the subgraph represented in (c) is not rigid. In the corresponding multi-agent system, this could be seen as arising from a combination of unfortunate selections among the various possible information architectures available to the three agents of the cycle.

in removing one edge $(j, k) \in E$ and adding a vertex i and three edges (j, i), (i, k) and (i, l) for some $l \in V, l \neq j, k$. In the sequel, these operations will always be considered with the directed meaning. A *Henneberg sequence (directed case)* is then a sequence of graphs $G_2, G_3, ..., G_{|V|}$, such that each graph G_{i+1} $(i \geq 2)$ can be obtained by performing a vertex addition or an edge splitting on G_i, and G_2 is a graph of two vertices connected by one directed edge. As in the undirected case, all the graphs of such a sequence are minimally rigid. Moreover, since the out-degree of each of their vertices is always smaller or equal to two, Lemma 1 guarantees that they are also constraint consistent and thus minimally persistent. This implies that one can always assign a direction to all the edges of a minimally rigid undirected graph such that the resulting graph is minimally persistent. It is indeed possible to obtain every minimally rigid undirected graph by performing a sequence of (undirected) vertex additions and edge splitting on an initial seed of two vertices and one edge. [In order to obtain a minimally persistent graph, one can then simply perform the same sequence of the directed version of these operations]. However, it is still an open question as to whether, given an undirected rigid (but not minimally rigid) graph, there exists an assignment of directions to the edges such that the resulting directed graph is persistent.

Since every undirected minimally rigid graph can be obtained as the result of a Henneberg sequence, and since there always exists a minimally persistent graph resulting from the same sequence, it is natural to ask if every minimally persistent graph can be obtained in that way. Unfortunately, the existence of counterexamples force us to answer negatively to this question. Consider indeed the cycle of length 3 or any minimally persistent graph for which all the vertices have a positive out-degree. Since both vertex addition and edge splitting conserve the out-degree of all the already existing vertices, and since the first graph (G_2) of a Henneberg sequence contains a vertex with a zero out-degree, they cannot be obtained as a result of a



(a)

(b)

Figure 10. Representation of the directed version of the *vertex addition* (a) and the *edge splitting* (b).

Henneberg sequence. Actually, not all counterexamples have this property. Figure 11 shows indeed another counterexample with a vertex that has a zero out-degree, as we shall now argue (We were not able to find a smaller graph with the same properties). Let us indeed consider each of the possible graphs from which this graph could be obtained by the last of the directed Henneberg sequential operations. Because of their in- and out-degrees, the only vertices that could have been added are 7 and 2, and one would have used edge splitting in either case. But, if 7 is removed by the reverse operation, one must introduce the edge $(4,6)$ since the edge $(4,3)$ is already present in the graph. This will create a double edge, i.e. a cycle of length 2. Call the resulting graph G'. By Laman's criterion, a graph $G' = (V', E')$ satisfying $|E'| = 2\,|V'| - 3$ and having a subgraph G'' for which $|E''| > 2\,|V''| - 3$ is not rigid. Identify G'' with the double edge and the vertices joining by the double edge. Then we see that G' cannot be rigid and thus persistent. On the other hand, if 2 is removed, the graph obtained G', which includes a new edge $(3,6)$, has a subgraph $G'' = (V'', E'')$ with $V'' = \{3,4,6,7\}$ and $|E''| = 6 > 2\,|V''| - 3$; this again prevents the graph G' from being rigid and therefore minimally persistent. Note that in both cases, the absence of persistence comes from the absence of rigidity.

A more comprehensive examination of Hennberg sequences for directed graphs is undertaken in [9]; by expanding the set of allowed operations beyond directed versions of vertex addition and edge splitting, one can construct all minimally persistent graphs through a sequence of standard operations. We now show that minimal rigidity is not just a sufficient condition but also a necessary condition for a persistent graph to be minimally persistent.

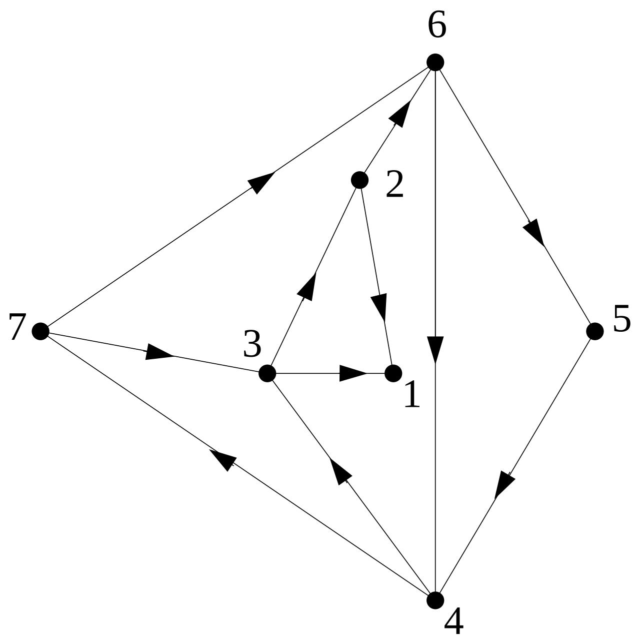


Figure 11. One can verify that this graph is minimally persistent. However, it cannot be obtained from a smaller minimally persistent graph by either vertex addition or edge splitting (depicted in Figure 10).

