Complexité de la Détection de Rigidité Dans les Systèmes de Contraintes Géométriques

Christophe Jermann **Gilles Trombettoni**
Projet COPRIN,
Laboratoires INRIA-I3S-CERMICS
2004, route des Lucioles, BP 93,
06902 Sophia Antipolis
France
emails: `{Christophe.Jermann, Gilles.Trombettoni}@sophia.inria.fr`

Abstract

Cet article traite de la complexité du problème *Minimum Dense*. La densité est une propriété structurelle d'un graphe modélisant un système géométrique. Elle est utilisée pour détecter des sous-systèmes rigides. Identifier un sous-graphe dense nécessite une analyse des degrés de liberté du système géométrique. Cette analyse est l'étape clé des nombreux algorithmes ayant pour but la détection de rigidité ou la résolution de systèmes géométriques [DMS98, BFH$^+$95]. En particulier, la méthode de rigidification récursive [Kra92, HLS00] utilise cette propriété pour identifier des sous-systèmes rigides à résoudre séparément avant de les assembler.

Le problème *Minimum Dense*, consistant à identifier un sous-graphe dense de taille minimum, a été prouvé NP-difficile dans le cas général [HLS97]. Dans cet article, nous introduisons une restriction qui rend le problème traitable en temps polynomial. De plus, cette restriction peut être vérifiée en temps polynomial.

1 Introduction

La modélisation par contraintes est une approche prometteuse dans le domaine de la conception assistée par ordinateur. Elle permet de construire un objet en spécifiant, de façon déclarative, les propriétés géométriques qui le caractérisent.

Un système géométrique peut être transcrit en un ensemble d'équations non-linéaires généralement algébriques. La *rigidification récursive* est l'une des approches principales étudiées par la communauté des contraintes géométriques [DMS98, FH97, JTNR00, Kra92, BFH$^+$95, HV95] pour résoudre ces systèmes[1]. La rigidification récursive est une méthode constructive permettant de résoudre un système rigide en identifiant ses sous-parties rigides puis en les assemblant récursivement. Cette méthode opère au niveau géométrique et produit une séquence de petits sous-systèmes d'équations appelés *blocs*. Chaque bloc de la séquence est alors résolu séparément, fournissant l'ensemble des configurations du sous-système géométrique correspondant. Ces configurations sont alors combinées pour former les solutions du système géométrique global.

Détecter un sous-système rigide de petite taille est l'étape clé de la méthode de rigidification récursive (cf. [HLS97]). Différentes approches pour accomplir cette étape ont été proposées.

Les premières tentatives [DMS98, Kra92, FH97, BFH$^+$95] étaient basées sur la reconnaissance de schémas spécifiques rigides au sein du système. Ces approches étaient limitées par leur répertoires de schémas recensés qui ne pouvaient couvrir toutes les instances de sous-systèmes rigides.

Des approches plus récentes sont basées sur une analyse des degrés de liberté du système, et l'utilisation d'une propriété structurelle pour la détection de rigidité [LM98].

L'approche proposée par Hoffmann et al. [HLS97] est aussi basée sur cette propriété. Elle consiste à utiliser un algorithme de flot pour détecter un sous-graphe *minimal dense*. Intuitivement, un sous-graphe dense correspond à un sous-système géométrique (génériquement) rigide. Ce sous-graphe est minimal s'il ne contient aucun sous-graphe dense. L'algorithme de flot, appelé `Minimal_Dense` a une complexité $O(n^2(m+n))$ en pire cas, n étant le nombre de sommets (objets géométriques) du graphe et m son nombre d'arêtes (contraintes géométriques).

Détecter un sous-graphe dense de taille *minimum* est un problème prouvé NP-difficile dans le cas général [HLS98]. Personne n'a, dès lors, tenté d'exhiber des sous-classes polynomiales de ce problème. Dans cet article, nous présentons une restriction qui définit une sous-classe de systèmes géométriques pour lesquels l'utilisation de l'algorithme `Minimal_Dense` permet de répondre en temps polynomial au problème *Minimum Dense*. De plus, cette restriction, qui impose que le graphe initial ne contienne aucun sous-graphe sur-dense, *est vérifiable en temps polynomial* (cf. fin de la section 2).

