Journal of Integer Sequences, Vol. 25 (2022),
Article 22.1.3

Generating All Rigidity Circuits on at Most 10 Vertices and All Assur Graphs on at Most 11 Vertices

Ciaran Mc Glue
Maynooth University (NUI)
Maynooth
Co. Kildare
Ireland
ciaranmcglue90@gmail.com

Klara Stokes[1]
Umeå University
Department of Mathematics and Mathematical Statistics
901 87 Umeå
Sweden
klara.stokes@umu.se
and
Maynooth University (NUI)
Maynooth
Co. Kildare
Ireland

Abstract

We present an inventory and the enumeration of all non-isomorphic rigidity circuits on up to 10 vertices, as well as all non-isomorphic Assur graphs on up to 11 vertices. Assur graphs and Baranov trusses are closely related. We clarify the relation between Baranov trusses and $(2, 3)$-tight graphs on the one hand, and between Assur groups and Assur graphs on the other hand.

[1]The second author received financial support from Knuth and Alice Wallenberg Foundation: grant numbers 2020.0001 and 2020.0007, and from the Catalan Government under grant 2017 SGR 00705 and by the Spanish Ministry of Economy and Competitivity under grant TIN2016-80250-R.

1 Introduction

When originally described by Russian engineer Leonid Assur in 1914, Assur groups were defined as kinematic chains that have zero degrees of mobility, and from which simpler structures of the same mobility cannot be obtained. The terminology used in most of the literature on this topic comes from engineering. For example, an Assur group is not a group in the mathematical sense. In 2010, Servatius et al. gave the concept of Assur group a formal mathematical model in terms of graph theory [20]. The resulting *Assur graphs* are the topic of this article. The term Assur group is, however, commonly used in the engineering literature, as will be reflected below, when we briefly survey the applications of Assur graphs in engineering.

A *planar linkage*, or *kinematic chain* (with revolute joints only) may be described as a collection of rigid convex polygonal links that are joined at revolute joints. In the theory of mathematical rigidity, planar linkages correspond to so-called *planar body-pin frameworks*; the rigid polygonal links are modeled by rigid bodies, and the joints are called *pins*; see the discussion in Section 3. Within that context, the word pin denotes a zero-dimensional hinge between a pair of planar bodies.

In this article we will, however, use the term *pinned* to denote a joint that has its position fixed in the plane, and we will then say that the joint or vertex is *grounded* or *pinned.* We will also talk about *links* instead of bodies.

An *Assur group* is then a collection of rigid polygonal links, joined at their vertices. Some of the vertices of an Assur group are pinned/grounded, such that the linkage contains no rigid sublinkage and such that it is not only rigid, but due to the pinned/grounded joints, allows no motion at all. As Hahn et al. mention [7], Assur groups are useful when describing different types of mechanisms and robotic structures—for example, parallel robots such as Stewart Platforms.

Assur groups are closely related to Baranov trusses. A *Baranov truss* is a planar linkage that is rigid but contains no rigid sublinkage. A Baranov truss admits rigid motions, but an Assur group does not. By removing any link of a Baranov truss and grounding its joints, an Assur group is obtained. This property has also been used to define Baranov trusses. Rojas and Thomas [19] defined them, for example, as "non-overconstrained linkage(s) with zero-mobility from which an Assur group can be obtained by removing any of its links...". In their article they relate Baranov trusses with given numbers of links to the number of Assur groups these Baranov trusses can generate upon removal of a link.

Orki et al. [17] discuss the use of Assur tensegrity structures to model the mechanics of a crawling caterpillar. In this instance, an Assur group is considered to be a structure that maintains its stability through "tensional integrity" through the use of cables (tension) and struts (compression). The internal musculature of the caterpillar is modelled by these tensegrity structures such that the non-moving part of the caterpillar's body is considered a stable object, and the moving parts are compressing and expanding to move the caterpillar forwards.

Shai et al. use a similar concept involving tensegrity structures modelled using Assur

groups [21]. In their article, an adjustable robot is constructed such that, internally, it has zero degrees of freedom, but it is still mobile and can bear external loads. This provides a motivation to be able to measure the degree of freedom experienced by such frameworks.

Jacobs et al. [14] describe an algorithm involving placing pebbles at joints within a framework to describe its inner degrees of freedom. This algorithm can be applied to planar frameworks to determine their rigidity and, while not the focus of this article, is a useful tool in Assur group verification and analysis.

Both the engineering community and the mathematical community could benefit from having examples of Assur graphs readily available. In this article we generate an inventory of small Assur graphs, following an algorithm described by Servatius et al. [20]. In the process we also generate an inventory of small rigidity circuits.

Huang and Ding provided a database of Baranov trusses with up to 13 links [9] and used it to build a database of Assur groups with up to 12 links [10], using the fact that the removal of any link and grounding its joints gives an Assur group. As will be made clear in Section 3, Assur graphs provide a graph theoretical model for Assur groups, but there may be many Assur graphs modelling the same Assur group. This explains why our inventory of Assur graphs is different from the one presented by Huang and Ding, and the number of graphs also differ. More precisely, they count the number of Assur groups for each number of links, but Assur graphs do not have links. Instead, we count the number of Assur graphs for each number of vertices (more precisely, the sum of the number of pinned and the number of unpinned vertices).

We hope that our results will stimulate research in graph rigidity and that this article with its inventory will also make the theory of Assur graphs more accessible to readers who are not experts in rigidity theory.

The article starts with this introduction in which we have briefly surveyed the applications in engineering of graph rigidity with a particular focus on Assur graphs. In Section 2 we introduce notation and concepts that we will need, also giving the state of the art from a mathematical point of view. In Section 3 we establish a relationship between Baranov trusses and $(2,3)$-tight graphs, as well as between Assur groups and Assur graphs. After that, in Section 4, we generate and count the number of rigidity circuits on up to 10 vertices, and we also generate and count the number of Assur graphs on up to 11 vertices. We then end the article with a short discussion.

2 Preliminaries

2.1 Graphs and frameworks

Graphs are the natural choice for mathematical modelling of networks and relationships. An *(undirected) graph* is a set of vertices together with a set of edges joining some of the pairs of the vertices. When used as models, the vertices typically model the elements of some set, and the edges model the connections or the relationships between these elements.

