

Tipo de artículo: Artículo original

Solucionador de restricciones geométricas mediante métodos de reducción de grafos para el sistema AsiXmec

Geometric constraint solver using graph-based reduction methods for AsiXmec system

Yanays Fernández Miranda^{1*} , <https://orcid.org/0000-0002-2137-1201>

Ángel Alberto Vazquez Sánchez² , <https://orcid.org/0000-0002-3130-7983>

¹ Departamento de Matemática, Facultad de Tecnologías Educativas, Universidad de las Ciencias Informáticas. Correo electrónico: yfernandez@uci.cu

² Departamento de Inteligencia Computacional, Facultad 4, Universidad de las Ciencias Informáticas. Correo electrónico: aavazquez@uci.cu

* Autor para correspondencia: yfernandez@uci.cu

Resumen

El Diseño Asistido por Computadora constituye actualmente la principal herramienta para el diseño de piezas 2D y 3D. Debido al alto costo de estos productos en el mercado internacional, Cuba decidió implementar su propio software para esta industria, el sistema informático multiplataforma AsiXmec. El objetivo de esta investigación fue diseñar un solucionador de restricciones geométricas eficiente para esta herramienta. Para lograrlo se planteó utilizar programación C++ y QtCreator como entorno de desarrollo. Para integrar el solucionador diseñado al sistema se utilizaron artefactos ingenieriles de la metodología ágil XP. Se diseñó el algoritmo de reducción basado en grafos incorporándole el método de descomposición y recombinación (S-DR). La propuesta tiene entre sus funcionalidades solucionar restricciones de horizontalidad, verticalidad, paralelismo, perpendicularidad, concetricidad, tangencia y simetría, con la novedad que permite satisfacer un conjunto grande de restricciones para darle solución a un problema de restricción. Se propusieron casos de estudio para comprobar la eficiencia del algoritmo y se estableció una comparación con otros algoritmos conocidos. Finalmente, se obtuvo el diseño de un solucionador de restricciones geométricas óptimo para la herramienta AsiXmec.

Palabras clave: AsiXmec; método de reducción de grafos; restricciones geométricas; solucionador.

Abstract

Computer Aided Design is currently the main tool for the design of 2D and 3D parts. Due to the high cost of these products in the international market, Cuba decided to implement its own software for this industry, the AsiXmec multiplatform computer system. The objective of this research was to design an efficient geometric constraint solver for this tool. To achieve this, we proposed to use C++ programming and QtCreator as a development environment. To integrate the designed solver to the system, engineering artifacts from the agile XP methodology were used. The graph-based reduction algorithm designed incorporates the decomposition and recombination (S-DR) method. One of the functionalities of the proposal is to solve horizontality, verticality, parallelism, perpendicularity, concentricity, tangency and symmetry constraints, with the novelty that it allows satisfying a large set of constraints to solve a constrained problem. We proposed case studies to check the efficiency of our proposal and a comparison with other well-known algorithms. Finally, the design of an optimal geometric constraint solver for the AsiXmec tool is obtained.

Keywords: AsiXmec; geometric constraint; graph-based reduction algorithm; solver.



Esta obra está bajo una licencia *Creative Commons* de tipo **Atribución 4.0 Internacional** (CC BY 4.0)

Recibido: 18/09/2022
Aceptado: 06/11/2022
En línea: 08/11/2022

Introducción

El Diseño Asistido por Computadora (CAD, por sus siglas en inglés) se usa en casi todas las industrias, en proyectos tan variados como el diseño de paisajes, la construcción de puentes, el diseño de edificios de oficinas, la animación en películas y el diseño de partes y piezas para la industria. Con los programas de diseño CAD 2D o 3D, se pueden realizar diferentes tareas, como crear un modelo 3D de un diseño, aplicar materiales y efectos de iluminación, y documentar el diseño con cotas y otras anotaciones (Arroyo & Espinosa, 2006).

La investigación se enmarca en software CAD empleados para modelar piezas de la industria a ser impresas en 2D y 3D. Internacionalmente, estos softwares se dividen por nivel de experiencia en principiante, nivel intermedio y profesional, siendo estos últimos extremadamente costosos con precios que oscilan entre los 3000 y 10000 euros por licencia anual. CREO, es uno de los softwares líderes del mercado en diseño de productos, desarrollado por *Parametric Technology Corporation*. También se destaca Solidworks, desarrollado por la empresa francesa *Dassault Systemes*, el cual es conocido por ser práctico y detallado en las piezas industriales que diseña (Howard & Contreras, 2020).

