



HAL
open science

De la rigidité structurelle

Christophe Jermann, Bertrand Neveu, Gilles Trombettoni

► **To cite this version:**

Christophe Jermann, Bertrand Neveu, Gilles Trombettoni. De la rigidité structurelle. 2002. hal-01408614

HAL Id: hal-01408614

<https://hal.science/hal-01408614v1>

Preprint submitted on 16 Feb 2017

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

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

De la rigidité structurelle

Christophe Jermann¹ - Bertrand Neveu - Gilles Trombettoni

INRIA Sophia Antipolis, 2004 route des lucioles, BP 93, 06902 Sophia Antipolis Cedex

{Christophe.Jermann, Bertrand.Neveu, Gilles.Trombettoni}@sophia.inria.fr

Résumé : *La rigidité structurelle, formulation structurelle issue du théorème de Laman [Lam70] sur la rigidité de systèmes à barres en 2D, est généralement considérée comme une bonne heuristique pour la détection de sous-parties rigides d'un système géométrique. La principale différence entre rigidité et rigidité structurelle vient du fait que cette dernière ne tient pas compte de la géométrie. Nous proposons une nouvelle définition de la rigidité structurelle qui introduit la géométrie.*

A partir de l'algorithme Dense proposé par Hoffmann et al [HLS97], nous proposons un nouvel algorithme correspondant à notre définition et discutons de son applicabilité à des problèmes réels. Mots-clés : Contraintes géométriques, Rigidité, Rigidité structurelle

1 Introduction

La détection de sous-systèmes rigides est une étape clé dans de nombreux algorithmes de résolution de systèmes géométriques [Kra92, BFH⁺95, LM97, DMS97, HLS00, JTNR00]. En particulier, les méthodes de résolution par *rigidification récursive* ont pour principe de décomposer un système géométrique en sous-systèmes rigides résolus séparément puis assemblés.

Les techniques utilisées jusqu'ici pour détecter des parties rigides se classent en deux catégories :

- Les approches basées sur la reconnaissance de modèles rigides pré-définis [Kra92, BFH⁺95, FH97].

- Les approches basées sur un mécanisme de flots ou de couplage maximum [LM97, HLS97].

La seconde catégorie, plus récente et plus générale, se base sur une définition structurelle de la rigidité, appelée *rigidité structurelle*.

Cette définition a été dérivée d'un théorème caractérisant les systèmes à barres rigides en 2D [Lam70]. Certaines différences bien connues existent entre la rigidité et la rigidité structurelle [Hen92, LM97]. Hoffmann et al [Sit00] ont introduit une heuristique afin de faire correspondre au mieux ces deux définitions.

Cet article remet en cause la définition de rigidité structurelle et l'heuristique proposée par Hoffmann et al (section 2). Nous proposons une nouvelle définition de la rigidité structurelle plus conforme à la rigidité (section 4). L'algorithme Dense proposé par Hoffmann et al [HLS97] pour détecter des sous-systèmes structurellement rigide est présenté et discuté (section 5) avant d'être mis à jour pour prendre en compte la nouvelle définition (section 6).

2 Définitions

Cette section présente la notion de la rigidité structurelle. Cette notion est une heuristique pour la détection de rigidité, comme nous le montrons sur des exemples. Elle présente aussi une amélioration de cette définition, proposée par Hoffmann et al [Sit00], et nous montrons aussi sur des exemples que cette modification ne suffit pas à mettre en correspondance rigidité structurelle et rigidité.

Definition 1 Système géométrique

Un système géométrique $S = (O, C)$ dans un espace de dimension d se définit par l'ensemble O

¹Bourse de Thèse BDI en association avec le CNRS et la région Provence Alpes Côte d'Azur

de ses objets géométriques, et l'ensemble C des contraintes géométriques portant sur des objets de O .

$S' = (O', C')$ est un sous-système de S (noté $S' \subset S$) s'il vérifie $O' \subset O$, $C' = \{c \in C \mid c \text{ ne porte que sur des objets de } O'\}$ et $|C'| > 0$ (donc $|O'| > 1$).

Résoudre un système géométrique consiste à fournir une position, une orientation et des dimensions à chaque objet géométrique de O de telle sorte que toutes les contraintes géométriques de C sont satisfaites.

Une définition informelle de la rigidité suffira dans cet article pour la comparer à la rigidité structurelle : Un système géométrique est *rigide* si ses solutions sont indéformables et tout déplacement de l'une de ses solutions dans l'espace considéré et encore une solution de ce système. On dira alors d'un système géométrique qu'il est *sous-rigide* si l'une de ses solutions est déformable, et qu'il est *sur-rigide* si l'une de ses solutions n'est pas déplaçable dans tout l'espace.