Graphs are often represented using their *adjacency matrix*, which is a matrix defined by the relationship between the vertices. In this matrix the rows and columns are both indexed by the vertices of the graph, so it is always a square matrix. The entry (u, v) in this matrix is 1 if there is an edge between vertex u and vertex v, otherwise it is zero. The *incidence matrix* of a graph describes the relationship between the vertices and the edges of the graph. The entry (v, e) in this matrix is 1 if the vertex v is on the edge e. In the signed incidence matrix, first a preferred orientation is chosen for each edge. Then the entry (v, e) is 1 if v is the beginning of e, -1 if v is the end of e and 0 otherwise. The adjacency and the incidence matrices are not uniquely defined, but depend on the indexing of the vertices and the edges.

For example, the adjacency matrix, the incidence matrix, and the signed incidence matrix of the graph $G = (V, E)$ defined by the vertex set

$$V = \{v_1, v_2, v_3, v_4\}$$

and the edge set

$$E = \{(v_1, v_2), (v_2, v_3), (v_3, v_4), (v_4, v_1)\}$$

are (indexing in order of appearance and orienting the edges as listed):

$$\begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix} \quad \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \end{pmatrix} \quad \begin{pmatrix} 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 \\ -1 & 0 & 0 & 1 \end{pmatrix}.$$

Adjacency matrix Incidence matrix Signed incidence matrix

Graphs can be represented in Euclidean space of dimension d by assigning a point configuration (a set of point coordinates) to the set of vertices and then linking the vertices that are connected by an edge via straight line segments. Such realizations of graphs are called *frameworks*. Typically, it is allowed for two vertices that do not share an edge to be assigned the same location. It is also allowed for two edges to cross. Figure 1a and Figure 1b show two embeddings of the graph G in the Euclidean plane as frameworks. Frameworks are used to model, for example, bar and joint structures, strut and cable structures, and sensor networks.

2.2 Rigidity of graphs

From an engineering point of view, a framework of a graph is *rigid* if the bar and joint structure it models maintains its shape and composition even under applied force; it will not bend, flex, or change shape when pressure is applied to any of its joints/vertices.

For example, a bar and joint structure modelled by the framework in Figure 1b can be obtained from the one modelled by the framework in Figure 1a by applying pressure to the top vertices, while preserving the length of the bars. Note that the distances between all

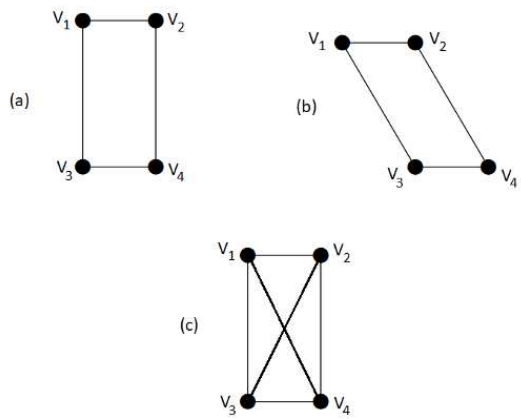


Figure 1: In (a) and (b): two frameworks of the same non-rigid graph. In (c): a framework of a rigid graph on the same vertex set.

vertices that are joined by edges are maintained, but the distances between the other vertices have changed. These frameworks are therefore by definition not rigid.

Figure 1c shows a framework that can be obtained from the framework in Figure 1a by adding the two diagonal edges. It is not possible to change the shape of this framework without changing the lengths of the edges, so it will stay the same, should pressure be applied. We see that the framework, and perhaps also the graph, should satisfy some criteria of rigidity. Indeed, as we will see, this graph is rigid. More precisely, it is globally, continuously, and infinitesimally rigid, but not minimally rigid.

Call the continuous rigid motions of Euclidean space the *trivial motions*. The continuous trivial motions of the plane are then the translations, the rotations, and the compositions of these motions. Note that the trivial motions do not change the distances between the vertices of the framework of a graph. This is indeed one reason for why we call them trivial motions within this context.

Now consider continuous motions of the Euclidean space that are not rigid/trivial; the motions that move each vertex of a framework independently. A framework of a graph is globally rigid if all frameworks with the same set of edge lengths have the same distances also between non-adjacent vertices. A framework can be globally flexible without there being a continuous motion of the framework taking it to another framework with the same set of edge lengths.

A framework of a graph is *(continuously) rigid* if every continuous non-trivial motion of the vertices that preserves the lengths of all edges also preserves the distances between all the vertices. Describe such a continuous motion of the vertices by a continuous function $P(t) = (P_1(t), \ldots, P_n(t))$ that returns a point configuration for the vertex set for each input time t. Let $K_{(i,j)}(t) = \|P_i(t) - P_j(t)\|^2$ denote the square of the length of the edge (v_i, v_j). Then $K_{(i,j)}(t)$ is constant, because the edge lengths are preserved by $P(t)$. Therefore the

derivative of $K_{(i,j)}(t)$ is $2(P_i(t) - P_j(t)) \cdot (P_i'(t) - P_j'(t)) = 0$. The set of equations obtained in this way from the set of edges of the graph, is, as we will see now, used to linearize the problem of finding the continuous motions of the graph.

An *infinitesimal motion* of the framework of a graph (V, E) with point configuration $P = (P_1, \ldots, P_n)$ is an assignment of vectors $M = (\overrightarrow{m}_1, \ldots, \overrightarrow{m}_n)$ to the points given by P to the vertices in V, satisfying

$$(P_i - P_j) \cdot (\overrightarrow{m}_i - \overrightarrow{m}_j) = (P_i - P_j) \cdot \overrightarrow{m}_i + (P_j - P_i) \cdot \overrightarrow{m}_j = 0$$

for every edge $(v_i, v_j) \in E$. For planar frameworks, this defines a system of linear equations with $2n$ unknowns, the two coordinates (a_i, b_i) of the vector $\overrightarrow{m}_i$ at each point P_i. The coefficients of the equation defined by the edge (v_i, v_j) are $P_i - P_j$ at the position (a_i, b_i) and $P_j - P_i$ at (a_j, b_j), and otherwise 0. The matrix containing the coefficients of this system is called the *rigidity matrix* of the framework. The subspace of solutions always contains the set of vectors corresponding to the derivatives of the continuous rigid planar motions of the framework (the translations, the rotations, and the compositions of these). This is a subspace of dimension three.

If there are no other solutions, then we say that the framework is *infinitesimally rigid*. A graph is infinitesimally rigid if a framework of the graph with a generic point configuration is infinitesimally rigid, where generic means that the dimension of the space of infinitesimal flexes at this point configuration is minimal, that is, the rank of the rigidity matrix is maximal. (Many authors take generic to mean that the coordinates of the point configuration are algebraically independent, since this implies that the rank of the rigidity matrix will be maximal. However, we think that since this is not a necessary condition; see, for example, Figure 4, it is better to choose the definition of generic in a way that actually captures the desired property. Some authors also call such frameworks regular.) Note therefore that when we talk about infinitesimally rigid graphs, we are not discussing the rigidity of all frameworks of the graph, but only of the generic ones. The frameworks that are not generic are called singular.