Además de los anteriores, entre los softwares CAD con herramientas profesionales ideales para el diseño, simulación, visualización y documentación de productos se destaca el Autodesk Inventor, desarrollado en California, Estados Unidos. Las versiones del Autodesk Inventor actualmente tienen un costo de 2190 USD por suscripción anual, comercializadas solo para países con convenio (Shih & Randy, 2021). Países de América Latina como México, Brasil y Argentina utilizan Autodesk. Cuba necesita desarrollar su propia herramienta de diseño paramétrico para facilitar la confección de piezas para la industria y sortear limitaciones del bloqueo económico comercial y financiero.

En la Facultad 4 de la Universidad de las Ciencias Informáticas (UCI), se desarrolla el proyecto de Investigación y Desarrollo “Plataforma para el modelado paramétrico en ingeniería basado en tecnologías de código abierto”. Este proyecto, aprobado por el Programa Nacional de Automática, Robótica e Inteligencia Artificial, está relacionado con el diseño CAD. Como parte del proyecto, se desarrolla el sistema CAD nombrado AsiXmec, que incorpora un modelador geométrico y paramétrico, basados en el paradigma de diseño mediante características e historial de operaciones. AsiXmec está pensado para acortar el ciclo de diseño, en un área caracterizada por una elevada competitividad. Para ello, captura tanto la geometría del diseño CAD como la intención del diseñador, utilizando los



conceptos conocidos como modelado paramétrico y diseño basado en restricciones geométricas (Disem & Vertex, 2015). Un componente, usualmente conocido con el término solucionador, juega un papel importante en los sistemas CAD basados en restricciones, pues resulta el encargado de resolverlas.

Los solucionadores de restricciones geométricas se utilizan precisamente en el modelado paramétrico, además pueden ser empleados en la modelación de moléculas, los análisis de tolerancia, la robótica y las demostraciones de teoremas geométricos. El diseño basado en restricciones geométricas permite al usuario describir figuras especificando un boceto grosero, y adicionándole a éste restricciones. Así, el solucionador de restricciones debe derivar automáticamente la figura correcta que se necesita. Típicamente, en 2D, el diseño basado en restricciones detalla objetos geométricos como puntos, líneas, círculos y cónicas mediante un conjunto de restricciones: distancia entre puntos, puntos y líneas, líneas paralelas, ángulos entre líneas, relaciones de incidencia entre puntos y líneas, puntos y círculos, relaciones de tangencia entre líneas y círculos o entre círculos. Varios métodos de resolución se han propuesto para resolver sistemas de restricciones geométricas, entre ellos: simbólicos, numéricos, basados en reglas y de construcción de grafos.

Los solucionadores basados en reglas confían en la formulación de predicados (Sunde, 1988). Aunque proveen un estudio cualitativo de las restricciones geométricas, la enorme cantidad de cálculos que necesitan (búsqueda y coincidencia exhaustiva) los hacen inapropiados para aplicaciones que necesitan una respuesta en tiempo real.

En contraste, los solucionadores basados en la construcción de grafos tienen su origen en la teoría de grafos. Se basan en el análisis de la estructura de un grafo de restricciones y su estrategia provee mecanismos para desarrollar algoritmos robustos y eficientes (Arinyo & Puertas, 2010). En particular, pueden utilizar el método de reducción de grafos para, basado en el concepto de esqueletos¹, planificar la descomposición de un sistema bien restringido en subsistemas más pequeños y recombinar las soluciones de estos subsistemas para derivar la solución del sistema completo.

Por otra parte, los métodos simbólicos son extremadamente consumidores de tiempo; son exponenciales en tiempo y espacio, y pueden utilizarse únicamente para sistemas pequeños (Anderl, 1995). Los métodos numéricos, variante implementada actualmente en el sistema AsiXmec para la resolución de sistemas de ecuaciones, presentan complejidad algorítmica $O(n^3)$ o peor (Fudos & Hoffmann, 1997). La mayoría de estos métodos presentan incluso

¹Enfoque de técnicas para el reconocimiento de imágenes binarias; sintetizan y ayudan a entender el objeto, debido a que preservan la información estructural que integra las características de la imagen.



dificultades para manipular sistemas de ecuaciones indeterminados o sobredeterminados. De hecho, en AsiXmec, cuando se complejiza la geometría y aparecen más de 30 restricciones, la velocidad de reacción del modelo ante un evento de modificación se compromete, volviendo impracticable su uso. Algunas implementaciones realizan un preprocesamiento para descomponer el sistema de ecuaciones en cuestión, antes de correr el método numérico concreto; otras estrategias optimizan el proceso de resolución podando el espacio de soluciones.