La rigidité structurelle est une propriété de comptage des *degrés de liberté* du système. Intuitivement, un degré de liberté représente une possibilité de mouvement continu indépendant (rotationnel ou translationnel). Plus formellement :

Definition 2 Degrés de liberté

Le nombre de degrés de liberté $dof(o)$ d'un objet géométrique o est égal au nombre de paramètres de l'objet qu'il faut fixer pour déterminer sa position, son orientation et ses dimensions.

Le nombre de degré de liberté $dof(c)$ retiré par une contrainte géométrique c est égal au nombre de paramètres que cette contrainte permet de déterminer².

Le nombre de degrés de liberté $dof(S)$ d'un système géométrique $S = (O, C)$ est la différence entre le nombre de degrés de liberté de offerts par ses objets et le nombre de degrés de liberté retirés par ses contraintes : $dof(S) = \sum_{o \in O} dof(o) - \sum_{c \in C} dof(c)$.

Par exemple, un point en 3D (resp. 2D) possède 3 (resp. 2) degrés de liberté puisque fixer un point consiste à déterminer ses 3 (resp. 2) coordonnées. Une contrainte de distance entre 2 points en dimension quelconque retire 1 degré de liberté : elle s'exprime par 1 équation algébrique de la forme $(x - x')^2 + (y - y')^2 + \dots = d^2$.

G. Laman a proposé le théorème suivant qui caractérise les systèmes à barres (constitués uniquement de points et contraintes de distances) rigides en 2D :

Théorème 1 [Lam70]

Un système géométrique à barre $S = (O, C)$ en 2D est rigide ssi $dof(S) = 3$, et $\forall S' = (O', C') \subset S$, $dof(S') \geq 3$.

La définition de *rigidité structurelle* applicable à tout système en dimension d est une généralisation issue du théorème précédent :

Definition 3 Rigidité Structurelle

Soit un système géométrique $S = (O, C)$ dans un espace de dimension d . On dit que S est structurellement rigide (noté s_rig1) ssi $dof(S) = \frac{d(d+1)}{2}$ et $\forall S' \subset S$, $dof(S') \geq \frac{d(d+1)}{2}$.

3 Différence entre rigidité structurelle et rigidité

La rigidité structurelle ne correspond à la rigidité que pour les systèmes à barres en 2D. Dans le cas général, elle est considérée comme une bonne heuristique pour l'identification de systèmes rigides ou sur-rigides [Hen92, LM97, HLS97] car :

²En pratique, on compte un degré de liberté par équation modélisant une contrainte donné. Ce principe est issu de l'hypothèse selon laquelle une équation permet de déterminer l'une de ses inconnues sur les réels. Cette hypothèse peut être fautive : l'équation $x^2 + y^2 = 0$ fixe à elle seule 2 degrés de liberté.

- Un système dégénéré peut être s_rig1 sans être rigide. (cf. le contre-exemple en 3D de Laman [Lam70]).
- Un système redondant peut être rigide sans être s_rig1 .

Cependant, même dans le cas générique est non redondant, la rigidité structurelle ne correspond pas toujours à la rigidité. Nous illustrons ce fait sur l'exemple suivant :

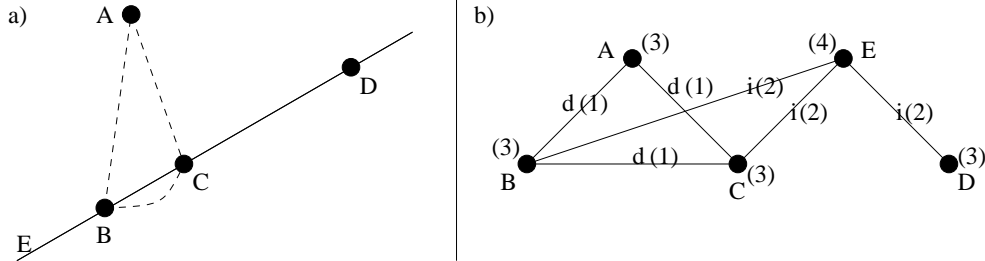


FIG. 1 – a) Système 3D constitué de 4 points, A, B, C et D, et une droite E. b) Graphe objets-contraintes représentant ce système géométrique.

Le système géométrique 3D de la figure 1(a) est constitué de 4 points, A, B, C, D possédant chacun 3 degrés de liberté et d'une droite E possédant 4 degrés de liberté. Les paires de points AB, AC et BC sont reliés par des contraintes de distance retirant 1 degré de liberté chacune. Les points B, C et D sont reliés à la droite E par des contraintes d'incidence retirant 2 degrés de liberté chacune. Le sous-système constitué du triangle ABC basé sur la droite E est rigide. Le système est cependant globalement sous-rigide puisque le point D peut bouger le long de la droite E indépendamment des autres points.

Selon la définition 3, le sous-système ABC n'est pas s_rig1 . En effet, ABC contient le segment AB qui a 5 degrés de liberté seulement. Comme $\frac{d(d+1)}{2} = 6$ en 3D, ce sous-système viole la partie de la définition 3 disant que tout sous-système doit avoir au moins 6 degrés de liberté.

