



Temática: IV Taller Internacional de Matemática Computacional y Bioinformática.

Breve revisión sobre Resolución de Restricciones Geométricas

Brief overview on geometric constraints solving

Angel Alberto Vazquez Sánchez^{1*}, Augusto Cesar Rodríguez Medina², Lisset Salazar Gómez³

¹ Universidad de las Ciencias Informáticas. aavazquez@uci.cu

² Universidad de las Ciencias Informáticas. augusto@uci.cu

³ Universidad de las Ciencias Informáticas. lsgomez@uci.cu

* Autor para correspondencia: aavazquez@uci.cu

Resumen

Usando los software de diseño paramétrico asistido por computadora (CAD) los diseñadores pueden crear modelos geométricos que son actualizados a través de la modificación de los valores de los parámetros de control. Una de las vías en la que se controla la geometría con estos parámetros de control es mediante la Resolución de Restricciones Geométricas (GCS). La resolución de restricciones geométricas es fundamental para resolver un problema de restricción geométrica. Estos problemas aparecen en diferentes áreas como la ingeniería mecánica, modelado químico molecular, visión por computadora, diseño asistido por computadora, localización en redes de sensores inalámbricos, análisis de tolerancia, geometría dinámica, realidad virtual y robótica. En el presente trabajo se realiza una breve revisión de los métodos empleados para realizar la Resolución de Restricciones Geométricas, entre los que se encuentran los enfoques algebraicos, basados en grafos, basados en lógica, geometría dinámica y métodos evolutivos. Además, se realiza la descripción de diferentes softwares y marcos de trabajo que implementan diferentes enfoques y métodos para resolver estos tipos de problemas.

Palabras clave: Resolución de Restricciones Geométricas, Restricciones Geométricas, Grafo de Restricciones, Solver

Abstract

Using parametric computer-aided design (CAD) software, designers can create geometric models that are updated by modifying the values of control parameters. One of the ways in which geometry is controlled with these control parameters is through Geometric Constraint Solving (GCS). GCS is fundamental to solving a geometric constraint problem. These problems appear in different areas such as mechanical engineering, molecular chemical modeling, computer vision, computer aided design, localization in wireless sensor networks, tolerance analysis, dynamic geometry, virtual reality and robotics. In this paper, a brief review of the methods used to perform Geometric Constraint Resolution is made, among which are algebraic, graph-based, logic-based, dynamic geometry and evolutionary approaches. In addition, a description of different software and frameworks that implement different approaches and methods to solve these types of problems is made.

Keywords: Geometric Constraint Solving, Geometric Constraints, Constraint Graph, Solver



Introducción

En la segunda mitad de la década de los 80 del pasado siglo, se introdujeron técnicas revolucionarias para automatizar los procesos de diseño; surgieron así los paradigmas del Diseño Paramétrico y del historial basado en rasgos (History Based Feature). En esencia, se introdujo en los sistemas de diseño un elemento para la resolución de problemas geométricos con restricciones; en la literatura especializada esta pieza de programa ha recibido la denominación de **Sistema para la resolución de restricciones geométricas** (Geometric Constraint Solver); con frecuencia se refieren a los problemas que resuelven como **Sistemas de restricciones geométricas** (Geometric Constraint System) y el acrónimo en inglés es GCS.

A los diseñadores les fue posible entonces, realizar un esbozo aproximado de la idea que tenían acerca de un objeto sin preocuparse de la posición exacta de los elementos geométricos, pues podían modificar la misma imponiendo restricciones hasta lograr una representación totalmente definida que se ajustara a la forma deseada. Fue posible además, actualizar automáticamente un diseño existente, luego de modificar la configuración del esbozo mediante la variación de las restricciones impuestas. Autores como (Ait-Aoudia et al., 2009) refieren que “...El modelado geométrico mediante restricciones permite describir formas a los usuarios mediante la especificación de un boceto y la adición a este de restricciones geométricas...”

Para empezar a comprender la base teórica que soporta a tales sistemas, ilustramos el problema con un ejemplo sencillo en \mathbb{R}^2 que representa el perfil de una pieza. Supongamos que para realizarlo el diseñador esboza un esquema aproximado del perfil como se muestra en la Figura 1. Seguidamente, le aplica las restricciones hasta conformar el perfil deseado (véase Figura 2). Es precisamente el GCS, el mecanismo empleado para que el programa de diseño pueda ubicar los elementos geométricos en posiciones que satisfagan las condiciones de las restricciones impuestas, liberando al diseñador de esa tarea.

La *Resolución de Restricciones Geométricas* es un problema con aplicaciones en múltiples áreas como la ingeniería mecánica, modelado químico molecular, visión por computadora, diseño asistido por computadora, localización en redes de sensores inalámbricos, análisis de tolerancia, geometría dinámica, realidad virtual y robótica (Ait-Aoudia et al., 2009; HOFFMANN & VERMEER, 1995; Moussaoui, 2016; Hua Yuan et al., 2007).

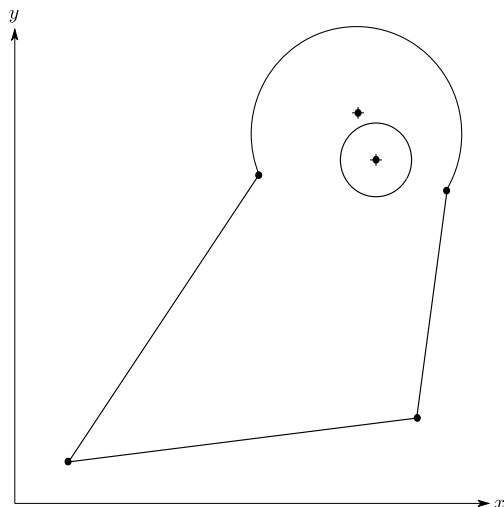


Figura 1: Croquis aproximado del perfil de una pieza sin restricciones geométricas

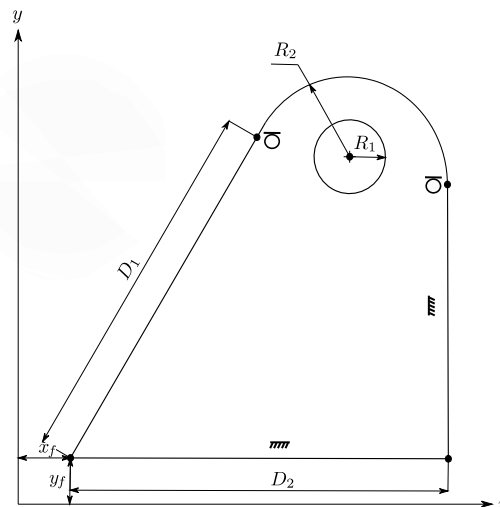


Figura 2: Perfil resultante de la aplicación de restricciones geométricas al croquis de la Figura 1.

Un *Sistema de Restricciones Geométricas (GCS)* consiste en un conjunto finito de elementos geométricos junto con relaciones de diferentes tipos llamados restricciones. En la geometría plana (espacio 2D), podemos citar, por ejemplo, puntos, líneas y círculos, restringidos con distancia, ángulo, incidencia y paralelismo. En el modelado de sólidos (espacio 3D), los elementos geométricos pueden ser puntos, líneas, planos, cilindros y esferas (Moussaoui, 2016). Una *realización* (o *solución*) de una GCS es una colocación (o configuración) de las primitivas geométricos que satisface las restricciones del GCS (Sitharam et al., 2018). Un *Problema de Restricciones Geométricas (GCP)* puede ser visto como un conjunto de objetos geométricos que se encuentran relacionados entre sí (a estas relaciones se les llama restricciones) en un espacio geométrico (normalmente Euclideo).