La estrategia numérica actualmente implementada en el sistema AsiXmec, para resolver las restricciones geométricas, impacta negativamente en la eficiencia del módulo de creación de entidades con decenas de restricciones. En resumen, el solucionador actual de AsiXmec compromete la eficiencia del sistema debido a:

1. El desempeño del software no es el esperado de acuerdo con los niveles de servicios pactados, teniendo en cuenta las necesidades del usuario (cuando las restricciones son más de 30 el sistema colapsa).
2. No proporciona resultados en un adecuado tiempo de respuesta, por lo que su desempeño no es eficiente bajo las condiciones establecidas.
3. Al aumentar la complejidad de la geometría no ofrece control de las soluciones en las que el usuario está interesado, por lo que tiene mayor costo computacional.

En este sentido, se impone la búsqueda de una solución para esta problemática, que conlleva a realizar esta investigación con el objetivo de: Diseñar el solucionador de restricciones geométricas de AsiXmec mediante el método de reducción basado en grafos.

Materiales y métodos

Con el fin de caracterizar los solucionadores existentes en las aplicaciones se ha realizado una observación para obtener y analizar la información referente a ellos. Se han investigado artículos publicados por la comunidad científica referentes al tema en cuestión, que recorren desde los primeros pasos en esta esfera en los 90s hasta la actualidad, los cuales permitieron una caracterización y valoración de los algoritmos existentes. Teniendo en cuenta las tecnologías empleadas en el sistema CAD en desarrollo, al cual se le incorporará los resultados de la presente investigación, se han identificado como tecnologías a emplear el lenguaje de programación C++, el framework de desarrollo Qt5.5 y la biblioteca OpenCascade Community Edition 0.17. Para la integración del solucionador de restricciones al sistema AsiXmec se diseñaron artefactos ingenieriles correspondientes a la metodología ágil *Extreme*



Programming (XP). Se realizó un análisis de estudio de casos para evaluar la efectividad del algoritmo para el sistema en situaciones reales.

Los solucionadores de los sistemas CAD contemporáneos son basados en la construcción de grafos que relacionan las restricciones con los elementos geométricos. Estos solucionadores deben ser capaces de manejar un gran volumen de elementos y restricciones geométricas, por lo que se utiliza la estrategia “divide y vencerás” para su solución. Esta estrategia aplicada a los sistemas de restricciones geométricas se denomina Descomposición-Recombinación (Ait-Aoudia, 2010).

En los métodos de reducción basados en grafos para planificar la descomposición del sistema en subsistemas más pequeños, se analiza la estructura de la restricción de manera que se encuentre una secuencia de grafos $G_i(G_1; G_2; \dots; G_k)$, donde G_1 es la restricción inicial de la gráfica G. Cada G_i contiene subgrafos (*clusters*) simplificados en un grafo $G_{i+1} = T_i(G_i)$ donde la transformación T_i es llamado mapa simplificador que opera en G_i . Si la gráfica G está bien restringida, entonces se termina la relación cuando el grafo G_k no puede ser más simplificado. Lo explicado anteriormente se realiza usando el término cluster, introducido por (Hoffman, 2001), para clasificar un subgrafo en particular.

Un problema se puede categorizar según la cantidad de soluciones posibles: está bien restringido si existe un número finito de soluciones al problema; mientras que un problema con infinitas soluciones es infra restringido. Un problema está súper-restringido si una restricción puede ser eliminada y todavía el sistema de restricciones tiene un número finito de soluciones (Sitharam, 2018).