Afin d'éviter ce type de problème, la notion suivante est introduite dans la définition de la rigidité structurelle [Sit00] :

Definition 4 Sous-système trivial

Soit un système géométrique $S = (O, C)$ dans un espace de dimension d . Un sous-système $S' = (O', C') \subset S$ est dit trivial s'il vérifie $|O'| \leq d$.

La rigidité structurelle est alors re-définie comme suit :

Definition 5 Rigidité Structurelle 2

Soit un système géométrique **non trivial** $S = (O, C)$ dans un espace de dimension d . On dit que S est s_rig2 ssi $dof(S) = \frac{d(d+1)}{2}$ et tout sous-système **non trivial** $S' \subset S$ vérifie $dof(S') \geq \frac{d(d+1)}{2}$.

Cette définition ne correspond toujours pas à la rigidité, même dans les cas génériques et non-redondants : le triangle ABC (cf figure 1(a)) est rigide mais trivial et n'est donc pas s_rig2 . Qui plus est, le sous-système BCDE est sous-rigide mais s_rig2 car il a bien 6 degrés de liberté et n'a aucun sous-système non trivial.

4 Système rigidifiable

Les deux définitions précédentes de la rigidité structurelle ne tiennent pas compte de la géométrie : en effet, pour être plus conforme à la rigidité, la définition de rigidité structurelle ne

devrait être appliquée qu'aux systèmes candidats à être rigides et s_rig1 , systèmes que nous appelons *rigidifiables*.

Definition 6 Système rigidifiable

Un système géométrique est dit *rigidifiable* si son ensemble d'objet peut être remplacé par un repère local.

Nous proposons la définition de rigidité structurelle suivante :

Definition 7 Rigidité Structurelle 3

Soit un système géométrique **rigidifiable** $S = (O, C)$ dans un espace de dimension d . On dit que S est s_rig3 ssi $dof(S) = \frac{d(d+1)}{2}$ et tout sous-système **rigidifiable** $S' \subset S$ vérifie $dof(S') \geq \frac{d(d+1)}{2}$.

Sur notre exemple, les seuls sous-systèmes rigidifiables sont : AE, ABC, ABD, ACD, ABCD, ABCE et ABCDE. Seuls les sous-systèmes ABC et ABCE sont s_rig3 ; ce sont bien les seuls sous-systèmes rigides et rigidifiables de cet exemple. Tous les sous-systèmes contenant le point D sont sous-rigides et ne sont pas s_rig3 , alors que l'un d'entre eux, BCDE, était s_rig2 (cf. définition 5). Il semble bien que notre nouvelle définition concorde mieux à la rigidité.

Cependant, déterminer si un système est rigidifiable est un problème qui peut s'avérer difficile. Par exemple, un système contenant trois points en 3D n'est rigidifiable que si ces 3 points ne sont pas alignés. Cette condition est trivialement vérifiée pour les systèmes à barres génériques. Mais, dans le cas général, vérifier l'alignement de 3 points est un problème de preuve de théorème difficile : un système basé sur des points et droites et uniquement des contraintes d'incidence pourra contenir 3 points alignés à cause du théorème de Pappus.

Ainsi, s'il est *aisé* de vérifier la rigidifiabilité pour une classe donnée de systèmes géométriques, il semble intéressant d'utiliser la définition 7.

5 Notion de densité et algorithme Dense

Cette section présente l'algorithme proposée par Hoffmann et al [HLS97] pour trouver un sous-système s_rig1 . Cette méthode, basée sur une distribution de flot classique, utilise la propriété de *densité* sur un *graphe objets-contraintes* représentant le système géométrique.

5.1 Densité

Le *graphe objets-contraintes* représentant un système géométrique se définit de la façon suivante :

Definition 8 Graphe objets-contraintes

Le *graphe objets-contraintes* associé à un système géométrique $S = (O, C)$ est un *graphe pondéré* $G = (V, E, w)$ où :

- $V = O$. Chaque objet géométrique de S est un sommet de G .
- $E = C$. Chaque contrainte géométrique de S portant sur un ensemble d'objets $O' \subset O$ devient une (*hyper*-)arête liant tous les sommets correspondants aux objets de O' .
- la fonction de pondération w associe à chaque sommet de V le nombre de degrés de liberté de l'objet géométrique correspondant; à chaque arête, le nombre de degrés de liberté retirés par la contrainte géométrique correspondante.

La figure 1(b) représente le graphe objets-contraintes associé au système géométrique 3D que nous avons utilisé dans nos exemples précédents.

La fonction de pondération w est étendue au calcul du nombre de degrés de liberté d'un sous-système (cf. définition 2). L'évaluation du poids d'un sous-graphe $G' = (V', E', w)$ s'effectue de la façon suivante : $w(G') = \sum_{v \in V'} w(v) - \sum_{e \in E'} w(e)$.