En (Hoffmann & Joan-Arinyo, 2005) describen este problema como una tupla (E, O, X, C) donde E es el espacio geométrico, O es el conjunto de entidades geométricas que definen el problema, X es el conjunto de variables cuyos valores deben ser determinados y C es el conjunto de restricciones de ese sistema. Otra definición de este problema es brindada por (Sun et al., 2018) como $GCP=(P, C)$, en el que $P=(p_1, p_2, \dots, p_n)$ es un conjunto de primitivas geométricas y $C=(c_1, c_2, \dots, c_m)$ es un conjunto de restricciones geométricas

Un problema puede ser categorizado según la cantidad de soluciones posibles: está *bien restringido* (*well-constrained*) si existe un número finito de soluciones al problema; mientras que un problema con infinitas soluciones es *infra-restringido* (*underconstrained*). Un problema está *super-restringido* (*overconstrained*) si una restricción puede ser eliminada y todavía el sistema de restricciones tiene un número finito de soluciones (Hoffmann & Joan-Arinyo, 2005; Sitharam et al., 2018).

En la siguiente sección se describen los principales métodos para resolver restricciones geométricas.

Principales métodos para resolver restricciones geométricas

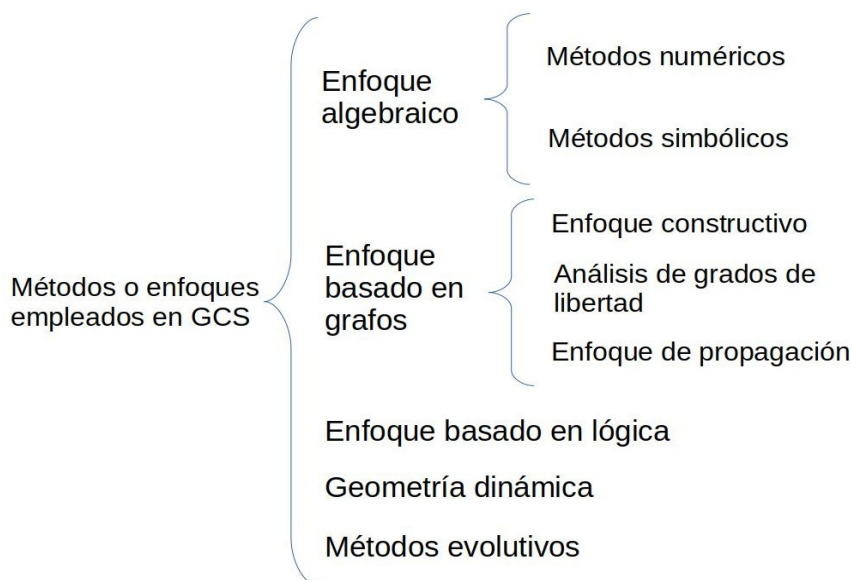


Figura 3: Enfoques en la resolución de restricciones geométricas.

Los métodos de GCS pueden ser clasificados a grandes rasgos como algebraicos(Ait-Aoudia et al., 2009), basados en grafos(Fudos & Hoffmann, 1997), o basados en la lógica(Joan-Arinyo & Soto, 1997); aunque en los últimos tiempos han aparecido nuevos enfoques a tener en cuenta. En (Ait-Aoudia et al., 2009; Bettig & Hoffmann, 2011; Hoffmann & Joan-Arinyo, 2005; Michelucci et al., 2006) se realizan estudios de los principales enfoques para la GCS(Figura 3). En las siguientes secciones del presente artículo se realiza una breve revisión de los diferentes enfoques para la GCS.

Enfoques algebraicos

En este enfoque el problema se describe mediante ecuaciones, que pueden ser lineales o cuadráticas y representan las restricciones aplicadas; las variables son las coordenadas de los elementos geométricos. La ventaja principal de este



enfoque es su completitud y su independencia dimensional. Una dificultad en este enfoque es que el sistema de ecuaciones es difícil de descomponer en subproblemas y que una solución completa tiene un costo computacional elevado. Sin embargo, sistemas pequeños derivan en muchos de los enfoques de solución y son resueltos rutinariamente (Bettig & Hoffmann, 2011; Michelucci et al., 2006). Los métodos algebraicos manejan fácilmente el caso 3D, pero en (Fudos & Hoffmann, 1997) se indica la necesidad de trabajar en alternativas para acelerar esa realización.

La forma de solucionar los sistemas derivados del enfoque algebraico es mediante métodos simbólicos o numéricos.

Métodos simbólicos

Solucionadores de ecuaciones generales emplean técnicas simbólicas tales como las bases Gröbner (Chyzak & Dumas, 2020) o el método de Wu-Ritt (Gallo & Mishra, 1991) para triangular el sistema de ecuaciones. En (Buchanan & de Pennington, 1993) se describe un solver construido sobre el algoritmo de Buchberger (Perry, 2017). También se reporta el uso de un método simbólico en (Kondo, 1992). En (Fudos & Hoffmann, 1997) se plantea que los métodos simbólicos suelen ser exponenciales en tiempo y espacio. Por tanto, solo pueden utilizarse para sistemas pequeños.

Métodos numéricos

Los métodos numéricos son el enfoque más antiguo para resolver restricciones. Este enfoque resuelve sistemas de ecuaciones grandes de manera iterativa. Los métodos como la iteración de Newton tienen buenos resultados si una solución intencionada puede ser provista al sistema y este no está mal condicionado. De modo que, si el punto inicial es tomado de la maqueta del usuario, entonces la maqueta debería estar cerca de la solución deseada. Sistemas no lineales tienen múltiples soluciones, pero los métodos numéricos pueden encontrar solo uno y puede no ofrecer control sobre las soluciones en las que el usuario está interesado (Bettig & Hoffmann, 2011; Johansson, 2019).

El método más utilizado es la iteración de Newton-Raphson (Light & Gossard, 1982; Lin et al., 1981; Nelson, 1985). Este es un método local y converge con mucha rapidez. Este método no se aplica para sistemas de ecuaciones consistentemente super-restringidos a menos que pasos especiales sean tomados, tales como resolver un problema de mínimos cuadrados.

La continuación homotópica (Allgower & Georg, 1993) es una familia de métodos que son globales y garantizan la convergencia. Ellos son exhaustivos y permiten determinar todas las soluciones de un problema de restricciones. Su



eficiencia es peor que el de Newton-Raphson (Bettig & Hoffmann, 2011). En (y Durand, 1998; Lamure & Michelucci, 1996) se aplica este método para la resolución de restricciones geométricas.

Los métodos numéricos son $O(n^3)$ o peores. Estos métodos adolecen de la falta de "explicación geométrica" durante el proceso de resolución. Además, la mayoría de los métodos numéricos tienen dificultades para manejar esquemas súper-restringidos o infra-restringidos. La ventaja de estos métodos es que tienen el potencial de resolver grandes sistemas no lineales que no pueda ser resuelto con ninguno de los otros métodos (Fudos & Hoffmann, 1997).