Global rigidity clearly implies (continuous) rigidity because there cannot be any continuous non-trivial motion without the distances between a pair of vertices changing. All infinitesimally rigid graphs are continuously rigid, by a result due to Asimow and Roth [1]. However not all rigid frameworks are infinitesimally rigid. For example, the complete graph on three vertices is globally rigid, hence (continuously) rigid in all frameworks. But if the vertices of the framework are on a single line, then there is an admissible infinitesimal motion of the vertex in the middle.

2.3 The rigidity matroid

A *matroid* is a combinatorial structure that generalizes the notion of linear independence from linear algebra. A matroid is a set U together with a set I of subsets of U, satisfying

1. The empty set is in I;

2. Every subset of a set of I is also in I;

3. If A and B are sets in I, and A has more elements than B, then there is an element $x \in A \setminus B$ such that the set $B \cup \{x\}$ is in I.

The set U is called the *base set* and the set I is the collection of *independent sets.*

Every matrix defines a matroid. Let U be the set of row vectors of a matrix and let I be the collection of linearly independent subsets of U. Then (U, I) is a matroid.

Every graph defines a matroid. Let U be the set of edges of the graph and let I be the collection of subsets of U that do not contain a cycle. Then (U, I) is a matroid. We will call this matroid the cycle matroid of the graph.

There are other ways to define matroids from graphs. In this article we are interested in the rigidity matroid of dimension two of a graph. The *rigidity matroid* of dimension d of a graph is defined by taking the base set U to be the set of edges and I the collection of subsets of edges corresponding to linearly independent sets of rows of the rigidity matrix. Note that the rigidity matrix of a graph in dimension one is essentially the signed incidence matrix after multiplying the row indexed by the edge e with the (possibly unknown) length of the edge e. The rigidity matroid of dimension one is therefore essentially the same as the cycle matroid of the graph. So, the rigidity matroid generalizes the concept of cycle matroid.

The rigidity matrix in dimension d has d times the number of columns of the incidence matrix. For example, Figure 1c shows a framework of the complete graph G on four vertices. By assigning the point configuration $P_0 = ((x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4))$ to the vertex set $V = \{v_1, v_2, v_3, v_4\}$, we can consider all two-dimensional frameworks of G. The rigidity matrix of G in two dimensions, indexing the vertices and edges as they appear in the definition above, is

$$M_2(G) = \begin{pmatrix} x_1 - x_2 & y_1 - y_2 & x_2 - x_1 & y_2 - y_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & x_2 - x_3 & y_2 - y_3 & x_3 - x_2 & y_3 - y_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & x_3 - x_4 & y_3 - y_4 & x_4 - x_3 & y_4 - y_3 \\ x_1 - x_4 & y_1 - y_4 & 0 & 0 & 0 & 0 & x_4 - x_1 & y_4 - y_1 \\ x_1 - x_3 & y_1 - y_3 & 0 & 0 & x_3 - x_1 & y_3 - y_1 & 0 & 0 \\ 0 & 0 & x_2 - x_4 & y_2 - y_4 & 0 & 0 & x_4 - x_2 & y_4 - y_2 \end{pmatrix}.$$

The rigidity matroid of dimension two of G is the matroid defined on the edge set with the collection of independent sets being the sets of edges that correspond to the independent rows of the matrix $M_2(G)$. The maximal independent sets (the bases of the matroid) are all the subsets of five edges. It is a fundamental property of matroids that all maximal independent sets have the same cardinality. This cardinality is the rank of the matroid. For example, the rigidity matroid of our graph G has rank five.

The rigidity matroid of a graph H on $|V|$ vertices is the induced submatroid of the rigidity matroid of the complete graph on $|V|$ vertices, obtained by removing the edges that are not in H together with all independent sets containing these edges. In terms of the

rigidity matrix, this corresponds to considering only the rows corresponding to the edges in $H = (V, E)$.

In two dimensions, the maximal rank of the rigidity matroid of a graph on $|V|$ vertices is $2|V|-3$, because the number of columns is $2|V|$ and the kernel corresponds to the infinitesimal motions of the trivial (rigid) motions, which is a subspace of dimension three. A graph that has a rigidity matroid of rank $2|V| - 3$ is rigid, because all the infinitesimal motions are trivial.

It is known how to characterize the independent sets of the rigidity matroid of dimension one and two in generic point configurations in terms of the combinatorial properties of the graph only. Such a characterization is not known in higher dimensions. In particular, this is an open problem for the important case of dimension three [12].

2.4 Tight graphs

A graph is $(2, 3)$-*sparse* if and only if every induced subgraph on $k \geq 2$ vertices has at most $2k - 3$ edges, and $(2, 3)$-*tight* if the graph itself also has exactly $2|V| - 3$ edges [5]. It can be seen that this happens exactly when the graph is a maximal independent set (a basis) in the rigidity matroid on $|V|$ vertices [5]. A $(2, 3)$-tight graph or *Laman graph* is a minimally rigid graph, in the sense that if any edge is removed, then the graph will no longer be rigid.

Henneberg proved that all $(2, 3)$-tight graphs can be constructed from the graph on two vertices with one edge in between them, using only two operations [8] (sometimes called the Henneberg operations):

1. *Vertex-addition.* Given a graph $G = (V, E)$, add a new vertex v and two new edges (v, u), (v, w), where $u, w \in V$.

2. *Edge-split.* Given a graph $G = (V, E)$, remove one edge $(u, v) \in E$ and instead add one new vertex w and three edges (u, w), (v, w) and (x, w), where $x \in V$ is a vertex different from u and v.

The $(2, 3)$-tight graphs were first introduced by Hilda Pollaczek-Geiringer in 1929 [18], but her results were apparently unknown to the community when Laman introduced them again in 1970 [15]. In most of the literature they are known as Laman graphs. The number of $(2, 3)$-tight graphs is sequence A227117 in the OEIS [22]. Until recently the sequence ended with $|V| = 8$, but the last value was incorrect. The sequence was corrected and extended by Capco et al. [3] up to $|V| = 12$. We have verified their enumeration up to $|V| = 10$.