Le plan de l'article est le suivant :

- La section 2 définit la notion de densité. Par mesure de simplicité, nous ne ferons aucun lien avec le domaine géométrique dans cette section, la densité étant uniquement une propriété de graphe.

- La section 3 démontre que, sous notre restriction, le problème *Minimum Dense* est décidable en temps polynomial par un algorithme basé sur `Minimal_Dense`.

[1]Les méthodes symboliques sont trop lentes en pratique pour attaquer de grands systèmes non-linéaires [RM96, Laz81].

- La section 4 établit le lien avec le domaine géométrique. Tout d'abord, nous expliquons comment un système géométrique peut être vu comme un graphe pondéré. Le paragraphe 4.1 donne alors une idée de la méthode de rigidification récursive. Le paragraphe 4.2 présente enfin le lien entre densité et rigidité. Une fois établi, ce lien nous permet d'étendre notre résultat de complexité du problème *Minimum Dense* au problème *Minimum Structurellement Rigide* (cf. paragraphe 4.3) dont l'intérêt est motivé en fin de section.

2 Le Problème *Minimum Dense*

Une analyse des degrés de liberté d'un système géométrique est une analyse structurelle menée sur un graphe pondéré $G = (V, E, w)$ modélisant ce système. Le lecteur se référera à la section 4 pour comprendre comment ce graphe est produit à partir du système géométrique.

La fonction de pondération w associe un entier strictement positif à chaque sommet et chaque arête de G. Par extension, nous définissons cette fonction pour des sous-graphes $G' = (V', E', w)$ de G 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], peut être définie de la façon suivante :

Définition 1 *Soient $G = (V, E, w)$ un graphe pondéré et W un entier (pas nécessairement positif). G est* **W-dense** *ssi $w(G) \leq W$.*

G est **exact-W-dense** *ssi $w(G) = W$.*

G est **sur-W-dense** *ssi $w(G) < W$.*

G est **sous-W-dense** *ssi $w(G) > W$.*

La figure 1(a) présente un graphe pondéré 3-dense.

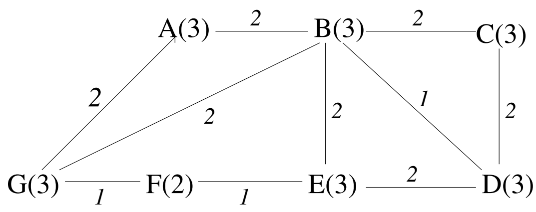

Figure 1: (a) Un graphe pondéré (exact-)3-dense. (b) Les sous-graphes $< B, C, D, E >$ et $< A, B, G >$ sont minimaux 3-denses. $< A, B, G >$ est minimum.

Définition 2 *Soient $G = (E, V, w)$ un graphe pondéré, W un entier et G' un sous-graphe de G.*

G' est **minimal W-dense** *ssi G' est W-dense et ne contient aucun sous-graphe W-dense.*

G' est **minimum W-dense** *ssi G' est W-dense il n'existe aucun sous-graphe W-dense de G ayant moins de sommets que G'.*

Le problème de trouver un sous-graphe minimum W-dense de G est appelé **Minimum Dense**.

La figure 1(b) présente des exemples de sous-graphes minimaux et minimum 3-denses.

La rigidification récursive produit une séquence de *blocs* d'équations. Un sous-graphe identifié comme dense correspondra à l'un de ces blocs. Il est donc souhaitable d'identifier les sous-graphes les plus petits possibles afin d'obtenir des blocs plus faciles à résoudre. Il semble donc intéressant de chercher à identifier des sous-graphes minimum W-denses. Hoffmann et al. ont prouvé dans [HLS97] que ce problème est NP-difficile dans le cas général[2].

Notre résultat de polynomialité est basé sur la restriction suivante : *Le graphe initial ne contient aucun sous-graphe sur-W-dense*. Nous nommerons cette restriction *non-sur-dense* dans la suite de cet article. Cette restriction peut être vérifiée en temps polynomial : un appel à l'algorithme `Minimal_dense`, décrit dans [HLS97], pour une densité $W - 1$ détecte un sous-graphe sur-W-dense en temps polynomial s'il en existe un.