Enfoques basados en grafos

En el enfoque basado en grafos, el GCP es trasladado a un grafo (o hiper-grafo) cuyos vértices representan los elementos geométricos y las aristas las restricciones entre ellos. En otras palabras, un grafo no dirigido $G=(V, E)$ donde $|V|=n$ y $|E|=m$ representa el problema de restricción.

En la Figura 4 se muestra un ejemplo donde el problema de restricciones (parte izquierda de la figura) es procesado como un grafo de restricciones. El solver analiza el grafo y formula una estrategia de solución donde los subproblemas están aislados y sus soluciones se combinan apropiadamente. Una fase subsecuente entonces resuelve todos los subproblemas y los combina. La ventaja de este tipo de solver es que los subproblemas a menudo son muy pequeños y caen a categorías más simples. La desventaja es que análisis de grafo de un solver plenamente competente es muy complicado (Hoffmann & Joan-Arinyo, 2005).

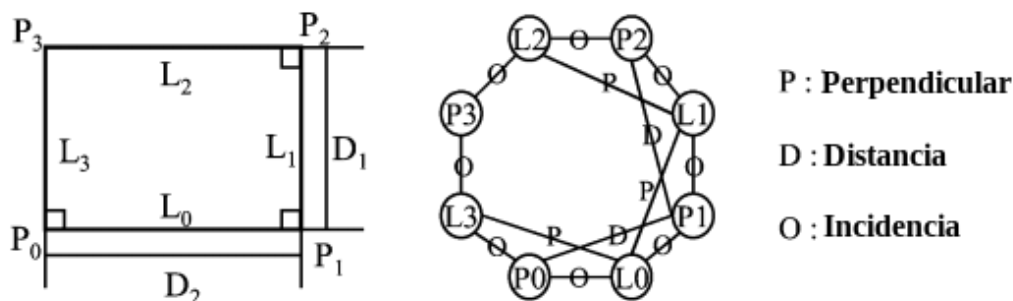


Figura 4: Croquis de primitivas geométricas con sus restricciones (izquierda) y el grafo de restricciones correspondiente (derecha). Fuente: (Lee et al., 2003)

El enfoque basado en grafos pueden distinguirse tres ramas principales: *enfoque constructivo*, análisis de *grados de libertad* y métodos de *propagación* (Bettig & Hoffmann, 2011).

Enfoque constructivo

En este enfoque, el grafo de restricciones se descompone y recombina para extraer los pasos básicos de construcción que deben resolverse. Una segunda fase elabora estos pasos, empleando métodos algebraicos y/o numéricos.

Primero se forma un número de cuerpos rígidos con tres grados de libertad, llamados clusters. Tres clusters pueden combinarse en un único cluster si comparten un mismo elemento geométrico. Geométricamente, la combinación corresponde a la colocación de los objetos geométricos asociados entre sí de manera que se puedan satisfacer las restricciones dadas (Fudos & Hoffmann, 1997).

El método de resolución de restricciones funciona en las siguientes fases conceptuales (Fudos & Hoffmann, 1997):

- Fase 1 (fase de análisis): Se analiza el grafo de restricciones y se estipula una secuencia de construcciones. Cada paso de esta secuencia corresponde a la colocación de tres cuerpos geométricos rígidos (clusters) que comparten un elemento geométrico (punto o línea). Esta fase consiste en dos partes:
 - El *análisis de reducción* que produce una secuencia de clusters locales y maneja los problemas bien restringidos y súper restringidos, y
 - El *análisis de descomposición* que produce una secuencia de descomposiciones (que corresponden a una secuencia reversa de uniones de clústeres) y maneja los casos con pocas restricciones. El resultado del análisis de reducción se introduce en el análisis de descomposición
- Fase 2 (fase de construcción): La construcción real de los elementos geométricos se lleva a cabo, en el orden determinado por la fase 1, resolviendo determinados conjuntos estándar de ecuaciones algebraicas.

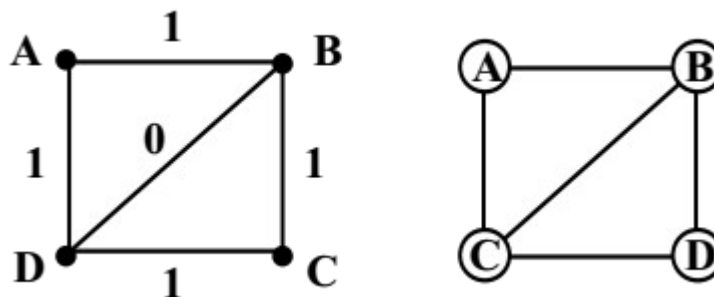


Figura 5: Un problema numéricamente infra-restringido (izquierda) y su grafo de restricciones asociado (derecha). Fuente: (Ait-Aoudia et al., 2009)



Este enfoque usa solamente la estructura del grafo de restricciones y olvida la información numérica, por lo que el grafo de restricciones puede estar estructuralmente bien restringido, pero numéricamente infra-restringido (Ait-Aoudia et al., 2009).

Análisis de grados de libertad

Los vértices del grafo son etiquetados con el número de grados de libertad del objeto geométrico que representan. En 2D, un punto puede tener dos grados de libertad, un círculo 3. Cada arista es etiquetada con los grados de libertad cancelados con la restricción representada. Si los vértices incidentes son puntos en 2D, por ejemplo, una restricción incidente cancela dos grados de libertad, una restricción de distancia cancela un grado de libertad. Este grafo es analizado por una estrategia de solución (Bettig & Hoffmann, 2011).

En (Fudos et al., 2016) se definen los grados de libertad como: el número de variables independientes con las que se puede instanciar y posicionar un objeto geométrico. Un cuerpo asociado a un sistema de ejes de coordenadas cartesiano tiene tres grados de libertad, dos traslaciones en las direcciones x e y respectivamente y una rotación; así mismo podemos considerar el posicionamiento de los elementos geométricos que conforman el perfil de una pieza, que según su tipo requieren cierta cantidad de variables para ser definidos; la Tabla 1 muestra los grados de libertad asociados a diferentes elementos geométricos en el plano.

Tabla 1: Grados de libertad de objetos elementales y cuerpos rígidos. Fuente: (Fudos et al., 2016)

Elemento	Significado geométrico de DoF	2D	3D
Punto	Coordenadas (x,y)	2	3
Línea	2D: distancia desde el origen y dirección 3D: distancia desde el origen, dirección en 3D dirección en el plano	2	4
Segmento	Coordenadas del punto inicial y final (x_1,y_1) y (x_2,y_2)	4	6
Arco de círculo	Coordenadas del centro (x_c,y_c) , el radio R y una coordenada x o y para fijar los extremos del arco	5	
Plano	Distancia desde el origen, dirección en 3D		3
Círculo, radio fijo	Coordenadas del centro (x,y) y una distancia R para	2	5

	el radio		
Círculo, radio variable	Coordenadas de centro, radio y orientación en 3D	3	6
Cuerpo rígido	2D: 2 desplazamientos, 1 rotación 3D: 3 desplazamientos, 3 rotaciones	3	6
Esfera, de radio fijo o variable	3 desplazamientos o 3 desplazamientos, radio	3	4
Elipse, ejes variables	centro, longitudes de los ejes, orientación de los ejes	5	7
Elipsoide, ejes variables	centro, longitudes de los ejes, orientación de los ejes	5	7

Las restricciones geométricas consumen uno o más grados de libertad y se expresan mediante un número igual de ecuaciones independientes. Un conjunto básico de restricciones geométricas en el plano se resume en la Tabla 2.