2.5 Rigidity circuits

A circuit in a matroid is a subset C of the base set U such that

- C is not an independent set,

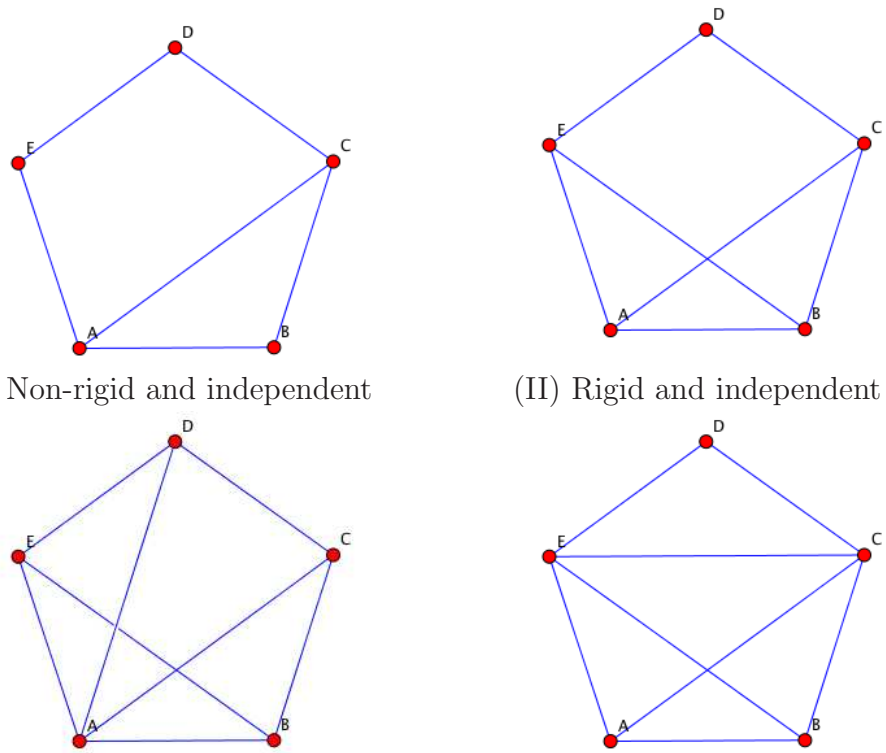
- the removal of any element from C gives an independent set.

$\|V\|$	2	3	4	5	6	7	8	9	10	11	12
#	1	1	1	3	13	70	608	7222	110132	39273	44176717

Figure 2: The number of $(2,3)$-tight graphs on $|V|$ vertices [3] is sequence A227117 in the OEIS [22].

Applying this definition to the rigidity matroid, a *rigidity circuit* is a subset of the edge set of a graph H that is not independent, but the removal of any edge gives an independent set. Note that if an edge is added to a maximal independent set, then the result is not necessarily a rigidity circuit, but it always contains exactly one rigidity circuit. Depending on where the edge is added this rigidity circuit is the entire graph or a smaller graph.

For example, consider the four graphs in Figure 3.



(I) Non-rigid and independent (II) Rigid and independent

(III) Rigid and dependent (IV) Rigid and dependent

Figure 3: Examples of rigidity and independence for graphs.

Graph I is not rigid in the plane, because the position of vertex D and E can vary without varying the position of the other three vertices, nor varying the length of the edges. Graph I is also independent in the plane, because the rank of its rigidity matrix equals the number

of edges. With the purpose of making this example explicit, we present the rigidity matrix of this particular framework, with vertices equally distributed on the unit circle, in Figure 4. The rank of this matrix is six, which is maximal, even though the coordinates of the point configuration are algebraically dependent.

Graph II is obtained from graph I by adding the edge (B, E). Graph II is rigid and independent in the plane, therefore graph II is minimally rigid in the plane: a $(2, 3)$-tight graph.

Graph III is obtained from graph II by adding the edge (A, D). Graph III is rigid and dependent in the plane. By removing any edge from graph III we obtain an independent graph, so graph III is a rigidity circuit; it is minimally dependent in the plane.

Graph IV is obtained from graph II by adding the edge (C, E). Graph IV is also rigid and dependent in the plane, but it is not minimally dependent. Removing the edge (C, D) or (D, E) leaves a graph that is still dependent. Graph IV is therefore not a rigidity circuit, but it contains a unique rigidity circuit: the induced subgraph on the vertices A, B, C and E. This is the complete graph on four vertices, which we denote by K_4.

Circuits of the planar rigidity matroid are rigid, but this is not necessarily true in higher dimensions. Berg and Jordán [2] proved that all planar rigidity circuits can be generated from K_4 using only two operations:

1. *Edge-split.* See Section 2.4.

2. *Two-sum.* Given two graphs $G = (V, E)$ and $G' = (V', E')$, choose two adjacent vertices in each graph, $u, v \in V$ and $u', v' \in V'$ such that $(u, v) \in E$ and $(u', v') \in E'$. Identify $u = u'$ and $v = v'$ and remove the (now multiple) edge $(u, v) = (u', v')$.

2.6 Assur graphs

A *pinned framework* is a framework in which some of the vertices are pinned, in the sense that their position is locked. An *unpinned framework* is a graph $G = (V, E)$ of vertices and edges together with a point configuration that makes a point correspond to each vertex. A pinned framework is a pinned graph $G = (I, P, E)$ of inner vertices, pinned vertices and edges, together with a point configuration that makes a point correspond to each inner vertex and to each pinned vertex. For example, a pinned framework makes a natural mathematical model of the Meccano structure of bars and joints in Figure 5, given that the positions of the three loose ends are fixed/pinned in the plane.

The concepts of rigidity defined above can be extended to pinned graphs and pinned frameworks. A pinned graph $G = (I, P, E)$ is called *pinned isostatic* if $E = 2|I|$ and the graph $\tilde{G} = G \cup K$ is infinitesimally rigid as an unpinned graph, where K is the complete graph on the pinned vertex set P. A pinned framework in $\mathbb{R}^d$ is a function assigning a point in $\mathbb{R}^d$ to each vertex in $P \cup I$. A pinned framework of the pinned graph G is rigid if the rigidity matrix of $\tilde{G}$ with the columns corresponding to the vertices in P removed has rank $2|I|$, it is independent if the rows corresponding to E are independent, and isostatic if it is rigid and independent.

Assur graphs are minimally rigid pinned graphs that do not contain smaller minimally rigid pinned graphs. Compare this with the notion of a $(2,3)$-tight graph; a graph that is minimally rigid in the plane, and may contain minimally rigid proper subgraphs. Assur graphs were characterized by Servatius et al. [20] as follows:

Theorem 1. *Let $G = (I, P, E)$ be a pinned isostatic graph. Then G is an Assur graph if any of the following equivalent conditions are satisfied:*

1. *All proper subsets of vertices $I' \cup P' \subseteq I \cup P$ with $|I'| > 0$ induce a pinned subgraph $G' = (I', P', E')$ with $|E'| \leq 2|I'| - 1$.*

2. *If the set of pinned vertices P is contracted to a single vertex p^*, then the resulting induced unpinned graph is a rigidity circuit.*

3. *Either the graph has a single inner vertex of degree 2, or each time we delete a vertex, the resulting pinned graph has a motion of all inner vertices (in generic position).*

4. *Deletion of any edge from G results in a pinned graph that has a motion of all inner vertices (in generic position).*

Servatius et al. [20] proved that all Assur graphs can be constructed from the unique Assur graph on three vertices and the unique Assur graph on five vertices, using only the following three operations:

1. *Edge-split.* See Section 2.4.