La notion de *densité*, introduite dans [HLS97], est définie comme suit :

Definition 9 Densité

Un graphe pondéré $G = (V, E, w)$ est K -dense si et seulement si $w(G) \leq K$. Plus précisément, on dit que :

- $G = (V, E, w)$ est sur- K -dense si et seulement si $w(G) < K$.
- $G = (V, E, w)$ est exact- K -dense si et seulement si $w(G) = K$.
- $G = (V, E, w)$ est sous- K -dense si et seulement si $w(G) > K$.

La proposition suivante, déductible directement des définitions 3 et 9, établit le lien entre rigidité et rigidité structurelle :

Proposition 1 Soit G le graphe pondéré modélisant un système géométrique S en dimension d . Les deux propositions suivantes sont équivalentes :

- G est exact- $\frac{d(d+1)}{2}$ -dense et aucun sous-graphe de G n'est sur- $\frac{d(d+1)}{2}$ -dense.
- S est un système géométrique s_rig1 .

5.2 Algorithme Dense

L'algorithme Dense a pour but d'extraire un sous-graphe K -dense d'un graphe pondéré. En choisissant $K = \frac{d(d+1)}{2}$, ce sous-graphe correspondra à un sous-système s_rig1 s'il est exact- K -dense ou bien à un sous-système sur- s_rig1 s'il est sur- K -dense. L'algorithme Dense peut retourner un sous-graphe sur- K -dense alors qu'il existe un sous-graphe exact- K -dense dans le graphe objets-contraintes. Détecter une sur-rigidité est cependant un processus utile puisqu'un système contenant un sous-système sur-rigide est sur-rigide.

L'algorithme Dense est basé sur une distribution de flot dans un réseau objets-contraintes $G^* = (S, E', V', T, E^*, w^*)$ construit à partir du graphe objets-contraintes $G = (V, E, w)$:

- La source du réseau est S , son puits est T .
- Chaque objet de V devient un sommet-objet de V' .
- Chaque contrainte de E devient un sommet-contrainte de E' .
- L'ensemble E^* des arcs du réseau se définit par : Pour tout objet v et toute contrainte e , il existe un arc (S, e) de capacité $w^*(S, e) = w(e)$, un arc (v, T) de capacité $w^*(v, T) = w(v)$ et, si e contraint v , un arc (e, v) de capacité $w^*(e, v) = \infty$.

La distribution de flot dans ce réseau, assurée par un classique algorithme de calcul de flot maximum de type Ford-Fulkerson [FF62], représente la répartition des degrés de liberté retirés par les contraintes entre les objets du système. Afin de détecter des systèmes K -denses, on distribuera une capacité supplémentaire de $K + 1$ dans le réseau. Si la distribution de cette surcharge échoue, cela prouvera qu'il existe un sous-graphe possédant moins de $K + 1$ degrés de liberté, c'est-à-dire un sous-graphe K -dense (cf. définition 9).

Cette surcharge de $K + 1$ degrés de liberté doit être appliquée de toutes les façons possibles dans le réseau. Pour cette raison, l'algorithme Dense procède comme suit :

1. Le réseau est construit incrémentalement en introduisant à chaque étape un nouvel objet et toutes les contraintes liées à cet objet et à des objets déjà dans le réseau.
2. A chaque étape, pour chaque sommet-contrainte e nouvellement introduit, on surcharge l'arête (S, e) temporairement par une capacité $K + 1$ ($w^*(S, e) = w(e) + K + 1$) et on tente de distribuer un flot maximum.
3. Si, pour l'une des façons de surcharger l'un de ces réseaux, le flot maximum distribuable ne sature pas toutes les arêtes provenant de la source, alors le sous-réseau traversé lors de la recherche d'un chemin améliorant correspond à un sous-graphe K -dense.

Afin d'identifier des sous-système s_rig2 et non plus s_rig1 , Hoffmann et al introduisent l'heuristique suivante : la surcharge n'est appliquée qu'aux étapes où le réseau contient au moins $d + 1$ objets.

L'exemple suivant va illustrer les problèmes de cet algorithme et de l'heuristique qui lui est adjointe :

1. L'algorithme présente les mêmes failles que la définition 3, c'est-à-dire qu'il peut identifier comme sur- s_rig1 des parties rigides.
2. L'heuristique ne permet pas toujours de retourner des sous-systèmes s_rig2 ³.