De esta forma, se procede a esbozar los principales pasos lógicos del algoritmo propuesto, los cuales son definidos posteriormente:



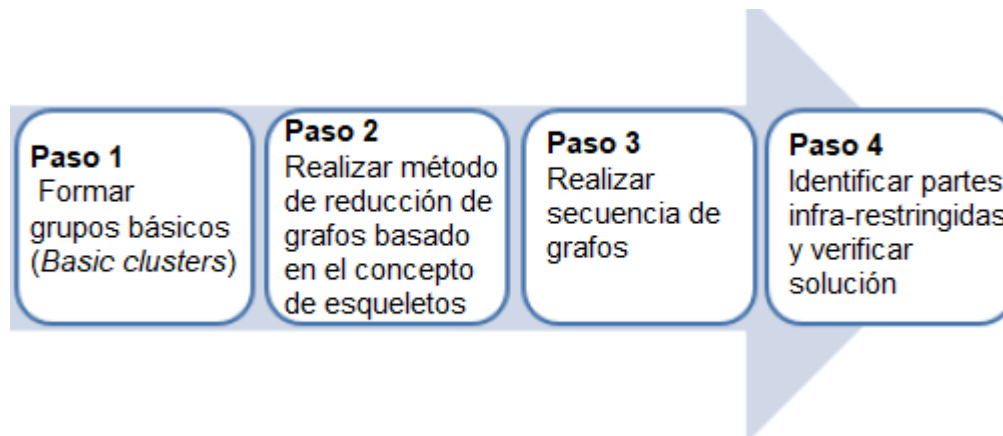


Figura 1. Pasos para la construcción del algoritmo. Fuente: Elaboración propia

Paso1. Formación de grupos básicos

Una fase importante del algoritmo de reducción de grafos es la formación de grupos básicos. Esto se basa en una estructura geométrica, donde los elementos se reconocen relativamente el uno al otro.

Un grupo básico es un grafo mínimo denso y su extensión secuencial se calcula sumando objetos geométricos, uno cada vez. En la figura 2 se muestra este proceso para una gráfica de restricción, donde se encuentran tres grupos básicos C1, C2 y C3.

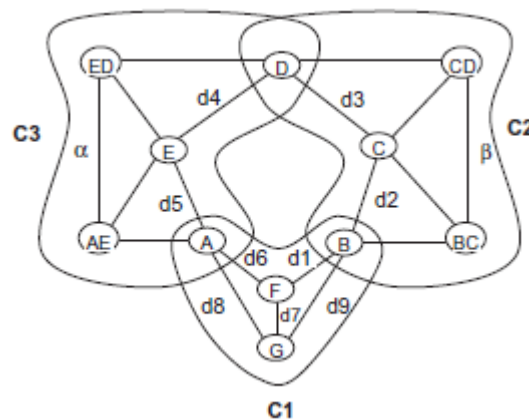


Figura 2. Representación del proceso de formación de grupos básicos. Fuente: (Sun M Li, 2018)

Para encontrar un cluster, el algoritmo empleado, comienza con un punto arbitrario desde el cual recupera todos los puntos donde se alcanza densidad. Si este es un punto central se produce un cluster; si es un punto-frontera, ningún punto alcanza densidad desde este y se procede al siguiente punto. Debido a que se usan parámetros globales de



densidad, se pueden fusionar dos clusters de diferente densidad si están cerca uno del otro. El agrupamiento recursivo de los puntos de un cluster solo es necesario en condiciones donde pueden detectarse fácilmente.

Paso 2. Método de reducción de grafos basado en el concepto de esqueletos

El método de reducción de grafos basado en el concepto de esqueletos, introducido por (Hoffmann & Joan-Arinyo, 2005), se realiza en este trabajo como sigue: Cuando los grupos básicos son encontrados se realiza la transformación del grafo simplificándolo. Después de esta transformación se forma el esqueleto que son grafos G_s (V_s , E_s) donde V_s son los vértices y E_s son las aristas.

Si se añade una arista y no es miembro del set inicial de restricciones, se inserta una restricción “virtual” cuyo valor es fácilmente determinado debido a que los nodos pertenecen a una estructura rígida (cluster básico). Esto se hace para facilitar el proceso de resolución.

Una vez que los nodos del esqueleto están situados, cada cluster se posiciona relativamente al esqueleto calculando los parámetros requeridos, obteniéndose como resultado el esquema final. Para ello, se debe tener en cuenta que cada cluster básico comparte con el esqueleto al menos dos nodos.

Paso 3 Secuencia de grafos

Para realizar la planificación de la descomposición del sistema, con el método de reducción de grafos se utiliza el algoritmo de descomposición-recombinación (S-DR), de manera que se realice una secuencia de grafos $G_i (G_1; G_2; \dots; G_k)$. G_1 es la restricción inicial del grafo G . Cada G_i contiene varios clusters simplificados en un grafo $G_{i+1} = T_i(G_i)$ donde T_i es la transformación (mapa simplificado) dando el esqueleto de G_i (G_{i+1} es el esqueleto de G_i). Si el grafo de restricción G está bien restringido, entonces el proceso termina cuando G_k consta de un cluster básico. Si el grafo G_1 está estructuralmente bien restringido, entonces todos los grafos G_i están bien restringidos.