2. *Two-sum.* See Section 2.5.

3. *Pin-rearrangement.* Two pinned graphs G and H can be obtained from each other using pin-rearrangement if G can be obtained from H by first identifying all pinned vertices, obtaining a rigidity circuit R, and then choosing one vertex of R and splitting it into at least two pinned vertices.

The two smallest Assur graphs cannot be obtained from other Assur graphs using the three operations above. Frameworks of these two Assur graphs are given in Figure 6.

3 Baranov trusses, (2,3)-tight graphs (Laman graphs), Assur groups and Assur graphs and how they are related

Because this is an article with a combinatorial flavor, we will here define planar linkages in terms of their structural scheme. For a more thorough background on these concepts from the viewpoint of mechanical kinematics, see, for example, [16].

For us then, a *planar linkage* is a collection of rigid and convex polygonal shapes in the Euclidean plane, some connected pairwise on the vertices. We allow *digons*, that is, polygons with two vertices, but we do not regard a single vertex to be a polygonal shape. In the mathematical rigidity literature, such planar linkages are known as *planar body-pin frameworks*. The rigid and convex polygonal shapes correspond to rigid bodies and the vertices are called pins. The bodies can take any shape, but most important results assume that the pins are in generic position. Exactly two bodies meet at each pin, so the proofs of results about body-pin frameworks often rely on modeling it (dually) as a graph with a vertex for each body and an edge for each pin. In this article we will, however, use the terms *link* and *vertex*, to avoid confusion between the unrelated terms "pin of a body-pin framework" and the "pinned vertices" used in the definition of Assur graph.

It is possible to restrict the definition to linkages with non-overlapping links (except at the vertices, of course), but we will not do this here. The linkage is called closed when at least two links meet in every vertex, otherwise open. Assur groups are examples of open linkages, while Baranov trusses are examples of closed linkages.

We say that a linkage is rigid if a change in the distance between any pair of vertices would imply a change in the distance between some pair of vertices that are in the same link. Within this context, we define a planar Baranov truss to be a minimally rigid planar linkage, that is, a linkage that is rigid, but contains no proper rigid sublinkage. The number of links n and the number of joints j in a Baranov truss is related by what is known as Grübler's equation: $2j = 3(n-1)$. Grübler's equation has the following consequence, which we did not find described anywhere in the literature.

Lemma 2. *A Baranov truss contains at least three links with exactly two joints.*

Proof. If s links have exactly two joints, then counting the number of joints per link twice gives $2j \geq 3(n-s) + 2s = 3n - s$. Together with Grübler's equation, this implies that $s \geq 3$ in a Baranov truss. □

Define a maximal proper $(2,3)$-tight subgraph H of a graph G to be a subgraph of G such that it is $(2,3)$-tight, and not properly contained within any other $(2,3)$-tight subgraph, except possibly for G.

Theorem 3. *Baranov trusses and $(2,3)$-tight graphs are related in the following manner.*

1. *Every Baranov truss can be constructed from a $(2,3)$-tight graph by finding all maximal proper $(2,3)$-tight subgraphs and replacing them with links. The number of vertices of the link created from a $(2,3)$-tight subgraph H is the number of maximal proper $(2,3)$-tight subgraphs that share a vertex with H (hence not necessarily the number of vertices of H).*

2. *Every $(2,3)$-tight graph can be constructed from a Baranov truss, by replacing any link on n joints with a $(2,3)$-tight graph on at least n vertices, assigning the n joints to n randomly chosen vertices of the $(2,3)$-tight graph.*

Proof. The proof is divided in two parts, corresponding to the two parts of the statement of the theorem.

1. Given a Baranov truss B on ℓ links, replace every link on n vertices with a $(2,3)$-tight graph on n vertices; a graph $G = (V, E)$ is obtained. Its edge set E can be partitioned into $E = \bigsqcup E_i$ (this means that $E_i \cap E_j = \emptyset$ for $i \neq j$, $i, j \in \{1, \ldots, \ell\}$) such that each part E_i consists of the edge set of one of the $(2,3)$-tight graphs that have replaced the links.

   Let F be a non-empty subset of E, covering a vertex set $W \subseteq V$, such that F contains edges from s different edge sets $E_{i_1}, \ldots, E_{i_s}$, with the associated partition of F into $F = F_{i_1} \sqcup \cdots \sqcup F_{i_s}$, each edge set F_i covering the vertex set W_i. For every i, $|F_i| \leq 2|W_i| - 3$, because $F_i \subseteq E_i$ and E_i is the edge set of a $(2,3)$-tight graph, so if $s = 1$, we are done.

   Now assume that $s > 1$ and that $s < \ell$, so that there are at least one E_i from which F contains no edge. If the graph (W, F) was rigid, then B must contain a rigid proper sublinkage (consisting of the links defining $E_{i_1}, \ldots, E_{i_n}$), but it does not, so (W, F) must be flexible. Assume now that F is not an independent set of edges. Then F contains a subset R of edges covering a set of vertices U that defines a rigidity circuit (a minimally dependent set of edges), and so the graph (U, R) is rigid in the plane. But since R is dependent it cannot be contained within any of the E_i, so the same argument we just used to prove that (W, F) is flexible in the case of $s > 1$ applies also to R. Therefore (U, R) must be flexible, a contradiction. So (W, F) must be independent and flexible, therefore $|F| < 2|W| - 3$.

   Now consider the case that $s = \ell$. From Lemma 2 we know that there are at least three links with exactly two joints, so we may assume that $s > 2$ and that E_s consists of a single edge. This implies that $F_s = E_s$ and that $(W, F) = (V, E) = G$. Consider the edge set $\bar{F} = F \setminus F_s$, covering the vertex set $\bar{W}$. Then $\bar{F}$ is an edge set with edges in $\ell - 1$ different edge sets $E_1, \ldots, E_{\ell-1}$. The previous arguments then show that $|\bar{F}| < 2|\bar{W}| - 3$. We have that $|F_s \cap \bar{F}| = 0$ and $W_s \subseteq \bar{W} = W = V$, so $|E| = |F| = |\bar{F} \cup F_s| = |\bar{F}| + |\bar{F}_s| = |\bar{F}| + 1 < 2|\bar{W}| - 2 = 2|V| - 2$, that is, $|E| \leq 2|V| - 3$. Finally, because B is rigid, (V, E) must be rigid, so that $|E| = 2|V| - 3$.