Proposition 3. *A graph $G = (V, E)$ is minimally persistent if and only if it is persistent and satisfies $|E| = 2|V| - 3$.*

Proof: First, observe that a persistent graph $G = (V, E)$ satisfying $|E| = 2|V| - 3$ is minimally persistent; for by Laman's criterion (Theorem 1), removing one edge would indeed mean losing rigidity and thus persistence.

Conversely, let us now suppose, to obtain a contradiction, that there exists a minimally persistent graph $G = (V, E)$ for which $|E| > 2|V| - 3$ (The case of the reverse inequality is trivial since G could then not be rigid and therefore persistent). We are going to show that there always exists a persistent edge subgraph, which contradicts our minimality hypothesis.

If there is a vertex with an out-degree larger than 2, one can use Proposition 1 to build this subgraph. But, if no such vertex exists, we know by Lemma 1 that every subgraph of G is constraint consistent. Moreover, because G is rigid, Laman's criterion provides a rigid subgraph $G' = (V', E')$ for which $|E'| = 2|V'| - 3$. Since G' is also constraint consistent, it is persistent.

■

It is actually possible to give a more specific characterization of minimal persistence that relies on the vertex out-degrees.

Theorem 4. *A rigid graph (with more than one vertex) is minimally persistent if and only if* *one of the following two conditions is satisfied.*

- *Three vertices have an out-degree 1 and all the others have an out-degree 2.*
- *One vertex has an out-degree 0, one vertex has an out-degree 1, and all the others have an out-degree 2.*

Proof:

- *Sufficient condition:* By Lemma 1, a rigid graph satisfying either of the two above conditions is constraint consistent and therefore persistent. Moreover, each of these two conditions implies that the number of edges is $|E| = 2|V| - 3$.[Observe that the number of edges is precisely the sum of the out degrees of all the vertices]. It follows then from Proposition 3 that G is minimally persistent.
- *Necessary condition:* Let us take $G = (V, E)$, minimally persistent. By Proposition 1, there is no vertex $i \in V$ with an out-degree larger than 2. [If the contrary were the case, one could indeed remove one edge and obtain a persistent graph with less edges]. Since by Proposition 3 the sum of the out-degrees is $|E| = 2|V| - 3$, G satisfies necessarily one of the two conditions of the present theorem.

■

The use of Laman's criterion in the proof of Proposition 3 and in the comments about the counterexample of Figure 9 can introduce some confusion about the cycles of length 2, i.e., the "double edges". It has to be indeed clearly understood that although they are equivalent from an undirected point of view (and thus for any undirected notion such as rigidity) they are considered as two different edges. However, such a cycle could never belong to the minimally rigid subgraph required by Laman's criterion (Theorem 1) and therefore appear in any minimally rigid graph. Taking two such edges as an edge subset E'' would indeed yield $|E''| = 2 > 1 = 2|V(E'')| - 3$. If an undirected graph contains two edges between the same vertices, one can thus always remove one of them without affecting the rigidity of this graph. There is an analogous result for persistent graphs.

Proposition 4. *If a persistent graph contains two edges incident to the same pair of vertices (and having opposite directions), then at least one of the two graphs obtained by removing one of the edges is also persistent. A minimally persistent graph never contains two such edges.*

Proof: Let $G = (V, E)$ be a persistent graph and $i, j \in V$ two vertices such that $(i, j), (j, i) \in E$. If $d^+(i) \geq 3$ or $d^+(j) \geq 3$, the result is trivial by Proposition 1. If it is not the case, then (i, j) and (j, i) are edges of every edge subgraph S satisfying $d^+(k, S) = \min(2, d^+(k, S))$ for all $k \in V$, and the rigidity of such a subgraph will thus not be affected if one of the two edges is removed. Theorem 3 guarantees then that one can remove either one of these two edges an obtain a persistent graph.

■

However, unlike in the case of rigidity, adding an edge to two already connected vertices in a directed graph *does not necessarily* preserve persistence, as shown in Figure 9 (a) and (b).