Tabla 2: Tipos de restricciones y los grados de libertad que eliminan. Fuente: (Fudos et al., 2016)

Tipo	Restricción	2D	3D
Distancia punto-punto	Una ecuación representando la distancia entre dos puntos p_1, p_2 bajo la métrica $ $: $ p_1 - p_2 = d$ con $d > 0$	1	1
Ángulo entre líneas y planos	Ángulo entre dos líneas (puede representarse por el ángulo entre los vectores normales). Excepciones en 3D: el paralelismo entre líneas elimina 2 DoF. La ortogonalidad línea-plano elimina 2 DoF	1	1
Punto sobre punto	Para cualquier métrica $ p_1 - p_2 = 0$ es equivalente a $p_1 = p_2$.	2	3
Distancia punto-línea	Una ecuación representando la distancia punto-línea	1	1
Distancia paralela línea-línea	Paralelismo y distancia	2	3
Distancia paralela plano-plano	Paralelismo y distancia	2	3
Punto sobre línea	Igual que la distancia punto-línea. En 3D la dimensión es reducida.	1	2
Línea sobre línea	Igual que en distancia paralela entre líneas en 2D. En 3D un grado de	2	4

	libertad adicional es cancelado.		
Punto sobre plano	Igual que distancia punto-plano	-	1
Punto sobre plano	Igual que la distancia punto-plano	-	1
Línea sobre plano	Paralelismo plano-línea y distancia cero	-	2
Fijación de objeto elemental	Fijar todos o algunos de los grados de libertad de un objeto elemental.	DoF	DoF

Con este tipo de análisis se puede determinar cuando un problema de restricciones geométricas es super-restringido o infra-restringido, por ejemplo se usa este enfoque en (Krammer, 1992, 1991). Un método de solución simbólico es derivado usando reglas que tienen un significado geométrico. En (Hsu & Brüderlin, 1997) se resuelve el problema de las restricciones en dos fases, generando primero una representación de reglas simbólicas, seguido por elaborar esas reglas mediante su resolución. Si el razonamiento geométrico falla, entonces se intenta un método numérico. En (Latham & Middleditch, 1996) se descompone el grafo en componentes mínimamente conectados que llaman *conjuntos balanceados*. Si un conjunto balanceado está en un conjunto predefinido de patrones, entonces el subproblema es resuelto por una construcción geométrica, si no, se intenta una solución numérica. Este método también trata con restricciones simbólicas e identifica problemas infra- y super- restringidos. Los problemas super-restringidos son abordados priorizando las restricciones dadas.

Enfoques de propagación

Estos métodos codifican el problema de restricciones mediante un grafo en el que los vértices representan variables y ecuaciones y las aristas están etiquetadas con las ocurrencias de las variables en las ecuaciones. La propagación trata de orientar las aristas del grafo de forma que todas las aristas que inciden en un vértice de ecuación sean aristas entrantes excepto una. Si se consigue, el sistema de ecuaciones se ha triangularizado (Bettig & Hoffmann, 2011). Los algoritmos de orientación incluyen la propagación de grados de libertad y la propagación de valores conocidos, por ejemplo, (Freeman-Benson et al., 1990). El método falla cuando la orientación crea bucles, por lo que los algoritmos incluyen técnicas para romper los bucles y pueden recurrir a solucionadores numéricos. En (Borning et al., 1996) describe un algoritmo de propagación local que puede tratar con desigualdades.



Enfoque basados en lógica

En este enfoque, el problema de las restricciones se traduce en un conjunto de afirmaciones y axiomas geométricos. Aplicando el razonamiento geométrico, esta representación se transforma de manera que los pasos específicos de la solución se hacen explícitos. El sistema dispone de un conjunto de pasos de construcción que se resuelve asignando los valores de coordenadas adecuados a las entidades geométricas (Bettig & Hoffmann, 2011).

En (Aldefeld, 1988; Brüderlin & others, 1990; Sohrt & Brüderlin, 1991; Yamaguchi & Kimura, 1990) utilizan la lógica de primer orden para obtener información geométrica aplicando un conjunto de axiomas de la geometría de Hilbert. Estos métodos producen localizaciones geométricas donde los elementos deben estar. En (Sunde, 1987; y Verroust et al., 1992) consideran dos tipos de restricciones: conjuntos de puntos situados con respecto a un sistema local de coordenadas y conjuntos de segmentos cuyas direcciones son fijas. El razonamiento se realiza básicamente mediante un sistema de reescritura sobre los conjuntos de restricciones. El problema se resuelve cuando todos los elementos geométricos pertenecen a un único conjunto. Luego en (y Freixas et al., 2010; Joan-Arinyo & Soto, 1997) se amplía el sistema de restricciones con un tercer tipo que consiste en conjuntos que contienen un punto y una recta, tal que la distancia perpendicular punto-línea es fija.

Los solucionadores basados en reglas se basan en la formulación de predicados. Aunque proporcionan un estudio cualitativo de las restricciones geométricas, la "enorme" cantidad de cálculos necesarios (búsqueda exhaustiva y comparación) los hacen inapropiados para las aplicaciones prácticas (Fudos & Hoffmann, 1997).

Geometría dinámica

Dado un sistema *infra-restringido*, podemos añadir restricciones para convertir el problema en *bien restringido*. Estas restricciones adicionales puede ser entendidas como parámetros cuando son dimensionales y varían el valor de estos, diferentes soluciones surgen las cuales pueden ser entendidas como una configuración geométrica dinámica. Un ejemplo sencillo puede ser un ensamble pistón-manivela. Sistemas como (Kortenkamp & Richter-Gebert, 2007) son diseñados para tratar con estos problemas. Varios trabajos han abordado estos problemas desde una perspectiva de resolución de restricciones, incluyendo (Freixas et al., 2010).



Métodos evolutivos

En este enfoque, el problema de restricciones geométricas es reinterpretado como un problema de optimización que es atacado usando métodos evolutivos y poblacionales como algoritmos genéticos, enjambres de partículas, entre otros (Chunhong et al., 2006; Gao et al., 2009; Sun et al., 2018; Ye, 2019; Yi et al., 2010; H Yuan et al., 2006; Hua Yuan et al., 2009). En estos métodos prometen incrementar la eficiencia y la confiabilidad de los GCS, poseyendo mejores propiedades de convergencia (Sun et al., 2018).

Situación actual

En la siguiente tabla se muestra un resumen de los principales métodos y enfoques que permiten solucionar un GCP.

Tabla 3: Comparación de los métodos o enfoques para la solución de restricciones geométricas.