   We conclude that $G = (V, E)$ is rigid and independent in the plane, so G is a $(2,3)$-tight graph. The Baranov truss B may be constructed from G in the following way. Find all edge sets $F \subsetneq E$ that are maximal with respect to the property that $|F| = 2|W| - 3$, where W is the set of vertices covered by F. Construct a linkage on the joint set V and for every such edge set F construct a link as a convex rigid polygonal shape on its joint/vertex set W.

2. Let G be a $(2,3)$-tight graph. Find all maximal proper $(2,3)$-tight subgraphs H and replace them with a link L, such that a vertex v in H is a vertex in L if v is in at least two maximal proper $(2,3)$-tight subgraphs. The result is a linkage that is rigid

but has no rigid sublinkages, hence a Baranov truss, from which G can be constructed by reversing the construction, making sure to use the correct $(2,3)$-subgraph for each link.

□

Note that infinitely many $(2,3)$-tight graphs can be constructed from any Baranov truss, but only one Baranov truss can be constructed from each $(2,3)$-graph, using the construction described in Theorem 3.

Just as any Baranov truss can be constructed from a $(2,3)$-tight graph by finding all maximal proper induced $(2,3)$-tight subgraphs and replacing them with rigid polygonal "links", so can any Assur group be constructed from an Assur graph, in exactly the same way. Also, from any Assur group we may construct an Assur graph, by replacing any rigid polygonal "link" with n joints by a $(2,3)$-tight graph on at least n vertices. Hence we may state the relation between Assur groups and Assur graphs in the following explicit way.

Theorem 4. *Assur groups and Assur graphs are related in the following manner.*

1. *Every Assur group can be constructed from an Assur graph by finding all maximal proper $(2,3)$-tight subgraphs and replacing them with links. The number of vertices of the link created from a $(2,3)$-tight subgraph H is the number of maximal proper $(2,3)$-tight subgraphs that share a vertex with H (hence not necessarily the number of vertices of H).*

2. *Every Assur graph can be constructed from an Assur group by replacing any link on n vertices with a $(2,3)$-tight graph on at least n vertices.*

We see that an Assur group can be represented by infinitely many Assur graphs, by choosing $(2,3)$-tight graphs of different orders. For example, both the Assur graph on three vertices and the Assur graph on five vertices, correspond to the Assur group on two links. In the case of the Assur graph on three vertices, the two links correspond to the two edges of the graph. The Assur graph on five vertices contains a $(2,3)$-tight graph on four vertices (one of which is pinned) which can be replaced by a rigid polygonal link with only two joints. Therefore the $(2,3)$-tight graph on four vertices does not have to be replaced by a quadrilateral but can be replaced by a digon (a polygon with two vertices).

4 Counting rigidity circuits

Our objective is to construct an inventory of Assur graphs on at most N vertices for some suitable N. The first step in the method that we chose was to create an inventory of all rigidity circuits on at most $N' = N - 1$ vertices. We could not find such an inventory in the literature, nor could we find any enumeration of rigidity circuits. The following then, is the method we used to generate such an inventory.

The smallest rigidity circuit is K_4. As described by Berg and Jordán [2] (and briefly in Section 2.5), all rigidity circuits can be constructed inductively from K_4 using only two operations: edge-split and two-sum. Consider a rigidity circuit G on $n > 4$ vertices, created from K_4 using these two operations. If the last operation applied was an edge-split, then G was created from a rigidity circuit on $n-1$ vertices, because applying the edge-split operation to a graph will always add one vertex to the graph. If the last operation was a two-sum, then $G = (V, E)$ was created from two rigidity circuits $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, such that $n = |V| = |V_1| + |V_2| - 2$ and $|E| = |E_1| + |E_2| - 2$. This is because a two-sum identifies two pairs of vertices in the resulting graph, and the two edges the graphs were joined on are removed.

Consequently, to generate all rigidity circuits on n vertices, we applied the edge-split operation to all rigidity circuits on $n-1$ vertices, and the two-sum operation to all pairs of rigidity circuits of orders n_1 and n_2 satisfying $n_1 + n_2 - 2 = n$. When applying the edge-split to a given graph, a list of graphs was obtained, because the outcome of the operation depends on the choice of an edge. When applying the two-sum to two given graphs, a list of graphs was obtained, because the outcome depends on the choice of one edge from each graph, as well as their relative orientation. Some of the graphs obtained in this way are isomorphic. To avoid counting isomorphic copies of the same graph we applied a test for isomorphism. Using this method, we created an inventory of all rigidity circuits on up to 10 vertices.

With the purpose of avoiding erroneous calculations, we made two independent implementations, one in Python using the library NetworkX and another one in SageMath. The number of rigidity circuits on n vertices is given in Figure 7; these numbers are catalogued in the OEIS as sequence A350484 [22]. The inventory can be accessed in the data accompanying this paper at the following URL:

https://cs.uwaterloo.ca/journals/JIS/VOL25/Stokes/data.html .

5 Counting Assur graphs

As described by Servatius et al. [20], all Assur graphs except the Assur graph on three vertices can be constructed from the Assur graph on five vertices, using only three operations. However, the connection with the rigidity circuits given by the second part of Theorem 1 allows for another method of generating all Assur graphs. This method is also briefly described in Section 3.3 of Servatius et al. [20, Section 3.3].

Theorem 1 implies that every Assur graph can be constructed from a rigidity circuit by selecting a vertex and splitting it into several pinned vertices. When we split a vertex of a rigidity circuit in this way, the edges are preserved. In particular, the number of pinned vertices cannot be larger than the degree of the selected vertex. The number of pinned vertices can, however, be smaller, because a pinned vertex can have several edges. For an illustration, see Figure 9 that shows two Assur graphs obtained from the framework of the unique rigidity circuit on five vertices shown in Figure 8.

We generated all Assur graphs on up to 11 vertices from the set of rigidity circuits on up

to 10 vertices, by selecting each admissible vertex and then splitting it into the appropriate number of pinned vertices. In order to generate all possible Assur graphs with m pinned vertices from a given rigidity circuit $G = (V, E)$ and a given selected vertex $v \in V$, we created a list of all partitions with m parts of the edges incident with v. For a given partition, the corresponding Assur graph was created by letting each part in the partition correspond to a pinned vertex.