Appelons `MD` la classe générale (NP-difficile) du problème Minimum Dense et $\mathtt{MD}_{nsd}$ la sous-classe du problème dont les instances vérifient la restriction *non-sur-dense*.

3 Complexité du problème *Minimum Dense*

Cette section prouve que $\mathtt{MD}_{nsd}$ est dans P, c'est-à-dire que la restriction *non-sur-dense* rend le problème Minimum Dense décidable en temps polynomial.

Ce résultat repose essentiellement sur un autre résultat de polynomialité, celui du problème **Minimal Dense**. En effet, l'algorithme `Minimal_Dense` trouve un sous-graphe minimal W-dense en temps polynomial dans un graphe et pour un W quelconques.

La proposition suivante, alliée à l'algorithme `Minimal_Dense`, est la clé de la complexité de $\mathtt{MD}_{nsd}$.

Proposition 1 *Soient G un graphe pondéré* **vérifiant la restriction** *non-sur-dense, et G_1, G_2 deux sous-graphes de G.*

Si $G1$ et $G2$ sont minimaux *W-dense, alors ils ne partagent aucune arête.*

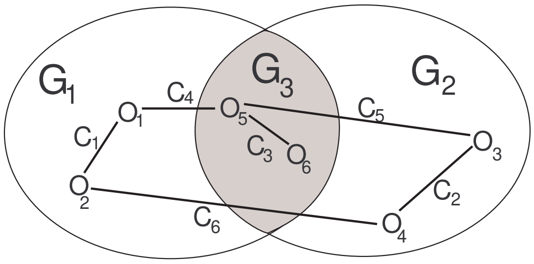

Figure 2: Exemples de graphes G_1 et G_2 ayant pour intersection G_3.

[2]En utilisant une transformation polynomiale depuis Clique [Pap94], le problème de décision correspondant a été prouvé NP-complet.

Démonstration (par contradiction) Supposons que G_1 et G_2 partagent des arêtes. Puisque G_1 et G_2 sont minimaux denses, G_1 (resp. G_2) ne peut être un sous-graphe de G_2 (resp. G_1).

Soient :

- G_3 l'intersection de G_1 et G_2;
- V_3 l'ensemble des sommets de G_3, V_1 celui de $G_1 - G_3$ et V_2 celui de $G_2 - G_3$;
- $v_i = \sum_{v \in V_i} w(v)$ $(1 \leq i \leq 3)$;
- E_{ij} l'ensemble des arêtes joignant un sommet de V_i à un sommet de V_j $(1 \leq i \leq j \leq 3)$;
- $e_{ij} = \sum_{e \in E_{ij}} w(e)$ $(1 \leq i \leq j \leq 3)$.

La figure 2 illustre ces différents ensembles sur un exemple. Dans cet exemple, les sommets o_1, o_2, o_5 et o_6 sont dans G_1, les sommets o_3, o_4, o_5 et o_6 sont dans G_2 et les sommets o_5 et o_6 sont dans G_3. Il en résulte les ensembles suivants : $V_1 = \{o_1, o_2\}$, $V_2 = \{o_3, o_4\}$, $V_3 = \{o_5, o_6\}$, $E_{11} = \{c_1\}$, $E_{12} = \{c_6\}$, $E_{13} = \{c_4\}$, $E_{22} = \{c_2\}$, $E_{23} = \{c_5\}$ et $E_{33} = \{c_3\}$.

Les relations (1) et (2) établissent le fait que G_1 et G_2 sont minimaux W-dense :

$$\begin{array}{ll} (1) & v_1 + v_3 - e_{11} - e_{13} - e_{33} = W \\ (2) & v_2 + v_3 - e_{22} - e_{23} - e_{33} = W \end{array}$$

Notez que la restriction *non-sur-dense* empêche les poids de G_1 et de G_2 d'être inférieurs à W.