Método o enfoque	Completo	Complejidad temporal	Aplicable en 2D	Aplicable en 3D
Algebraico	-	-	-	-
- métodos numéricos	No	$O(n^3)$	Si	Si
- métodos simbólicos	Si	$O(2^n)$	Si	Si
Basado en grafos	Si	$O(n^2)$	Si	Si
Basado en lógica	No	-	Si	-
Métodos evolutivos	No	$O(n \log(n))$	Si	Si

Para comprender mejor el contenido de la tabla debemos considerar lo siguiente:

- Los métodos numéricos poseen una complejidad de $O(n^3)$ y además carecen de “explicación geométrica”. Según (Ershov et al., 2003) este enfoque requiere de una buena aproximación inicial y no garantiza la convergencia a una solución. Para sistemas pequeños puede constituir una alternativa viable, pero tiene problemas para manejar sistemas con muchas piezas y esquemas super-restringidos o infra-restringidos. Por su parte, los métodos simbólicos poseen una complejidad temporal exponencial, por lo que solamente es factible aplicarlos en sistemas pequeños.
- En (Fudos & Hoffmann, 1997) se plantea que el enfoque basado en grafos constructivos posee una complejidad temporal de $O(n^2)$, aunque plantea que debería ser posible aumentar la velocidad hasta un $O(n \log(n))$. De igual manera se plantea en (Joan-Arinyo, 2009) que los diferentes métodos que siguen este



enfoque (constructivo, análisis de grados de libertad y el enfoque de propagación) poseen la complejidad temporal ya planteada ($O(n^2)$).

- En los enfoques basados en lógica se tiene como problema que las entidades geométricas se encuentran restringidos a tipos como puntos, líneas, vectores, y triángulos, mientras que las restricciones se encuentran sujetos a distancia entre puntos, ángulos entre líneas, entre otros. Lo que hace que este enfoque no sea completo, al no contemplar todas las posibilidades respecto a las entidades geométricas y a los tipos de restricciones entre ellos (HOFFMANN & VERMEER, 1995). En la mayoría de los artículos revisados no se plantea nada respecto a su complejidad temporal, sin embargo, en (Ait-Aoudia et al., 2009) se plantea que el número de computaciones necesarias para estos tipos de solver es “enorme” y carece de sentido práctico en sistemas reales, dado que se realiza búsqueda y emparejamientos exhaustivos. Además, este enfoque ha sido aplicado en escenarios 2D (Ait-Aoudia et al., 2009; Bruderlin, 1989; Verroust et al., 1992), por lo que no podemos asegurar su eficiencia en el escenario 3D.
- Existen diferentes tipos de métodos evolutivos que pueden ser aplicados al problema de restricciones geométricas, el más utilizado para este tipo de problema ha sido PSO. En (Sudholt, 2008) y en (Jansen & Neumann, 2011) se plantea que estos métodos pueden tener una complejidad temporal de $O(n \log(n))$. En la literatura revisada no se muestra explícitamente el cálculo de complejidad temporal de estos algoritmos cuando son usados para resolver el problema en cuestión (GCP) por lo que asumiremos la complejidad de estos métodos en general. En (Zhang et al., 2015) se plantea estos métodos tiene la tendencia a estancarse cerca de la solución óptima especialmente para problemas medianos y grandes. Por tanto, es posible encontrarnos instancias del problema donde usando este método no sea posible encontrar una solución factible.

Como podemos apreciar el enfoque basado en grafo es el que asegura una complejidad temporal aceptable mientras se asegura encontrar soluciones si existen.

Aunque en las últimas décadas se realizaron aportes que permitieron resolver problemas en dos dimensiones de forma factible y en tres dimensiones para el caso de piezas o cuerpos aislados, continúa siendo un desafío lidiar con ensambles en 3D con gran cantidad de componentes geométricos. Los métodos existentes presentan problemas de rendimientos por la complejidad en tiempo que poseen, en función de la cantidad de elementos y de restricciones geométricas aplicadas.



Con el desarrollo de los sistemas para el procesamiento paralelo o distribuido, resulta viable evaluar formas para utilizar los enfoques mencionados anteriormente en esas condiciones, con el objetivo de aumentar la eficiencia del proceso de resolución de restricciones para los casos de ensamblajes grandes.

Implementaciones conocidas de GCS

Existen diferentes implementaciones de GCS. Entre estos podemos listar los siguientes:

DCM (Dimensional Constraint Manager)

Es un solver comercial de D-Cubed (subsidiaria de Siemens PLM Software), integrado en [AutoCAD](#), [SolidWorks](#), [Creo](#) y otros sistemas CAD populares.

Según se explica en (Hoffmann, 2001): D-Cubed ha sido uno de los primeros pioneros en el enfoque algebraico de la resolución de restricciones geométricas para problemas 2D. DCM (Gestor de Restricciones Dimensionales), se ha convertido en una especie de estándar de la industria. Muy competente, este motor de resolución de restricciones se ha incorporado a muchos de los principales sistemas de CAD en los que permite realizar bocetos basados en restricciones.

D-Cubed 2D Dimensional Constraint Manager (2D DCM) es un solucionador de restricciones geométricas que permite técnicas de boceto intuitivas que pueden verse en muchas aplicaciones avanzadas de diseño 2D y 3D. Los usuarios crean y modifican bocetos en 2D de forma más eficiente aplicando cotas y restricciones geométricas que especifican y preservan con precisión la ubicación de las geometrías en un boceto en 2D.

Este solver posee una versión abierta disponible en [GitHub](#).

LGS

Es un solver comercial desarrollado por LEDAS y actualmente pertenece a Bricsys, integrado en Cimatron E y BricsCAD. En (Ershov et al., 2003) se describen algunas de las principales características de este solver. Este solver posee una arquitectura modular (Figura 6) que está basada en una representación interna que contiene toda la información acerca del modelo. LGS existe en forma de prototipo de solucionador 2D, pero las ideas en las que se basa LGS se aplican ahora en el solucionador 3D industrial desarrollado por LEDAS Ltd para una empresa líder

mundial en CAD. Está codificado en C++ teniendo una API al estilo C para una fácil integración con una amplia variedad de aplicaciones de software usando contenedores (*wrappers*) tales como .NET, Java y C++.

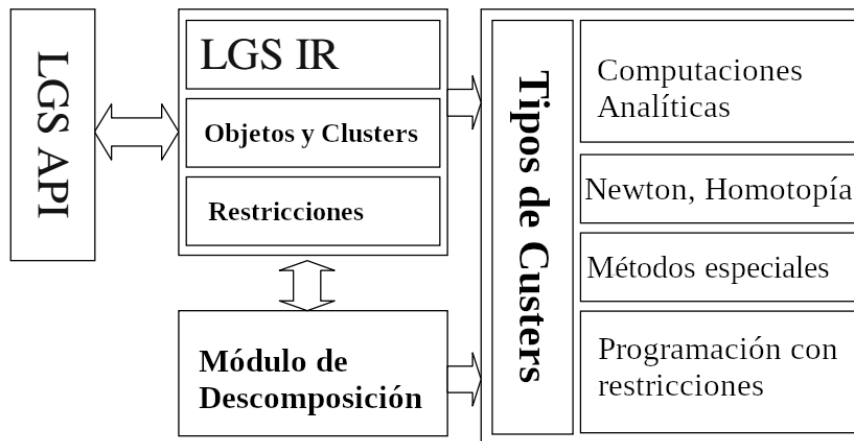


Figura 6: Arquitectura del solver LGS. Fuente: (Ershov et al., 2003)