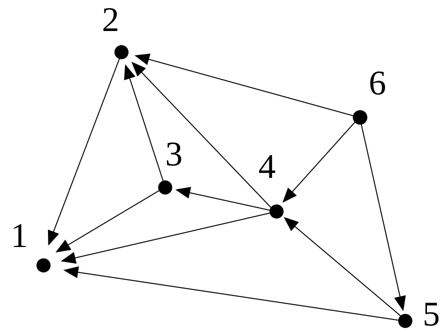


Figure 12. Example of cycle-free persistent graph. The numbers correspond to an order in which the vertices can be added according to Theorem 5.

5. CYCLE-FREE GRAPHS

In this section, we derive a simple criterion to decide the persistence of cycle-free graphs. We also show an explicit way to build all the persistent cycle-free graphs.

Proposition 5. *A graph obtained by adding one vertex to a graph $G = (V, E)$ and at least two edges leaving this vertex is persistent if and only if G is persistent.*

Proof: Let $G^* = (V^*, E^*)$ be the graph obtained by adding a vertex i and at least two edges leaving i to G. We call Σ the set of subgraphs S of G satisfying $d^+(k, S) = \min(2, d^+(k, G))$ for all $k \in V$, and Σ^* the corresponding set of subgraphs S^* of G^*. By the criterion of Theorem 3, it is sufficient to prove the equivalence between the existence of a non-rigid $S^* \in \Sigma^*$ and the existence of a non-rigid $S \in \Sigma$. In this purpose, we use the following result already mentioned in the introduction:

A graph obtained by adding one vertex to a graph $G = (V, E)$ and two edges from this vertex to other vertices is rigid if and only if G is rigid [1].

Suppose that there exists a non-rigid $S \in \Sigma$ as described above. The graph S^* obtained by adding i and two of the edges of $E^* \setminus E$ is not rigid. Moreover, it belongs to Σ^* since it is a subgraph of G^* and each one of its vertices has an out-degree $\min(2, d^+(k, G^*))$. Conversely if there exists a non-rigid $S^* \in \Sigma^*$, the subgraph S of G obtained by removing i from S^* is also not rigid, and belongs to Σ since each one of its vertices has an out-degree $\min(2, d^+(k, G))$. There is thus a non rigid graph in Σ if and only if there is a non rigid graph in Σ^*.

■

We thus know that a cycle-free graph obtained by successively adding vertices all with out-degree 2 to an initial seed of one directed edge connecting two vertices is persistent. We will now show that every persistent cycle-free graph can be obtained in such a way, as shown in the example in Figure 12, and derive from that fact a simple criterion for persistence in the particular case of cycle-free graphs.

Theorem 5. *A cycle-free graph having more than one vertex is persistent if and only if*

- *One vertex (called the leader) has an out-degree 0;*

- *One vertex (called the first follower) has an out-degree 1 and the corresponding edge is incident to the leader;*
- *Every other vertex has an out-degree larger or equal to 2.*

Moreover, every such graph can be obtained from an initial seed composed by the leader and first follower by adding vertices one by one in the way described in Proposition 5, i.e., each vertex is added with all its incident edges that are outgoing.

Proof: Let us consider a cycle-free graph $G = (V, E)$. Its vertices can be numbered in such a way that the numbering of the origin of an edge is always larger than the numbering of its destination (usual topological sort for cycle-free directed graphs [10, 11]). We relabel the vertices such that $n(i) = i, \forall i \in V$ (where $n : V \to 1..|V|$ is such a numbering), and have then

$$d^+(i) < i \qquad \text{and} \qquad (i,j) \in E \Rightarrow i > j. \tag{3}$$

If the graph is persistent, it is rigid. Corollary 1 and (3) imply then that $d^+(1) = 0$, $d^+(2) = 1$, $(2,1) \in E$ and for all others $i \in V$, $d^+(i) \geq 2$. The condition about the out-degrees is thus satisfied.

Conversely, if the graph satisfies the out-degrees condition described above, (3) implies that the vertices 1 and 2 are respectively the leader and the first follower and that $(2,1) \in E$. Moreover, the vertex labelled $|V|$ has $d^+(|V|) \geq 2$ and $d^-(|V|) = 0$ (if $|V| > 2$). Removing it does thus not modify the out-degree of the other vertices, and leads to a smaller cycle-free graph that still satisfies the out-degree condition. Doing this recursively, one finally obtains a graph on only two vertices: $(\{1,2\}, \{(2,1)\})$. G can thus be built by adding, one by one, $|V|-2$ vertices with (two or more) outgoing edges to this initial seed on two vertices. And since this seed is persistent, it follows from Proposition 5 that G also is.