Par hypothèse, G_3 contient au moins une arête. Ceci implique que G_3 est sous-dense. En effet, si G_3 était W-dense, G_1 et G_2 ne seraient alors pas minimaux puisqu'ils contiendraient un sous-graphe dense plus petit. Le relation (3) exprime le fait que G_3 est sous-dense.

$$(3) \quad v_3 - e_{33} = W + k \quad (\mathbf{k} > \mathbf{0})$$

Le relation suivante correspond au calcul du poids de $G_1 \cup G_2$:

$$\begin{aligned} (4) \quad w(G_1 \cup G_2) = {} & v_1 + v_2 + v_3 \\ & - e_{11} - e_{13} - e_{22} \\ & - e_{23} - e_{33} - e_{12} \end{aligned}$$

La relation (5) se construit à partir des 3 premières relations :

$$\begin{aligned} (5) : (1) + (2) - (3) = {} & v_1 + v_2 + v_3 \\ & - e_{11} - e_{13} - e_{22} \\ & - e_{23} - e_{33} \\ = {} & W - k \end{aligned}$$

En remplaçant la relation (5) dans la (4) on obtient :

$$(6) \quad w(G_1 \cup G_2) = W - k - e_{12}$$

Algorithm 1 `Minimum_Dense` (`G` : graphe pondéré; `W` : entier) retourne un sous-graphe minimum W-dense de G

```
Minimaux-Denses ← ∅
G' ← Minimal_Dense (G, W) {Algorithme de flot de Hoffmann et al.}
while G' ≠ GrapheVide do
  Minimaux-Denses ← Minimaux-Denses ∪ {G' }
  G ← G − Arêtes(G')
  G' ← Minimal_Dense (G, W)
end while
Return Min(Minimaux-Denses) {Min retourne l'élément de plus petite taille de l'ensemble passé en paramètre}
```

Puisque $k > 0$ et $e_{12} \geq 0$, la relation (6) signifie que $G_1 \cup G_2$ est sur-dense ($w(G_1 \cup G_2) < W$), ce qui contredit l'hypothèse de la proposition 1.

□

La proposition 1 nous permet de concevoir l'algorithme polynomial `Minimum_Dense`, pour résoudre le problème Minimum Dense. En effet, les sous-graphes minimaux denses ne partageant pas d'arêtes sous la restriction *non-sur-dense*, on peut calculer l'ensemble de tous les sous-graphes minimaux denses, appelé `Minimaux-Denses` dans l'algorithme, de manière gloutonne : L'algorithme `Minimal_Dense` est appelé itérativement pour trouver un sous-graphe minimal dense, dont les arêtes sont alors retirés du graphe principal avant la prochaine itération. La proposition 1 assure qu'aucun sous-graphe minimal dense n'est raté de cette façon. Le sous-graphe minimum dense est alors le plus petit dans l'ensemble `Minimaux-Denses`.

Complexité de $\texttt{MD}_{nsd}$ **:** l'algorithme `Minimum_Dense` est basé sur des appels à l'algorithme `Minimal_Dense` qui est $O(n^2(m+n))$. Le nombre d'appels à cet algorithme est borné par le nombre d'arêtes m dans le graphe initial. Ainsi, `Minimum_Dense` est $O(m \times n^2(m+n))$ et $\texttt{MD}_{nsd}$ est dans P.

Poids bornés : La transformation polynomiale utilisée dans [HLS98] pour prouver que `MD` est NP-complet génère des instances de graphes dont les sommets et arêtes ont des poids non-bornés, dépendants de la taille du sous-graphe minimum dense. En pratique, lorsque le graphe pondéré correspond à un système géométrique, ces poids sont bornés. Il est intéressant de noter que cette hypothèse n'est pas utilisée dans notre preuve de polynomialité. En effet, l'algorithme `Minimal_Dense` demeure polynomial que les poids soient bornés ou pas, ceci par propriété de l'algorithme de flot [AMO93].

4 Application à la Géométrie

Informellement, un système géométrique est constitué d'un ensemble d'*objets géométriques* dont les positions et orientations doivent satisfaire un ensemble de *contraintes géométriques*. Un objet géométrique est représenté par un ensemble de paramètres géométriques. Une contrainte géométrique est un ensemble d'équations sur les paramètres des objets géométriques qu'elle contraint.