C3D Solver

Solver disponible comercialmente que es una parte de [C3D Toolkit](#), integrado en KOMPAS-3D. El C3D Toolkit es un SDK responsable para la construcción, edición, visualización y conversión de modelos geométricos. Uno de los módulos de este SDK es el *C3D Solver* el cuál define dimensiones y restricciones para crear conexiones entre elementos geométricos en modelos 2D y 3D. C3D Solver analiza los grados de libertad disponibles y mantiene las restricciones a medida los cambios son aplicados a la geometría (*C3D Toolkit*, n.d.). Entre las capacidades de este solver se encuentran (*C3D Solver*, n.d.):

- es usado para las aplicaciones siguientes:
 - Creación de bocetos paramétricos 2D con administración de dimensiones y restricciones
 - Posicionamiento de los cuerpos en los ensamblajes mediante las relaciones de posición y las cotas en 3D
 - Reconstruir los modelos modificados manteniendo intactas las combinaciones definidas anteriormente
 - Modelación de mecanismos planos y espaciales
 - Creación de pipelines y esquemas 3D



- Este solver provee como principales restricciones dimensionales: distancia, distancia directa (solo en 2D), ángulos entre líneas y planos, y radios.
- Y por último provee las siguientes funcionalidades:
 - Creación y solución de restricciones paramétricas
 - Manipulación geométrica
 - Arrastrar geometrías
 - Satisfacer restricciones
 - Analizar grados de libertad (solamente en 2D)
 - Agrupar conjuntos rígidos (solamente en 3D)
 - Llamadas a la API de registro

GeoSolver

El paquete de Python GeoSolver provee clases y funciones para especificar, analizar y solucionar problemas de restricciones geométricas. Puede ser usado en aplicaciones de Python que necesiten resolver restricciones geométricas (der Meiden & Bronsvort, 2010). Entre sus principales características se encuentran:

- Problemas de restricciones geométricas en 3D, que consisten en:
 - variables de punto
 - restricción de distancia de dos puntos
 - restricción de ángulo de tres puntos
- Los problemas con otras variables geométricas (rectas, planos, esferas, etc.) y restricciones se pueden trasladar a estas restricciones básicas sobre variables puntuales
- Selección de soluciones:
 - restricciones de espiralidad (restricciones de reloj y de mano)
 - selección basada en prototipos (basada en bocetos)

- Encuentra una solución genérica, a partir de la cual se pueden derivar fácilmente todas las soluciones particulares, para diferentes valores de parámetros (por ejemplo, distancias y ángulos)
- Algoritmo incremental que puede tratar eficazmente los cambios en el GCP (es decir, añadir y eliminar restricciones y variables y cambiar los valores de los parámetros)
- Marco de resolución extensible, que permite nuevos tipos de variables y restricciones geométricas

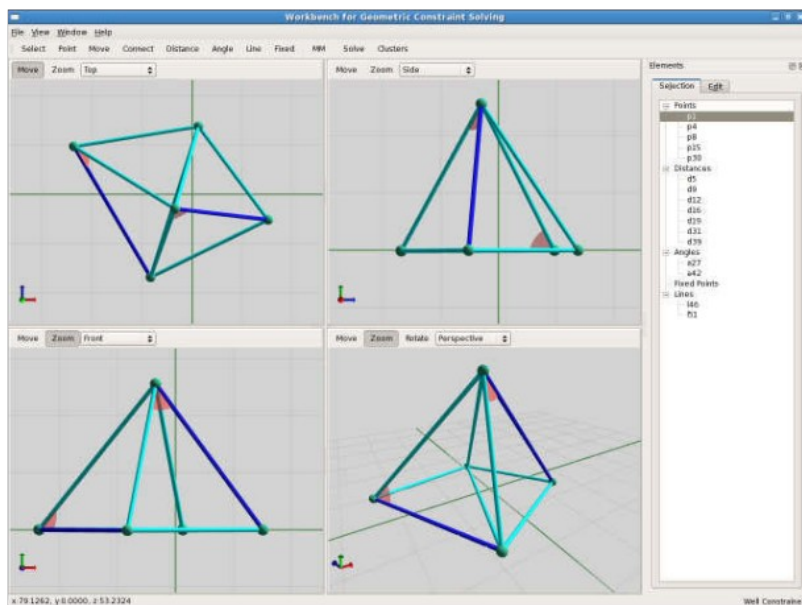


Figura 7: Vista de un área de trabajo de Geosolver. Fuente: (de Regt et al., 2008)

Este GCS posee un Workbench que provee una elegante GUI para la edición, análisis y solución interactiva de problemas de restricciones geométricas (Figura 7). Utiliza el paquete GeoSolver para el análisis y la resolución, y PyQt y pyOpenGL para los gráficos 3D interactivos. El workbench se utiliza actualmente para depurar/probar el paquete GeoSolver. También puede utilizarse como herramienta de enseñanza, y tal vez algún día pueda convertirse en un kit completo de herramientas de diseño geométrico (de Regt et al., 2008).

Entre las características del área de trabajo se encuentran:

- Fácil edición interactiva de los problemas de restricciones
- Vistas en 3D del problema y la solución
- Vista de descomposición: muestra cómo se descompone el problema en un árbol de clusters rígidos

FRONTIER

Este motor de restricciones geométricas Frontier, es diseñado para abordar las principales razones de la infrutilización de las restricciones geométricas en los sistemas actuales de diseño y ensamblaje en 3D (Oung et al., 2001).

Muchos de los puntos fuertes de FRONTIER se basan en el método de descomposición y recombinación basado en el grafo de grado de libertad denominado Algoritmo de Vértices de la Frontera (FA) para sistemas de restricciones geométricas. Este algoritmo se aplica también a la 3D, aunque la interfaz de usuario de FRONTIER se limita a la 2D. Las ventajas de FA, su rendimiento, su construcción y su comparación con los métodos de descomposición-recombinación existentes pueden encontrarse en los siguientes documentos.

Entre las principales ventajas de este GCS se encuentran:

- Frontier permite tanto (a) el uso de estructuras complejas, cíclicas y con restricciones espaciales como (b) el diseño basado en características.
- Frontier emplea una representación crucial de los subsistemas o clusters del plan DR (descomposición y recombinación), su jerarquía y su interacción (Hoffman et al., 2001).
- Frontier utiliza un lenguaje de representación de restricciones frep que permite que todos sus componentes compartan eficazmente las mismas estructuras de datos.

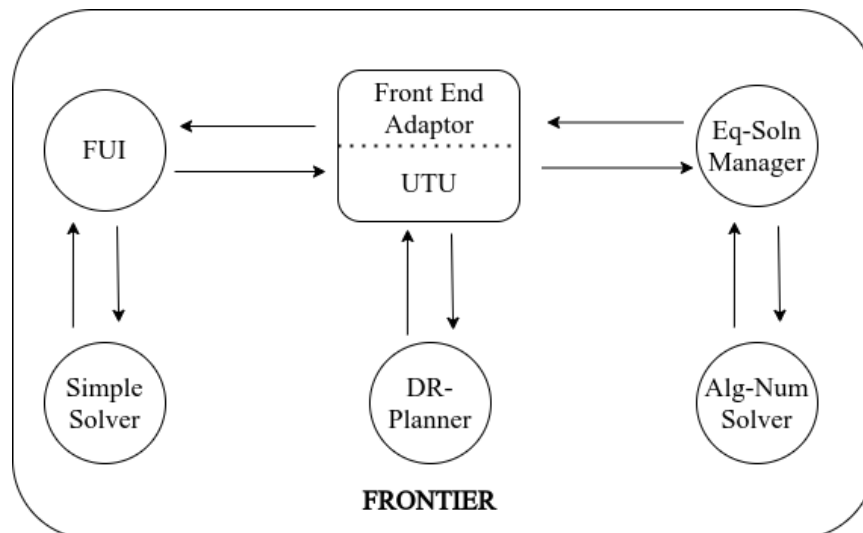


Figura 8: Organización de los módulos de FRONTIER. Fuente: (Oung et al., 2001)



- La implementación y organización interna de los componentes de Frontier (Figura 8), coordinada por la Unidad de Transferencia Universal (UTU), satisface dos requisitos que compiten entre sí: la comunicación sin fisuras entre los componentes, así como la completa portabilidad de cada uno de ellos.