Los mapeos T_i abstraen cada cluster básico a una estructura rígida cuyos nodos son los compartidos. G_{i+1} es estrictamente más pequeño que G_i . El proceso de simplificaciones terminará después de k pasos cuando G_k sea un cluster básico. A cada paso, el grafo G_{i+1} es obtenido de G_i sin importar el orden de las formaciones de cluster básicos.



En el caso de la planificación S-DR, usando grafos restringidos, la preservación de la solución implica que el proceso de abstracción no transforme correctamente subgrafos bien restringidos o súper restringidos a partes infra-restringidas y viceversa. Durante la construcción del plan $P = (G_1; G_2; \dots; G_k)$ los mapeos T_i abstraen cada cluster básico a una estructura rígida. Si el grafo G_k está bien restringido (rígido) entonces también el grafo G_{k+1} .

S-DR encuentra un mínimo denso en cada fase. Descubre un plan cuando la entrada es típicamente una construcción geométrica. Prácticamente, la solución es obtenida cuando los clusters básicos del grafo inicial son posicionados. El mapeo T_i nunca crea vértices adicionales.

Paso 4. Identificar partes infra-restringidas y verificar solución

Si la gráfica de restricción G está infra-restringida, el proceso para obtener la secuencia de grafos identificará las partes infra-restringidas.

La primera prueba se hace comprobando el número de nodos compartidos de un cluster básico con los otros clusters básicos (se asume que la gráfica de restricción está conectada). Si un cluster básico solo comparte un nodo entonces todo el cluster se puede mover relativamente para las otras partes.

La segunda prueba se hace durante la construcción de la secuencia de grafos $G_i(G_1; G_2; \dots; G_k)$. El proceso no terminará con un cluster básico, sino cuando $G_i = T_i(G_i)$, $G_{i+1} = G_i$ (todos los cluster están abstraídos y el grafo G_m es infra-restringido). El grafo G_i debe ser rígido agregando restricciones específicas para que el grafo inicial esté bien restringido.

Los requisitos o funcionalidades que provee la propuesta son:

RF1- Restricciones de horizontalidad.

RF2- Restricciones de verticalidad.

RF3- Restricciones de paralelismo.

RF4- Restricciones de perpendicularidad.

RF5- Restricciones de concentricidad.

RF6- Restricciones de tangencia.

RF7- Restricciones de simetría.



RF8- Mover restricciones de una entidad.

RF9- Marco de resolución extensible, que permite la adición de restricciones geométricas a nuevas entidades.

RF10- Manipulación geométrica.

RF11- Satisfacer conjunto de restricciones para la solución de problemas.

Los patrones de diseño de comportamiento tratan con algoritmos y la asignación de responsabilidades entre objetos. El patrón de diseño que se utiliza en la propuesta es el Observer, que permite definir un mecanismo de suscripción para que varios objetos reciban señales sobre cualquier evento que le suceda al objeto que están observando. Los suscriptores se suscriben a un evento y cuando se disparan las señales, en referencia a este evento, el suscriptor actúa. En el proyecto se actúa cuando el usuario adicione una entidad o restricción, lo cual representa el evento y como respuesta se procede a crear el grafo de restricciones para darle solución al problema de restricción. De esta manera, los objetos que deben observar el comportamiento de otros, sólo actúan durante un tiempo limitado o en casos específicos (cuando se generen las restricciones se procede a darles solución a las mismas.)

Resultados y discusión

A continuación, en la figura 3, se representa el diagrama que muestra los componentes que utiliza el solucionador de restricciones en la herramienta AsiXmec.