Because an Assur graph must have at least two pinned vertices, the rigidity circuit from which it is created must have at least one vertex less than the Assur graph. Of course, small rigidity circuits cannot in general be used to create large Assur graphs. However, if a relatively small rigidity circuit has a vertex with large degree, the number of pinned vertices in the resulting Assur graph can be large. But because the numbers we were dealing with were relatively small, it made no sense keeping track of these details. Therefore, we simply used the set of all rigidity circuits on at most $n - 1$ vertices to create the set of all Assur graphs on n vertices.

We made two independent implementations of this method, one in Python using NetworkX and another one in SageMath. The number of Assur graphs on N vertices is given in Figure 10; these numbers are catalogued in the OEIS as sequence A350485 [22]. The inventory can be accessed in the data accompanying this paper at the following URL:

https://cs.uwaterloo.ca/journals/JIS/VOL25/Stokes/data.html .

6 Conclusions

We have presented an inventory and the count of the number of rigidity circuits on up to 10 vertices, as well as an inventory and the count of the number of Assur graphs on up to 11 vertices. This is, to our knowledge, the first time these numbers are presented, and therefore also the first time such an inventory was created.

An inventory of Assur groups was presented by Huang and Ding [10], but we have thoroughly explained the relationship between the sets of $(2, 3)$-tight graphs and Baranov trusses on the one hand, and the Assur graphs and the Assur groups on the other. Hence it should be clear that our inventory is different from theirs.

The algorithm that we used is based on observations made by Servatius et al. [20]. It was implemented using the NetworkX library in Python and using SageMath, independently. With an input of a natural number N, our program outputs the set of rigidity circuits on $N - 1$ vertices together with the set of Assur graphs on N vertices, up to isomorphism. Of course, as N grows large, the complexity of the problem quickly grows, and computations become unfeasible.

We hope that the inventory and count of rigid graphs in this article prove to be a useful data set that can be used in further research. Having a complete set of small Assur graphs at hand may be useful when exploring the scope for applications of these structure in engineering, or for investigating previously overlooked mathematical properties of these graphs.

It would be useful to be able to expand this work to rigidity circuits and Assur graphs in three dimensions. However, it is still an open mathematical problem to determine exactly what graphs play the role of the $(2,3)$-tight graphs (Laman graphs) in three dimensions (we know it is not the $(3,6)$-tight graphs), and less is known for the Assur graphs.

7 Acknowledgments

The authors would like to thank Lars-Daniel Öhman and the anonymous referee for their comments as they greatly improved the exposition of the paper.

References

[1] L. Asimow and B. Roth, Rigidity of graphs II, *J. Math. Anal. Appl.* **68** (1979), 171–190.

[2] A. Berg and T. Jordán, A proof of Connelly's conjecture on 3-connected circuits of the rigidity matroid, *J. Combin. Theory Ser. B* **88** (2003), 77–97.

[3] J. Capco, M. Gallet, G. Grasegger, C. Koutschan, N. Lubbes, and J. Schicho, Computing the number of realizations of a Laman graph, *Electron. Notes Discrete Math.* **61** (2017), 207–213.

[4] J. Graver, Rigidity matroids, *SIAM J. Discrete Math.* **4** (1991), 355–368.

[5] J. Graver, H. Servatius, and B. Servatius, *Combinatorial Rigidity*, American Mathematical Society, 1993.

[6] A. Hagberg, P. Swart, and D. Schult, Overview of NetworkX—NetworkX 2.4 documentation, Networkx.github.io, 2020. Available at https://networkx.github.io/documentation/stable/.

[7] E. Hahn and O. Shai, The unique engineering properties of Assur groups/graphs, Assur kinematic chains, Baranov trusses and parallel robots, in *Proc. of the ASME 2016 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. Volume 5B: 40th Mechanisms and Robotics Conference*, ASME, 2016, Paper No. DETC2016-59135, V05BT07A074, pp. 1–9.

[8] L. Henneberg, *Die Graphische Statik der Starren Systeme*, B. G. Teubner, 1911.

[9] P. Huang and H. F. Ding, Structural synthesis of Baranov trusses with up to 13 links, *ASME J. Mech. Design* **141** (2019), 072301.

[10] P. Huang and H. F. Ding, Structural synthesis of Assur groups with up to 12 links and creation of their classified databases, *Mech. Mach. Theory* **145** (2020), 103668.

[11] B. Jackson, Notes on the rigidity of graphs, preprint, 2007. Available at https://www.researchgate.net/publication/228802533_Notes_on_the_Rigidity_of_Graphs.

[12] B. Jackson and T. Jordán, On the rank function of the 3-dimensional rigidity matroid, *Int. J. Comput. Geom. Appl.* **16** (2006), 415–429.

[13] B. Jackson and T. Jordán, The d-dimensional rigidity matroid of sparse graphs, *J. Combin. Theory Ser. B* **95** (2005), 118–133.

[14] D. Jacobs and B. Hendrickson, An algorithm for two-dimensional rigidity percolation: the pebble game, *J. Comput. Phys.* **137** (1997), 346–365.

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

[16] J. M. McCarthy and G. S. Soh, *Geometric Design of Linkages*, Springer, 2010.

[17] O. Orki, A. Ayali, O. Shai, and U. Ben-Hanan, Modelling of caterpillar crawl using novel tensegrity structures, *Bioinspir. Biomim.* **7** (2012), 046006.

[18] H. Pollaczek-Geiringer, Über die Gliederung ebener Fachwerke, *Z. Angew. Math. Mech.* **7** (1927), 58–72.

[19] N. Rojas and F. Thomas, On closed-form solutions to the position analysis of Baranov trusses, *Mech. Mach. Theory* **50** (2012), 179–196.

[20] B. Servatius, O. Shai, and W. Whiteley, Combinatorial characterization of the Assur graphs from engineering, *Eur. J. Comb.* **31** (2010), 1091–1104.

[21] O. Shai, I. Tehori, A. Bronfeld, M. Slavutin, and U. Ben-Hanan, Adjustable tensegrity robot based on Assur graph principle, in *Proc. of the ASME 2009 International Mechanical Engineering Congress and Exposition, Volume 10: Mechanical Systems and Control, Parts A and B*, ASME, 2009, pp. 257–261.

[22] N. J. A. Sloane et al., The On-Line Encyclopedia of Integer Sequences, *OEIS Foundation Inc.*, https://oeis.org.

[23] K. Waldron, G. Kinzel, and S. Agrawal, *Kinematics, Dynamics, and Design of Machinery*, Wiley, 2016.