Está implementado principalmente en Java y C++. El código de este solver se encuentra abierto en [Github](#).

Comparativa

En la siguiente tabla se realiza una comparación entre los diferentes solver vistos.

Tabla 4: Comparativa entre los diferentes solvers estudiados.

Nombre	Lenguaje de programación	Enfoque empleado	Arquitectura	2D	3D
DCM	C++	Basada en grafos	-	Si	No
LGS	C++	Basada en grafos	Modular	Si	Si
C3D Solver	C++	Basada en grafos	Software Development Kit (SDK)	Si	Si
GeoSolver	Python	Basada en grafos	Modular	Si	Si
Frontier	C++ y Java	Basada en grafos	Modular	Si	Si

Como se puede apreciar la tendencia principal es a desarrollar los solver usando el enfoque basado en grafos, implementando una arquitectura modular con el lenguaje C++, así mismo, en su mayoría implementan el solver para problemas en 2D y 3D.

Conclusiones

Durante las últimas décadas, los principales métodos para la resolución de restricciones geométricas han tenido poca variación, por lo que los avances principales han consistido en ampliar los tipos de objetos y restricciones que se reconocen en los solucionadores. El enfoque que por sus características resuelve de mejor manera el problema de las restricciones geométricas es el basado en grafos; y por tanto, es el enfoque más difundido, aunque se han usado otros enfoques como los métodos numéricos, basados en lógica, simbólicos. Una opción viable para poder elevar el

rendimiento en el proceso de resolución de restricciones en el caso de ensamblajes grandes en 3D es implementar variantes para el procesamiento distribuido o en paralelo de los métodos existentes.

Existen varias implementaciones comerciales y académicas de solver que se encuentran disponibles. Varios son de código abierto o al menos poseen una versión menor abierta. La mayoría de estos solvers se enfocan en el problema en 2D, aunque algunos poseen las capacidades para tratar ensamblajes en 3D. El principal método de resolución de restricciones geométricas implementado es el enfoque basado en grafos. Son implementados en diferentes lenguajes de programación donde el que predomina es C++ y generalmente poseen una arquitectura modular. No se describen capacidades de procesamiento en paralelo o distribuido en estos solvers, por lo que esto puede constituir un espacio de mejora para el rendimiento de estos sistemas en el futuro.

References

- Ahn, Y. J., Hoffmann, C. M., & Rosen, P. (2011). Length and energy of quadratic Bézier curves and applications. *To Appear*.
- Ait-Aoudia, S., Bahriz, M., & Salhi, L. (2009). 2D geometric constraint solving: An overview. *Proceedings - 2009 2nd International Conference in Visualisation, VIZ 2009*, 201–206. <https://doi.org/10.1109/VIZ.2009.29>
- Aldefeld, B. (1988). Variation of geometries based on a geometric-reasoning method. *Computer-Aided Design*, 20(3), 117–126.
- Allgower, E. L., & Georg, K. (1993). Continuation and path following. *Acta Numerica*, 2(1), 1–64.
- Bettig, B., & Hoffmann, C. M. (2011). Geometric constraint solving in parametric computer-aided design. *Journal of Computing and Information Science in Engineering*, 11(2). <https://doi.org/10.1115/1.3593408>
- Borning, A., Anderson, R., & Freeman-Benson, B. (1996). Indigo: A local propagation algorithm for inequality constraints. *Proceedings of the 9th Annual ACM Symposium on User Interface Software and Technology*, 129–136.
- Bruderlin, B. (1989). *Rule-based geometric modelling*.
- Brüderlin, B. D., & others. (1990). Symbolic computer geometry for computer aided geometric design. *Advances in Design and Manufacturing Systems, Tempe, AZ*.
- Buchanan, S. A., & de Pennington, A. (1993). Constraint definition system: a computer-algebra based approach to solving geometric-constraint problems. *Computer-Aided Design*, 25(12), 741–750.
- C3D Solver*. (n.d.). Retrieved May 9, 2021, from <https://c3dlabs.com/en/products/c3d-toolkit/solver/>



- C3D Toolkit*. (n.d.). Retrieved May 9, 2021, from <https://c3dlabs.com/en/products/c3d-toolkit/>
- Chunhong, C., Bin, Z., Limin, W., & Wenhui, L. (2006). The parametric design based on organizational evolutionary algorithm. *Pacific Rim International Conference on Artificial Intelligence*, 940–944.
- Chyzak, F., & Dumas, P. (2020). A Gröbner-basis theory for divide-and-conquer recurrences. *Proceedings of the 45th International Symposium on Symbolic and Algebraic Computation*, 99–106.
- de Regt, R., van der Meiden, H. A., & Bronsvort, W. F. (2008). A workbench for geometric constraint solving. *Computer-Aided Design and Applications*, 5(1–4), 471–482.
- der Meiden, H. A., & Bronsvort, W. F. (2010). A non-rigid cluster rewriting approach to solve systems of 3D geometric constraints. *Computer-Aided Design*, 42(1), 36–49.
- Durand, C. B. (1998). *Symbolic and numerical techniques for constraint solving*.
- Ershov, A., Ivanov, I., Preis, S., Rukoleev, E., & Ushakov, D. (2003). LGS: Geometric Constraint Solver. In M. Broy & A. V Zamulin (Eds.), *Perspectives of System Informatics* (pp. 423–430). Springer Berlin Heidelberg.
- Freeman-Benson, B. N., Maloney, J., & Borning, A. (1990). An incremental constraint solver. *Communications of the ACM*, 33(1), 54–63.
- Freixas, M., Joan-Arinyo, R., & Soto-Riera, A. (2010). A constraint-based dynamic geometry system. *Computer-Aided Design*, 42(2), 151–161.
- Fudos, I., & Hoffmann, C. M. (1997). A graph-constructive approach to solving systems of geometric constraints. *ACM Transactions on Graphics (TOG)*, 16(2), 179–216.
- Fudos, I., Hoffmann, C. M., & Joan-Arinyo, R. (2016). *Tree-decomposable and Underconstrained Geometric Constraint Problems*. 1–55. <http://arxiv.org/abs/1608.05205>
- Gallo, G., & Mishra, B. (1991). Wu-Ritt characteristic sets and their complexity. *Discrete and Computational Geometry: Papers from the DIMACS Special Year*, 6, 111–136.
- Gao, X., Sun, L., & Sun, D. (2009). Artificial Immune-Chaos Hybrid Algorithm for Geometric Constraint Solving. *Information Technology Journal*, 8. <https://doi.org/10.3923/itj.2009.360.365>
- Hoffman, C. M., Lomonosov, A., & Sitharam, M. (2001). Decomposition Plans for Geometric Constraint Systems, Part I: Performance Measures for CAD. *Journal of Symbolic Computation*, 31(4), 367–408. <https://doi.org/https://doi.org/10.1006/jsc0.2000.0402>
- Hoffmann, C. M. (2001). D-Cubed's Dimensional Constraint Manager. *Journal of Computing and Information Science in Engineering*, 1(1), 100–101. <https://doi.org/10.1115/1.1354994>