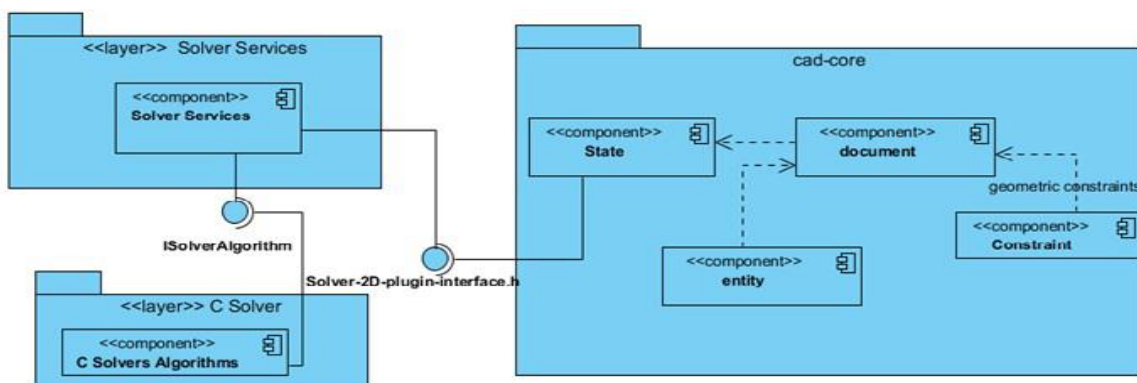


Figura 3. Diagrama de componentes. Fuente: Elaboración propia

Se realizó la propuesta de casos de estudio en tiempo real donde el algoritmo calcula los parámetros requeridos y se verifica que el grafo de restricciones restante sea de menor complejidad, de manera que quedan las restricciones resueltas con un tiempo de respuesta óptimo para garantizar el diseño de los objetos geométricos.

Para poder medir los resultados de las ejecuciones del algoritmo se calcula el error cometido al calcular cada problema de restricción. Este valor es el error total cometido al calcular el número de soluciones ($ErrorST$) y se define como la suma de errores cometidos al calcular el número de soluciones de cada cluster ($ErrorSC_i$). En el caso de problemas sobre-determinados, los grafos son infra-restringidos, con soluciones infinitas por lo que se calcula un error aproximado.

$$ErrorST = \sum_{i=1}^n ErrorSC_i$$

Con los resultados obtenidos se realiza una comparación con diferentes algoritmos: Algoritmo de Condensación (CA) (Ait-Aoudia, 2010) y el Algoritmo de Vértices de la Frontera (FA) (Puertas, 2018). La comparación se restringió a estos algoritmos por ser los más utilizados para la resolución de restricciones geométricas. Con esto se logran exponer elementos necesarios, para verificar la eficiencia del algoritmo propuesto, para la confección del solucionador de restricciones geométricas del sistema AsiXmec. Entre estos elementos se encuentra la garantía de que no se pierda información en la aplicación del método.

Después de comprobado el algoritmo, utilizado en la solución propuesta, se realizan las pruebas de integración. Estas pruebas validan que todos los elementos del sistema funcionan juntos correctamente probándolos en grupo. Las pruebas de integración aseguran que el solver de restricciones, una vez integrado al sistema AsiXmec, trabaje en conjunto con otros componentes de la plataforma sin detectar problemas ni conflictos entre ellos.

En la figura 4 se muestra la ventana de AsiXmec donde el usuario genera las entidades para posteriormente adicionar las restricciones necesarias.



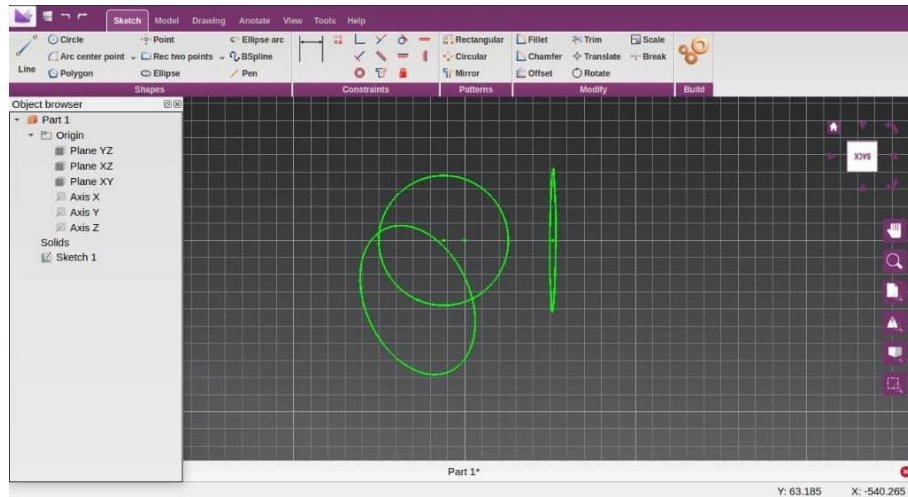


Figura 4. Ventana de esbozo. Fuente propia.