5.3 Exemple

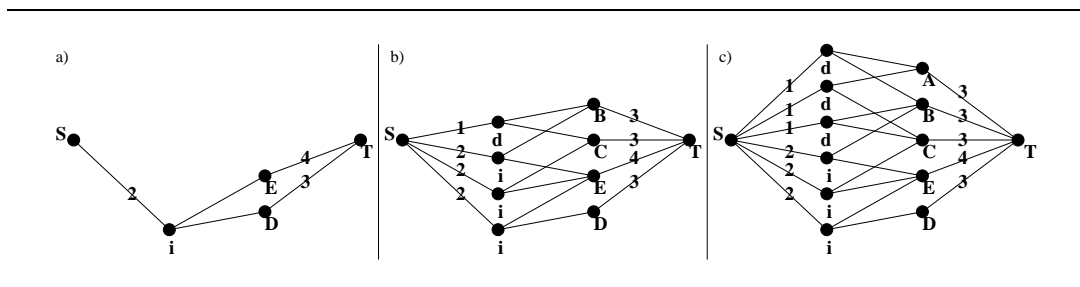


FIG. 2 – 3 étapes de la construction du réseau correspondant à l'exemple de la figure 1(a). a) Etape 2 : la droite E et le point D ont été ajoutés. b) Etape 4 : le points B et C ont été ajoutés. c) Etape 5 : le point A est ajouté. Les valeurs représentées sur les arcs sont les capacités. Les arcs sans valeurs ont une capacité infinie.

Illustrons le déroulement de l'algorithme Dense sur notre exemple de système géométrique 3D⁴. Les objets sont introduits dans le réseau dans l'ordre suivant : E, D, C, B, A. La figure 2 montre 3 étapes de la construction incrémentale du réseau G^* .

A l'étape 1, le réseau est initialisé avec juste l'objet E et aucun sommet-contrainte n'y est introduit. Aucun flot n'est donc distribué à cette première étape.

A l'étape 2, le réseau contient les sommets-objets E et D, et la contrainte d'incidence entre D et E est introduite (sommets i). L'algorithme surcharge l'arc liant la source S à la contrainte i et tente de distribuer ce flot. La distribution du flot surchargé échoue : la capacité de l'arc (S, i) vaut $2 + 6 + 1 = 9$ et il est impossible de distribuer cette capacité sur les arcs liant D et E aux puits T (la somme des capacités de ces arcs vaut $4 + 3 = 7$).

L'algorithme Dense devrait donc s'arrêter ici, retournant le sous-graphe DE. Le poids de ce sous-graphe est $4 + 3 - 2 = 5$ et il est donc sur-6-dense. Le sous-système correspondant est donc identifié sur- s_rig1 bien qu'étant rigide. C'est le premier problème de cet algorithme : il reproduit les erreurs de la définition 3 de rigidité structurelle.

C'est la raison de l'introduction de l'heuristique présentée en fin de section 5.2 : l'algorithme Dense prenant en compte cette heuristique devraient retourner des sous-systèmes s_rig2 et non plus s_rig1 . Rappelons que cette heuristique préconise de ne distribuer la surcharge que dans des réseaux de plus de $d + 1$ objets. Pour ce système 3D, cela signifie que l'algorithme Dense ne distribuera la surcharge qu'à partir de l'étape 4.

La figure 3(a) présente la distribution initiale (sans surcharge) qui est effectuée dans le réseau de l'étape 4. La surcharge est appliquée successivement au deux contraintes liées à l'objet B, dernier objet introduit dans le réseau. La figure 3(b) présente le résultat d'une distribution maximale si

³Hoffmann et al ont alors recours à un processus de rattrapage appelé *Look-up table* afin de fournir un résultat concordant avec la définition 5.

⁴Rappelons qu'en dimension $d = 3$, $\frac{d(d+1)}{2} = 6$ et Dense cherche donc à identifier des sous-graphes 6-dense par application de surcharges valant $6 + 1 = 7$.

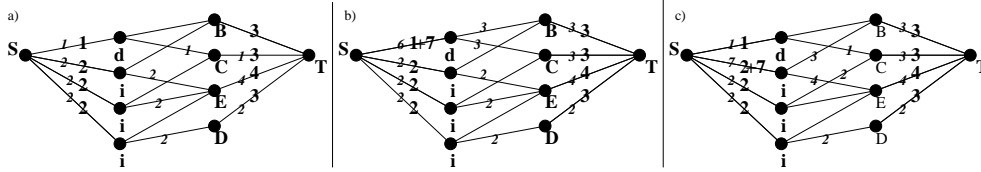


FIG. 3 – Distribution du flot pour le réseau de l'étape 4. a) Distribution du flot correspondant aux contraintes. b) Distribution de la surcharge (+7) sur l'arc (S, d_{BC}) . c) Distribution de la surcharge (+7) sur l'arc (S, i_{BE}) . Les nombres en italiques représentent l'attribution de flot maximale trouvée par la fonction *Augmenter*.

la surcharge est appliquée sur la contrainte d liée à B. La distribution sur ce réseau échoue : l'arc (S, d) de capacité (surchargée) $1 + 7 = 8$ n'est pas saturé. L'algorithme *Dense* retourne alors le sous-graphe BC correspondant aux sommets traversés pendant la tentative de distribution du flot. Ce sous-graphe contient moins de $d + 1$ objets, et le sous-système correspondant est donc trivial. C'est le second problème de l'algorithme : l'utilisation de l'heuristique ne lui permet pas de retourner à coup sûr un sous-système `s_rig2` ou `sur-s_rig2`.