Un système géométrique peut être représenté par un *graphe objets-contraintes* $G = (O, C, w)$, graphe pondéré dont :

- chaque sommet $o \in O$ représente un objet géométrique. Le poids d'un sommet caractérise le nombre de degrés de liberté de l'objet associé, c-a-d, le nombre de paramètres qui doivent être déterminées pour fixer cet objet relativement à un repère global.
- une arête $c \in C$ représente une contrainte géométrique. Son poids correspond au nombre de degrés de liberté retirés par la contrainte (généralement égal au nombre d'équations modélisant la contrainte).

Par exemple, un point en 2D possède 2 degrés de liberté, et une ellipse 3. Les sommets correspondants auront respectivement poids 2 et 3. Une distance entre points retire 1 degré de liberté, et une coïncidence 2. Les arêtes correspondantes auront respectivement poids 1 et 2. Le poids d'un sous-graphe correspond alors au nombre de degrés de liberté du sous-système correspondant.

4.1 Rigidification récursive

La rigidification récursive est l'une des principales méthodes utilisée pour résoudre les systèmes géométriques [DMS98, FH97, JTNR00, Kra92, BFH$^+$95, HV95]. Elle produit une séquence de sous-systèmes rigides plus simples à résoudre que le système complet. Informellement, un système est rigide s'il est indéformable et déplaçable dans tout l'espace géométrique considéré.

Le schéma de la méthode de rigidification récursive est le suivant :

1. Trouver un sous-système rigide
2. Tant que possible, *étendre* ce sous-système[3]
3. Remplacer le sous-système courant par un sommet unique représentant un repère local associé à ce sous-système pour la résolution
4. Répéter les étapes 1 à 3 jusqu'au point fixe[4]

[3] Une extension d'un sous-système rigide consiste à lui ajouter l'un de ses objets géométriques voisins de telle sorte que le sous-système résultant soit encore rigide.

[4] Un point fixe est atteint lorsque l'étape 1 échoue, c'est à dire qu'il n'existe plus aucun sous-système rigide dans le système.

Lorsque le point fixe est atteint, chaque sous-système rigide est résolu dans le repère local qui lui est associé à l'étape 3 de la méthode. Ensuite, toutes les solutions des sous-systèmes sont combinées en des solutions du système global.

Résoudre un sous-système rigide dans son repère local revient à résoudre un ensemble de blocs d'équations. Un bloc d'équations correspond soit à l'étape 1 de la méthode, soit à *une* extension de l'étape 2 de la méthode. De façon général, un bloc d'équations comprenant moins d'équations sera plus facile à résoudre. C'est pourquoi il est crucial pour les performances de cette méthode que les blocs d'équations produits par les étapes 1 et 2 soient aussi petits que possibles.

Comme il sera expliqué à la section suivante, sous l'hypothèse où la restriction *non-sur-dense* est vérifiée, un sous-graphe $\frac{d(d+1)}{2}$-dense *correspond* à un sous-système rigide. Il semble donc intéressant d'utiliser l'algorithme `Minimum_Dense` pour la détection de sous-systèmes rigides de taille minimum (étape 1 de la méthode de rigidification récursive).

4.2 Rigidité structurelle

La méthode de rigidification récursive fait appel à la propriété de *rigidité structurelle*. La rigidité structurelle est une généralisation de la caractérisation [Lam70] de la rigidité pour les systèmes à barres en 2D (systèmes constitués uniquement de points et distances). Elle est considéré comme une bonne heuristique pour identifier des sous-systèmes rigides dans le cas général [BFH+95, FH97, HLS00, LM98]. Cette notion est très proche de la notion de densité :

Définition 3 *Soit G le graphe objets-contraintes correspondant à un système géométrique S.*
En dimension d, S est **structurellement rigide** *ssi G est $\frac{d(d+1)}{2}$-dense et ne contient aucun sous-graphe sur-$\frac{d(d+1)}{2}$-dense.*

Il est immédiat de vérifier que, sous l'hypothèse *non-sur-dense*, un graphe objets-contraintes $\frac{d(d+1)}{2}$-dense correspond toujours à un système structurellement rigide.