El solucionador de restricciones diseñado constituye un resultado para implementar en el desarrollo de sistemas informáticos del país, basados en tecnologías de código abierto, destinados a la industria, al modelado y al diseño paramétrico, con el objetivo de avanzar en la soberanía tecnológica del país.

Conclusiones

La investigación realizada arrojó información valiosa sobre el algoritmo de reducción de grafos; así como de la aplicación del método de descomposición y recombinación para la resolución de sistemas de restricciones en un tiempo de respuesta óptimo. Se logró elaborar el diseño de la propuesta de integración del solucionador de restricciones geométricas al sistema AsiXmec, integrándolo al componente cad-core como plugin de esta herramienta. Se diseñó un solucionador de restricciones geométricas mediante métodos de reducción de grafos para la herramienta AsiXmec; lo que permitió facilitar el análisis de un conjunto de restricciones mediante el uso de grafos reducidos.

Conflictos de intereses

Los autores no poseen conflictos de intereses.

Contribución de los autores

1. Conceptualización: Yanays Fernández Miranda, Ángel Alberto Vazquez Sánchez.



2. Curación de datos: Ángel Alberto Vazquez Sánchez.
3. Análisis formal: Yanays Fernández Miranda, Ángel Alberto Vazquez Sánchez.
4. Investigación: Yanays Fernández Miranda, Ángel Alberto Vazquez Sánchez.
5. Metodología: Yanays Fernández Miranda, Ángel Alberto Vazquez Sánchez.
6. Administración del proyecto: Yanays Fernández Miranda.
7. Software: Yanays Fernández Miranda, Ángel Alberto Vazquez Sánchez.
8. Supervisión: Yanays Fernández Miranda.
9. Validación: Yanays Fernández Miranda.
10. Visualización: Yanays Fernández Miranda
11. Redacción – borrador original: Yanays Fernández Miranda, Ángel Alberto Vazquez Sánchez.
12. Redacción – revisión y edición: Yanays Fernández Miranda, Ángel Alberto Vazquez Sánchez.

Financiamiento

La investigación que da origen a los resultados presentados en la presente publicación recibió fondos de la Oficina de Gestión de Fondos y Proyectos Internacionales del Ministerio de Ciencia, Tecnología y Medio Ambiente de Cuba bajo el código PN223LH004-001.

Referencias

- Ait-Aoudia, S. a. (2010). *A 2D geometric constraint solver using a graph reduction method*. Advances in Engineering Software.
- Anderl, R. y. (1995). Parametric design and its impact on solid modeling applications. *In Proceedings of the third ACM symposium on Solid modeling and applications*, 1-12.
- Arinyo, R. J., & Puertas, M. I. (2010). Geometric constraintgraphs decomposition based on computing graph circuits.
- Arroyo, R. R., & Espinosa, R. A. (2006). Las ventajas e inconvenientes del cad/cam. *Sistemas computacionales*.
- Centro de Entornos Interactivos 3D, Vertex. (2015). «Manual de usuario del asistente mecánico asixmec». (U. d. Informáticas, Ed.) *Proyecto diseño y simulación de estructuras mecánicas*.
- Fudos, I., & Hoffmann, C. (1997). A graph-constructive approach to solving systems of geometric constraints. *ACM Transactions on Graphics (TOG)*, 179-216.
- Hoffman, C. M. (2001). Decomposition plans for geometric constraint systems, part i: Performance measures for CAD. *Journal of Symbolic Computation*.



- Hoffmann, & Joan-Arinyo. (2005). A brief on constraint solving. *Computer-Aided Design and Applications*.
- Howard, & Contreras, L. (2020). Top 10 de los mejores software cad para todos los niveles. *3Dnatives*.
- Puertas, M. I. (2018). Geometric constraintgraphs decomposition.
- Shih, & Randy. (2021). Learning autodesk inventor 2021. doi:978-1630573447
- Sitharam, M. J. (2018). Handbook of geometric constraint systems principles. *In Handbookof Geometric Constraint Systems Principles*.
- Sun M Li, Z. (2018). «Particle swarm optimization algorithm based on graph knowledge transfer for geometric constraint solving. *International Conference on Computer Engineering and Networks*.
- Sunde. (1988). Specication of shape by dimensions and other geometric constraints. (N. Holland, Ed.) *Elsevier Science Publishers BV*.