Hoffmann et al [Sit00] utilisent alors un processus de rattrapage que nous avons mentionné à la section précédente : une *Look-up table*. Cette table contient un ensemble de motifs *rigides et non-triviaux* et des règles de construction de ces motifs à partir de sous-motifs `s_rig1` et triviaux. Dans le cas présent, aucune règle de cette table ne s'applique puisque le réseau ne contient aucun sous-système rigide. Nous reprenons donc l'algorithme là où il s'était arrêté.

La figure 3(c), représente la distribution de flot maximum pour l'application de la surcharge à la seconde contrainte liée à B. Ce cas débouche aussi sur un échec de distribution, et identifie ainsi le sous-graphe BCE. Le sous-système correspondant est trivial. Aucune règle de la *Look-up table* ne peut s'appliquer pour la même raison que précédemment.

On passe donc à l'étape 5 de construction du réseau (cf. figure 2(c)). L'introduction du point A ajoute les deux contraintes de distance liant A, B et A, C. La distribution de la surcharge sur l'une ou l'autre de ces arêtes conduit à l'identification d'un sous-système `sur-6-dense` : AB ou AC selon l'arête surchargée. Ces sous-systèmes sont encore une fois triviaux. La *Look-up table* pourrait être capable de reconstruire le sous-système rigide et non trivial ABCE à partir de AB ou AC si elle contenait les règles adéquates⁵.

L'algorithme *Dense* tenant compte de l'heuristique et possédant une *Look-up table* adéquate peut donc finir par retourner un sous-système `s_rig2`. Cependant, il est facile d'exhiber des systèmes de taille un peu plus grande où les motifs et règles qui devraient servir à pallier les échecs de l'algorithme et de l'heuristique ne peuvent pas raisonnablement se trouver *a priori* dans la *Look-up table*. De plus, nous avons montré qu'un sous-système non rigide peut être `s_rig2` (cf. section 2). C'est pourquoi nous proposons dans la section 6 une réparation générale de l'algorithme *Dense* qui tient compte de la définition 7 de la rigidité structurelle.

5.4 Etape de minimisation

L'article [HLS97] présente aussi l'algorithme *Minimal_Dense*. Cet algorithme est entièrement basé sur l'algorithme *Dense* que nous avons présenté à la section précédente : il commence par identifier un sous-graphe K -dense (fonction *Dense*) avant de minimiser la taille de ce sous-graphe en essayant de lui retirer des sommets un à un et en appliquant la fonction *Dense* après chaque retrait. Cette étape de minimisation est décrite plus en détail dans l'article [HLS97].

⁵La *Look-up table* pourrait aussi reconstruire le sous-système rigide ABC si l'on acceptait que, dans le cas où elle intervient (cas considérés comme "spéciaux"), l'algorithme *Dense* pourrait retourner un sous-système trivial, ne répondant alors plus à la définition 5 de rigidité structurelle.

Le résultat de cet algorithme est un sous-graphe *minimal*⁶ K -dense inclus dans le sous-graphe K -dense initialement identifié par la fonction `Dense`.

L'intérêt de la minimisation est double :

1. un sous-graphe minimal $\frac{d(d+1)}{2}$ -dense correspond à un sous-système minimal `s_rig` ou `sur-s_rig`.
2. pour les besoins de méthodes de résolution du type *rigidification récursive*, obtenir de petits sous-systèmes conduit à une décomposition plus fine du système qui rendra la résolution plus efficace.

Le premier point est un corollaire de la proposition 1. En effet, si G est minimal $\frac{d(d+1)}{2}$ -dense, alors il ne peut contenir de sous-graphe $\frac{d(d+1)}{2}$ -dense (par définition de minimal). Ainsi, tous ses sous-graphes sont sous- $\frac{d(d+1)}{2}$ -dense.

6 l'algorithme `S_Rigide`

Cette section présente un nouvel algorithme se basant sur l'algorithme `Dense` et respectant notre définition de rigidité structurelle (cf. définition 7).

Pour ce faire, seule l'opération consistant à surcharger les arcs liant la source aux nouvelles contraintes un à un doit être modifiée. Pour le comprendre, il faut faire la constatation suivante : l'application d'une surcharge valant $\frac{d(d+1)}{2} + 1$ pour un système en dimension d correspond à poser un repère sur ce système et à lui retirer un degré de liberté supplémentaire. Ce principe était déjà utilisé dans la méthode structurelle proposée par Michelucci et al dans [LM97]. L'échec de la distribution de cette surcharge indique soit qu'un repère ne peut être posé sur le système, soit qu'une fois le repère posé, le système ne dispose plus d'aucun degré de liberté. Cet échec indiquera donc bien un système `sur-s_rig3` ou `s_rig3` à condition que le sous-système sur lequel on essaye de poser un repère soit *rigidifiable* (cf. définition 6). C'est cette condition nécessaire qui manque à l'algorithme `Dense` : rien n'assure que le sous-système constitué d'une seule contrainte et de tous ses objets soit rigidifiable. Michelucci et al avaient d'ailleurs noté ce fait dans leur article [LM97] puisque, en 3D, ils proposaient de répartir la surcharge entre 3 points et non pas 2. Notre réparation est donc une généralisation de ce principe : il faut appliquer la surcharge sur chacun des sous-systèmes rigidifiables du réseau contenant le dernier objet introduit.