Considérons que le graphe présenté à la figure 1 correspond à un système géométrique en 2D. On peut constater que ce système est structurellement rigide puisque le graphe est 3-dense et vérifie la restriction. Les 3 degrés de liberté du graphe représentent le fait que les objets de ce système sont rigides relativement les uns aux autres et peuvent être déplacés (translations et rotation) conjointement.

4.3 Complexité du problème Minimum Structurellement Rigide

Les résultats de complexité obtenus pour les classes `MD` et $\texttt{MD}_{nsd}$ du problème Minimum Dense peuvent être appliqués aux classes correspondantes du problème Minimum Structurellement Rigide, le problème de la recherche d'un sous-système structurellement rigide de taille minimum. Soient `SR` la classe la plus générale de ce problème, et $\texttt{SR}_{nsd}$ la sous-classe de `SR` sous la restriction *non-sur-dense*.

- SR est aussi NP-difficile. La preuve de NP-difficulté de MD [HLS98] s'appliquait uniquement pour une densité $W = 0$, alors que la propriété de rigidité structurelle considère $W = \frac{d(d+1)}{2}$ en dimension $d > 0$. Cependant, la transformation polynomiale depuis Clique peut être généralisée à tout W.
- Comme expliqué plus haut, sous la restriction *non-sur-dense*, la rigidité structurelle et la $\frac{d(d+1)}{2}$-densité sont équivalentes. Puisque Minimum_Dense peut trouver un sous-graphe minimum W-dense pour tout W, SR_{nsd} est aussi dans P.

4.4 Intérêt du problème Minimum Structurellement Rigide

Comme expliqué auparavant, chaque étape de la méthode de rigidification récursive produit un bloc d'équations qu'il faudra résoudre. Ceci explique l'importance de la taille des sous-graphes identifiés par l'étape 1 de cette méthode. Il semble donc intéressant d'utiliser l'algorithme Minimum_Dense pour accomplir cette étape quand cela est possible.

En pratique, des heuristiques sont parfois associées à l'algorithme Minimal_Dense pour l'accomplissement de l'étape 1 de la rigidification récursive. Ces heuristiques ne peuvent pas toujours être simplement prises en compte si c'est l'algorithme Minimum_Dense qui accomplit cette étape.

Notamment, une heuristique communément associée à l'algorithme Minimal_Dense vise à assurer la correction de l'étape 3 de la rigidification récursive. En effet, cette étape remplace le sous-système identifié par un repère local. Assurer qu'un sous-système peut être remplacé par un repère local est un problème difficile dans le cas général. L'heuristique visant à répondre à cette question est la suivante : un sous-système est remplaçable par un repère uniquement s'il contient au moins $d + 1$ objets en dimension d.

L'algorithme Minimal_Dense peut prendre en compte cette heuristique pour ne produire que des sous-graphes denses contenant au moins k objets, pour un k quelconque. Il en résulte que les sous-graphes retournés par cette version de Minimal_Dense ne sont pas nécessairement minimaux, puisqu'ils doivent contenir au moins k objets. Ainsi, la proposition 1 sur laquelle est basée notre résultat de complexité et notre algorithme Minimum_Dense ne peut s'appliquer pour ces sous-graphes, c-a-d, ils peuvent partager des arêtes. L'algorithme Minimum_Dense ne peut donc tenir compte de cette heuristique pour produire des sous-graphes *minimum dense de taille au moins k*.

Cependant, bien qu'intéressante en pratique pour les systèmes à barres [Whi87, Hen92, LM98], cette heuristique n'est ni nécessaire (cf. exemple 1) ni suffisante (cf. exemple 2) pour assurer qu'un système peut être remplacé par un repère local. C'est pourquoi nous lui préférons le critère de taille minimum des blocs à résoudre, penchant en faveur de l'utilisation de l'algorithme Minimum_Dense.