- Hoffmann, C. M., & Joan-Arinyo, R. (2005). A Brief on Constraint Solving. *Computer-Aided Design and Applications*, 2(5), 655–663. <https://doi.org/10.1080/16864360.2005.10738330>
- HOFFMANN, C. M., & VERMEER, P. J. (1995). Geometric Constraint Solving in \mathfrak{R}^2 and \mathfrak{R}^3 . In *Computing in Euclidean geometry* (pp. 266–298). World Scientific. https://doi.org/10.1142/9789812831699_0008
- Hsu, C. Y., & Brüderlin, B. (1997). A Hybrid Constraint Solver Using Exact and Iterative Geometric Constructions. In D. Roller & P. Brunet (Eds.), *Systems Development: Tools and Methods*. Springer-Verlag.
- Jansen, T., & Neumann, F. (2011). Computational complexity and evolutionary computation. *Proceedings of the 13th Annual Conference Companion on Genetic and Evolutionary Computation*, 1053–1080.
- Joan-Arinyo, R. (2009). Basics on geometric constraint solving. *Proceedings of 13th Encuentros de Geometría Computacional (EGC09), Zaragoza (Spain)*.
- Joan-Arinyo, R., & Soto, A. (1997). A correct rule-based geometric constraint solver. *Computers & Graphics*, 21(5), 599–609.
- Johansson, R. (2019). *Numerical Python: Scientific Computing and Data Science Applications with Numpy, SciPy and Matplotlib* (Second). Apress. <https://doi.org/https://doi.org/10.1007/978-1-4842-4246-9>
- Kondo, K. (1992). Algebraic method for manipulation of dimensional relationships in geometric models. *Computer-Aided Design*, 24(3), 141–147.
- Kortenkamp, U. H., & Richter-Gebert, J. (2007). *The interactive geometry software Cinderella. 2*. Springer-Verlag.
- Krammer, G. A. (1992). Solving Geometric Constraint Systems. MIT, Cambridge.
- Krammer, G. A. (1991). Using Degree of Freedom Analysis to Solve Geometric Constraint Systems. In J. Rossignac & J. Turner (Eds.), *Symposium on Solid Modeling Foundations and CAD/ CAM Applications*.
- Lamure, H., & Michelucci, D. (1996). Solving geometric constraints by homotopy. *IEEE Transactions on Visualization and Computer Graphics*, 2(1), 28–34.
- Latham, R., & Middleditch, A. (1996). Connectivity Analysis: A Tool for Processing Geometric Constraints. *Computer Aided Design and Manufacturing*.
- Lee, K.-Y., Kwon, O.-H., Lee, J.-Y., & Kim, T.-W. (2003). A hybrid approach to geometric constraint solving with graph analysis and reduction. *Advances in Engineering Software*, 34(2), 103–113.
- Light, R., & Gossard, D. (1982). Modification of geometric models through variational geometry. *Computer-Aided Design*, 14(4), 209–214.
- Lin, V. C., Gossard, D. C., & Light, R. A. (1981). Variational geometry in computer-aided design. *ACM SIGGRAPH Computer Graphics*, 15(3), 171–177.



- Michelucci, D., Foufou, S., Lamarque, L., & Schreck, P. (2006). Geometric Constraints Solving: Some tracks. *Proceedings SPM 2006 - ACM Symposium on Solid and Physical Modeling, 2006*, 185–196.
- Moussaoui, A. (2016). *Geometric Constraint Solver*.
- Nelson, G. (1985). Juno, a constraint-based graphics system. *Proceedings of the 12th Annual Conference on Computer Graphics and Interactive Techniques*, 235–243.
- Oung, J., Sitharam, M., Moro, B., & Arbree, A. (2001). FRONTIER: Fully Enabling Geometric Constraints for Feature-Based Modeling and Assembly. *Proceedings of the Sixth ACM Symposium on Solid Modeling and Applications*, 307–308. <https://doi.org/10.1145/376957.376995>
- Perry, J. (2017). Exploring the dynamic Buchberger algorithm. *Proceedings of the 2017 ACM on International Symposium on Symbolic and Algebraic Computation*, 365–372.
- Sitharam, M., John, A. St., & Sidman, J. (2018). Handbook of Geometric Constraint Systems Principles. In *Handbook of Geometric Constraint Systems Principles*. <https://doi.org/10.1201/9781315121116>
- Sohrt, W., & Brüderlin, B. D. (1991). Interaction with constraints in 3D modeling. *Proceedings of the First ACM Symposium on Solid Modeling Foundations and CAD/CAM Applications*, 387–396.
- Sudholt, D. (2008). *Computational complexity of evolutionary algorithms, hybridizations, and swarm intelligence*.
- Sun, M., Li, Q., Zhu, J., & Zhang, Y. (2018). Particle Swarm Optimization Algorithm Based on Graph Knowledge Transfer for Geometric Constraint Solving. *International Conference on Computer Engineering and Networks*, 452–462.
- Sunde, G. (1987). A CAD system with declarative specification of shape. In *Intelligent CAD Systems I* (pp. 90–104). Springer.
- Verroust, A., Schonek, F., & Roller, D. (1992). Rule-oriented method for parameterized computer-aided design. *Computer-Aided Design*, 24(10), 531–540.
- Yamaguchi, Y., & Kimura, F. (1990). A constraint modeling system for variational geometry. *Geometric Modeling for Product Engineering*, 221–233.
- Ye, T. (2019). *Treatment of Geometric Constraints in Well Placement Optimization*. STANFORD UNIVERSITY.
- Yi, W., Cao, C., & Zhang, C. (2010). The geometric constraint solving based on hybrid differential evolution and particle swarm optimization algorithm. *2010 International Conference on Intelligent Control and Information Processing*, 692–695.
- Yuan, H, Li, W., Yi, R., & Zhao, K. (2006). The TPSO algorithm to solve geometric constraint problems. *Journal of Computational Information Systems*, 2, 1311–1316.



- Yuan, Hua, Chang, X., Zhao, K., & Li, W. (2009). A Very Fast Converge Method for Geometric Constraint Solving. *2009 Second International Workshop on Computer Science and Engineering, 2*, 498–502.
- Yuan, Hua, Li, W., Zhao, K., & Yi, R. (2007). Parallel Search Algorithm for Geometric Constraints Solving. In R. Shumaker (Ed.), *Virtual Reality* (pp. 157–164). Springer Berlin Heidelberg.
- Zhang, Y., Wang, S., & Ji, G. (2015). A comprehensive survey on particle swarm optimization algorithm and its applications. *Mathematical Problems in Engineering, 2015*.