2010 *Mathematics Subject Classification*: Primary 05C30; Secondary 05C62, 52C25, 70B15.
Keywords: rigidity circuit, Assur graph.

(Concerned with sequences A227117, A350484, and A350485.)

Received August 20 2020; revised versions received September 16 2020; December 31 2021; January 1 2022. Published in *Journal of Integer Sequences*, January 2 2022.

Return to Journal of Integer Sequences home page.

$$
\begin{pmatrix}
-\frac{1}{4}\sqrt{2\sqrt{5}+10} & -\frac{1}{4}\sqrt{5}+\frac{5}{4} & \frac{1}{4}\sqrt{2\sqrt{5}+10} & \frac{1}{4}\sqrt{5}-\frac{5}{4} & 0 & 0 & 0 & 0 & 0 & 0 \\
-\frac{1}{4}\sqrt{-2\sqrt{5}+10} & \frac{1}{4}\sqrt{5}+\frac{5}{4} & 0 & 0 & \frac{1}{4}\sqrt{-2\sqrt{5}+10} & -\frac{1}{4}\sqrt{5}-\frac{5}{4} & 0 & 0 & 0 & 0 \\
\frac{1}{4}\sqrt{2\sqrt{5}+10} & -\frac{1}{4}\sqrt{5}+\frac{5}{4} & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{4}\sqrt{2\sqrt{5}+10} & \frac{1}{4}\sqrt{5}-\frac{5}{4} \\
0 & 0 & \frac{1}{4}\sqrt{2\sqrt{5}+10}-\frac{1}{4}\sqrt{-2\sqrt{5}+10} & \frac{1}{2}\sqrt{5} & -\frac{1}{4}\sqrt{2\sqrt{5}+10}+\frac{1}{4}\sqrt{-2\sqrt{5}+10} & -\frac{1}{2}\sqrt{5} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \frac{1}{2}\sqrt{-2\sqrt{5}+10} & 0 & -\frac{1}{2}\sqrt{-2\sqrt{5}+10} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{4}\sqrt{2\sqrt{5}+10}-\frac{1}{4}\sqrt{-2\sqrt{5}+10} & -\frac{1}{2}\sqrt{5} & -\frac{1}{4}\sqrt{2\sqrt{5}+10}+\frac{1}{4}\sqrt{-2\sqrt{5}+10} & \frac{1}{2}\sqrt{5}
\end{pmatrix}.
$$

Figure 4: The rigidity matrix of the framework in Figure 3 (I).

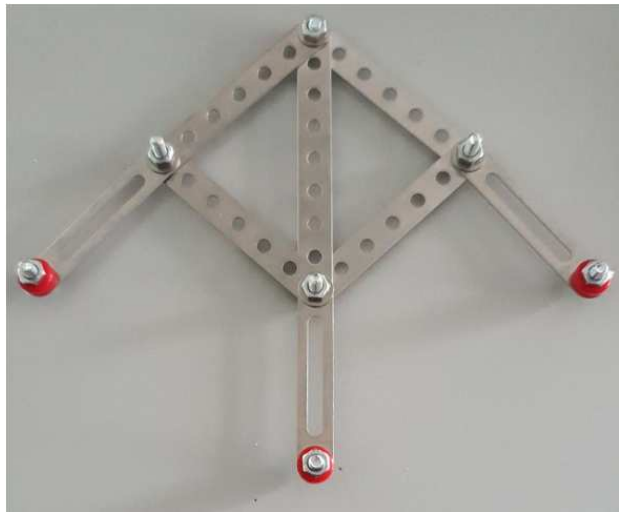


Figure 5: A Meccano realization of an Assur graph on seven vertices.

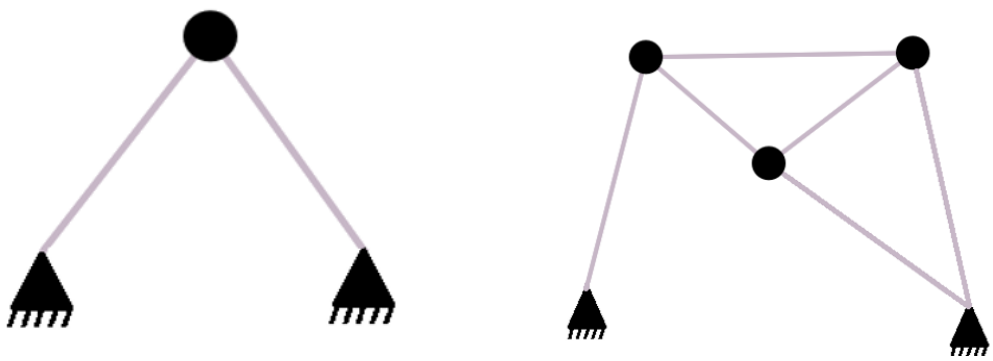


Figure 6: The unique Assur graph on three vertices and the unique Assur graph on five vertices.

$\|V\|$	4	5	6	7	8	9	10
#	1	1	4	15	109	1075	14506

Figure 7: The number of rigidity circuits on $|V|$ vertices, sequence A350484 in the OEIS [22].

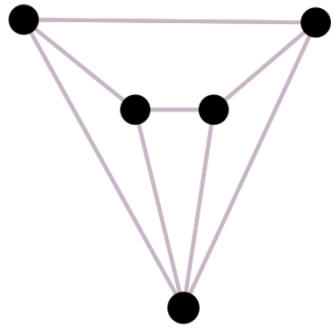


Figure 8: A framework of the unique rigidity circuit on five vertices.

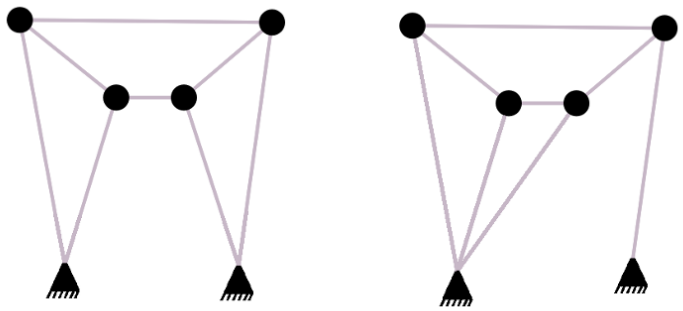


Figure 9: Two frameworks of two combinatorially non-isomorphic Assur graphs obtained from the rigidity circuit in Figure 8.

$\lvert V \rvert$	3	4	5	6	7	8	9	10	11
#	1	0	1	6	32	294	3433	50168	860308

Figure 10: The number of Assur graphs on $|V|$ vertices, sequence A350485 in the OEIS [22].