■

This result provides an algorithm with a low complexity to decide the persistence of a cycle-free graph. Moreover, if we apply it to a minimally persistent graph, we get the following corollary:

Corollary 2. *A cycle-free minimally persistent graph with more than one vertex always has a* leader-follower *structure (see Theorem 5) and can always be obtained as the result of a Henneberg sequence containing only vertex additions.*

6. CONCLUSIONS AND FURTHER WORKS

As mentioned in the previous sections, several questions remain open the main ones being the existence of a polynomial time criterion to decide if a graph is persistent, and an algorithm to assign directions to the edges of a rigid graph in order to obtain a persistent graph. We intend to examine the possible application to these issues of a pebble game approach, which can be used to determine in a polynomial time if an undirected graph is rigid [12].

Among the possible extensions of this work, one can remark that we always assumed that the graph representations lie in $\Re^2$. From a practical point of view, it would be desirable to extend the results to $\Re^3$. However, this does give rise to new difficulties. For undirected graphs, there is no known equivalent of Laman's theorem in three dimensions, and not all minimally rigid graphs can be obtained by Henneberg sequences. Moroever, other issues appearing in higher dimensions such as the possible presence of two leaders would also have to be treated [13, 14, 15]. Besides, we showed that a minimally persistent graph cannot always be built by performing a sequence of the two operations depicted in Figure 10 on an initial seed of two vertices. It is interesting to study if it is possible to obtain all minimally persistent graphs in such a way using other types of minimal persistence preserving operations. A partial answer to this problem is provided in [9]. Since we showed in Figure 9 that one cannot generally add edges indefinitely to a persistent graph without losing persistence, we may also be able to define and characterize *maximally persistent* graphs. Finally, another issue would be to consider the robustness of a persistent graph. One could assign to each edge a probability of breakdown and to each unconnected pair of vertices a probability of parasite edge appearance. There might be in this case a maximally robust persistent graph, i.e., a graph for which the probability of losing persistence is minimal. It is evident that if there is a finite probability of losing an edge, it would be desirable to have persistence both with and without it. This observation emphasizes the need to understand better the circumstances under which edges can be added to a persistent graph without losing the persistence property.

REFERENCES

1. G. Laman. On graphs and rigidity of plane skeletal structures. *J. Engrg. Math.*, 4:331–340, 1970.
2. W. Whiteley. Infinitesimally rigid polyhedra. I. Statics of frameworks. *Trans. Amer. Math. Soc.*, 285(2):431–465, 1984.
3. T. Tay and W. Whiteley. Generating isostatic frameworks. *Structural Topology*, (11):21–69, 1985.
4. T. Eren, B.D.O. Anderson, A.S. Morse, and P.N. Belhumeur. Information structures to secure control of rigid formations with leader-follower structure. In *Proc. of the American Control Conference*, pages 2966–2971, Portland, Oregon, june 2005.
5. L. Henneberg. Die graphische Statik der starren Systeme. Leipzig, 1911.
6. J. Baillieul and A. Suri. Information patterns and hedging brockett's theorem in controlling vehicle formations. In *Proc. of the 42nd IEEE Conf. on Decision and Control*, volume 1, pages 556–563, Hawaii, december 2003.
7. J.M. Hendrickx, B.D.O. Anderson, and V.D. Blondel. Rigidity and persistence of directed graphs. In *Proceedings of the 44th IEEE Conference on Decision and Control*, Seville, Spain, december 2005.
8. W. Whiteley. Some matroids from discrete applied geometry. In *Matroid theory (Seattle, WA, 1995)*, volume 197 of *Contemp. Math.*, pages 171–311. Amer. Math. Soc., Providence, RI, 1996.
9. J.M. Hendrickx, B. Fidan, C. Yu, B.D.O. Anderson, and V.D. Blondel. Elementary operations for the reorganization of persistent formations. *Preprint*.
10. T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein. Introduction to algorithms. pages 549–552. MIT Press, second edition, 2001.
11. S.O. Krumke and H. Noltemeier. Graphentheorische konzepte und algorithmen. B.G. Teubner, 2005.
12. D. J. Jacobs and B. Hendrickson. An algorithm for two-dimensional rigidity percolation: the pebble game. *J. Comput. Phys.*, 137(2):346–365, 1997.
13. C. Yu, J.M. Hendrickx, B. Fidan, B.D.O. Anderson, and V.D. Blondel. Three and higher dimensional autonomous formations: Rigidity, persistence and structural persistence. *Submitted*.
14. J.M. Hendrickx, B. Fidan, C. Yu, B.D.O. Anderson, and V.D. Blondel. Rigidity and persistence of three and higher dimensional formations. In *Proceedings of the First International Workshop on Multi-Agent Robotic Systems (MARS 2005)*, pages 39–46, Barcelona, Spain, september 2005.
15. C. Yu, J.M. Hendrickx, B. Fidan, B.D.O. Anderson, and V.D. Blondel. Structural persistence of three dimensional autonomous formations. In *Proceedings of the First International Workshop on Multi-Agent Robotic Systems (MARS 2005)*, pages 47–55, Barcelona, Spain, september 2005.