Exemple 1 *Soit un système géométrique en 3D constitué de 2 droites liées par 2 contraintes : une contrainte d'angle et une contrainte de distance*[5]*. Le graphe objets-contraintes correspon-*

[5] Une contrainte d'angle entre deux droites en 3D se calcule à partir de l'angle entre leurs vecteurs directeurs; une contrainte de distance entre droites en 3D établit la distance des deux points les plus proches de deux droites non-sécantes.

dant est constitué de 2 sommets de poids 4, liés par 2 arêtes de poids 1. Ce sous-système est bien 6-dense, structurellement rigide et rigide en 3D. De plus, ce système peut être remplacé par un repère local : il suffit de prendre le vecteur directeur de l'une des droites, le vecteur basé sur les points les plus proches de ces 2 droites et le produit vectoriel de ces 2 premiers vecteurs pour obtenir un repère local valide pour remplacer ce système. Cependant, ce sous-système ne vérifie par l'heuristique proposée pour assurer qu'il est bien remplaçable par un repère local : il ne contient que 2 objets au lieu des 4 préconisés par l'heuristique pour un système en 3D.

Exemple 2 *Soit le système géométrique suivant en 3D :*

- *Objets : 3 Points* (P_1, P_2, P_3) *et 1 droite* (L)
- *Contraintes : 3 Incidences* ($P_1 \in L, P_2 \in L, P_3 \in L$) *et 2 Distances* ($||P_1P_2|| = d_1, ||P_1P_3|| = d_2$)

Ce système est bien rigide en 3D. Cependant, il ne peut être remplacé par un repère local. En effet, il existe une infinité de repères locaux dans lesquels tous ces objets ont les mêmes coordonnées locales : il suffit que l'un des vecteurs du repère local soit colinéaire avec la droite L. Pourtant, ce sous-système vérifie l'heuristique puisque le système comporte bien 4 objets en 3D.

5 Conclusion

De façon générale, détecter un sous-graphe dense de taille minimum est un problème NP-difficile (MD). Cependant, cet article présente une restriction définissant une sous-classe polynomiale de ce problème ($\mathtt{MD}_{nsd}$). Cette restriction, qui impose que le graphe complet ne contienne aucun sous-graphe sur-dense, peut aussi être vérifiée en temps polynomial. Ces résultats devraient encourager la communauté des contraintes géométriques à utiliser l'algorithme `Minimum_Dense`, particulièrement pour améliorer les méthodes de rigidification récursive existantes.

Enfin, cet article a illustré l'étroite relation entre les problèmes Minimal Dense et Minimum Dense, et l'importance de notre restriction pour l'obtention de sous-classes polynomiales.

References

[AMO93] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows.* Prentice-Hall, 1993.

[BFH+95] William Bouma, Ioannis Fudos, Christoph Hoffmann, Jiazhen Cai, and Robert Paige. Geometric constraint solver. *Computer Aided Design*, 27(6):487–501, 1995.

[DMS98] Jean-François Dufourd, Pascal Mathis, and Pascal Schreck. Geometric construction by assembling subfigures. *Artificial Intelligence*, 99:73–119, 1998.

[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.

[HLS98] Christoph Hoffmann, Andrew Lomonosov, and Meera Sitharam. Geometric constraint decomposition. In B. Brüderlin and D. Roller, editors, *Geometric Constraint Solving and Applications*, pages 170–195. Springer, 1998.

[HLS00] Christoph Hoffmann, Andrew Lomonosov, and Meera Sitharam. Decomposition plans for geometric constraint systems. In *Proc. J. Symbolic Computation 2000*, 2000.

[HV95] C. M. Hoffmann and P. J. Vermeer. A spatial constraint problem. In J.-P. Merlet and B. Ravani, editors, *Computational Kinematics'95*, pages 83–92. Kluwer Academic Publishers, 1995.

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

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

[Laz81] D. Lazard. Résolution des systèmes d'équations algébriques. *Theoretical Computer Science*, 15:77–110, 1981.

[LM98] Hervé Lamure and Dominique Michelucci. Qualitative study of geometric constraints. In Beat Bruderlin and Dieter Roller, editors, *Geometric Constraint Solving and Applications*, pages 234–258. Springer, 1998.

[Pap94] Christos H. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.

[RM96] O. E. Ruiz and Ferreira P. M. Algebraic geometry and group theory in geometric constraint satisfaction for computer-aided design and assembly planning. *IIE Transactions on Design and Manufacturing*, 28:281–294, 1996.

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