Pour ce faire, nous introduisons un nouveau sommet R dans le réseau. Un arc (S, R) , dont la capacité égale la surcharge que l'on doit appliquer, lie la source S à ce sommet R . Un arc de capacité infinie lie le sommet R à chaque objet du sous-système rigidifiable sur lequel on tente de poser le repère. Si tous les arcs provenant de la source ne peuvent être saturés après ajout du sommet R , cela indiquera qu'il était impossible à la fois de poser un repère et de retirer un degré de liberté supplémentaire au sous-système rigidifiable lié à R . Le sous-réseau traversé lors de la dernière tentative de redistribution correspondra donc à un sous-graphe $\frac{d(d+1)}{2}$ -dense, comme dans l'algorithme `Dense`, mais le sous-système correspondant vérifiera de plus la condition d'être rigidifiable.

Il suffit en fait de tenter de distribuer la surcharge vers tous les sous-systèmes rigidifiables **minimaux** (au sens de l'inclusion). En effet, un sous-système rigidifiable est soit minimal, soit il contient un autre sous-système rigidifiable. Comme la surcharge doit être appliquée sur tous les sous-systèmes rigidifiables, elle sera en particulier appliquée sur tous les sous-systèmes rigidifiables minimaux. De plus, il est impossible que la distribution échoue sur un sous-système rigidifiable sans échouer sur tous les sous-systèmes rigidifiables minimaux qu'il contient, par simple propriété de distribution de flot : si la distribution d'un flot via un ensemble E de sommets liés au puits échoue, elle échouera aussi pour tout sous-ensemble $E' \subset E$ de ces sommets.

De plus, un sous-système rigidifiable minimal contient toujours moins de 3 objets dans la classe

⁶Un sous-graphe K -dense est minimal s'il ne contient aucun sous-graphe K -dense autre que lui-même.

des systèmes basés sur points, droites, plans, distance, angle, incidence, colinéarité et coplanarité. Cette propriété se prouve par énumération de tous les systèmes rigidifiables minimaux de cette classe. Il en résulte que le nombre de sous-systèmes rigidifiables minimaux contenant le dernier objet introduit dans le réseau est au pire quadratique dans la taille du réseau. Ceci assure que notre algorithme est polynomial pour les systèmes cités précédemment, sous-réserve qu'il soit facile de décider de la rigidifiabilité d'un sous-système de cette classe.

6.1 Exemple

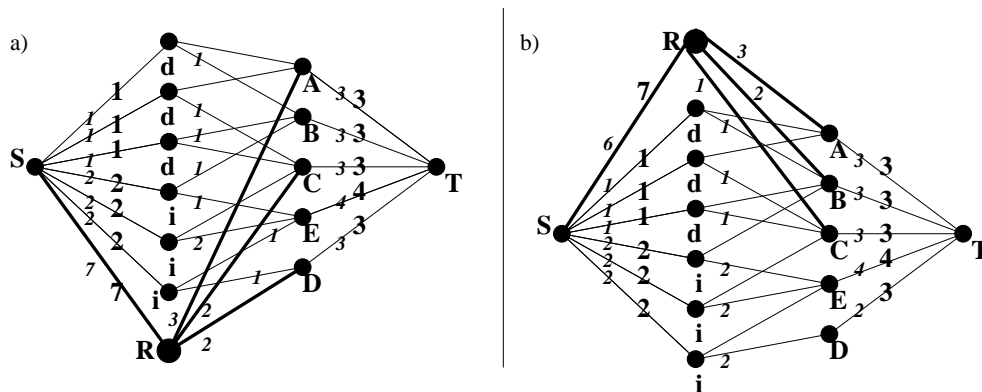


FIG. 4 – Distribution du flot pour le réseau de l'étape 5. a) Le sommet spécial R est lié aux objets A , C et D . b) Le sommet spécial R est lié aux objets A , B et C . Les nombres en italiques représentent le flot maximum sur chaque arc.

Illustrons le déroulement de l'algorithme `S_Rigide` sur notre exemple de système géométrique 3D. On reprend l'ordre de construction incrémental du réseau proposé à la figure 2 : E , D , C , B et A .

Aux étapes 1, 2, 3 et 4, aucun sous-système rigidifiable n'est présent dans le réseau et l'algorithme n'applique donc aucune surcharge.

A l'étape 5, le système complet contient les sous-systèmes rigidifiables minimaux suivants : AE , ABC , ABD , ACD . La figure 4 présente les tentatives de distribution de la surcharge sur les sous-systèmes rigidifiables ACD et ABC .

On constate que, pour le sous-système minimal rigidifiable ACD , la surcharge liée aux sommets A , C et D du réseau peut être distribuée (cf. figure 4(a)). Tous les arcs issus de la source sont saturés et aucun sous-système `s_rig3` n'est donc identifié. Cela est normal puisque le sous-système ACD n'est ni `s_rig3` ni rigide : le système possède $3 + 3 + 3 - 1 = 8$ degrés de liberté et le point D se déplace indépendamment des points A et C . Le même phénomène se produit pour le sous-système rigidifiable minimal ABD .

Sur le réseau correspondant à la tentative de distribution de la surcharge sur le sous-système rigidifiable minimal ABC (cf. figure 4(b)), la distribution échoue et le sous-graphe ABC est identifié comme étant sur-6-dense. Cela est correct puisque le triangle ABC constitue bien à la fois un sous-système rigidifiable, exact-6-dense, `s_rig3` et rigide. Il en aurait été de même pour la tentative de distribution de la surcharge sur le sous-système AE . Cette tentative aurait identifié le sous-système $ABCE$, ce qui est normal puisqu'il s'agit bien d'un sous système rigidifiable, exact-6-dense, `s_rig3` et rigide.

On constate donc que, sur cet exemple, notre algorithme a su identifier les parties effectivement `s_rig3` (et rigides) et uniquement celles-ci.

7 Conclusion

Cet article éclaircit les différences fondamentales entre rigidité, rigidités structurelles et densité. Ces différences impliquent que l'algorithme *Dense* ne pouvait retourner à coup sûr une partie effectivement rigide ou structurellement rigide.

Par ailleurs, il propose l'introduction de la notion de *rigidifiabilité*, notion géométrique informellement définie comme la possibilité pour un système d'être remplacé par un repère local, dans la définition de la rigidité structurelle. Cette nouvelle définition semble mieux correspondre à la rigidité que la définition précédente. Cependant, décider si un système est rigidifiable est un problème difficile dans le cas général. Il faudra donc identifier des classes de systèmes géométriques où ce problème est facile si l'on souhaite utiliser une définition de rigidité structurelle plus conforme à la rigidité.

A partir d'une adaptation polynomiale de l'algorithme *Dense*, il présente un algorithme pour la détection de sous-systèmes répondant à notre définition de rigidité structurelle pour les systèmes 2D et 3D constitués de points, droites et plans soumis à des contraintes de distance, angle, incidence, colinéarité et coplanarité.

Références

- [BFH⁺95] William Bouma, Ioannis Fudos, Christoph Hoffmann, Jiazhen Cai, and Robert Paige. Geometric constraint solver. *Computer Aided Design*, 27(6) :487–501, 1995.
- [DMS97] Jean-François Dufourd, Pascal Mathis, and Pascal Schreck. Formal resolution of geometrical constraint systems by assembling. In C. Hoffmann and W. Bronswort, editors, *Proc. Fourth Symposium on Solid Modeling and Applications*, pages 271–284, 1997.
- [FF62] L.R. Ford and D.R. Fulkerson. *Flows in Networks*. Princeton University Press, 1962.
- [FH97] Ioannis Fudos and Christoph Hoffmann. A graph-constructive approach to solving systems of geometric constraints. *ACM Transactions on Graphics*, 16(2) :179–216, 1997.
- [Hen92] Bruce Hendrickson. Conditions for unique realizations. *SIAM j Computing*, 21(1) :65–84, 1992.
- [HLS97] Christoph Hoffmann, Andrew Lomonosov, and Meera Sitharam. Finding solvable subsets of constraint graphs. In *Proc. Constraint Programming CP'97*, pages 463–477, 1997.
- [HLS00] Christoph Hoffmann, Andrew Lomonosov, and Meera Sitharam. Decomposition plans for geometric constraint systems. In *Proc. J. Symbolic Computation 2000*, 2000.
- [JTNR00] Christophe Jermann, Gilles Trombettoni, Bertrand Neveu, and Michel Rueher. A constraint programming approach for solving rigid geometric systems. In *Principles and Practice of Constraint Programming, CP 2000*, volume 1894 of *LNCS*, pages 233–248, 2000.
- [Kra92] Glenn Kramer. *Solving Geometric Constraint Systems*. MIT Press, 1992.
- [Lam70] G. Laman. On graphs and rigidity of plane skeletal structures. *J. Eng. Math.*, 4 :331–340, 1970.
- [LM97] Hervé Lamure and Dominique Michelucci. Qualitative study of geometric constraints. In Beat Brüderlin and Dieter Roller, editors, *Workshop on Geometric Constraint Solving and Applications*, pages 134–145, Technical University of Ilmenau, Germany, 1997.
- [Sit00] Meera Sitharam. Personal communication on the minimal dense algorithm. 2000.