

ALGORITMO GENÉTICO PARA LA OPTIMIZACIÓN DE SECUENCIAS DE ENSAMBLE

**Tesis presentada en opción al Título Académico de
Máster en Diseño y Fabricación Asistidos por
Computadora**

Autor: Ing. Edwin Estévez Parra

Tutor: Dr. C. Ricardo Lorenzo Ávila Rondón



FACULTAD DE INGENIERÍA



**Holguín, Cuba
Enero 2015**

Dedicatoria

Este trabajo está dedicado a mi familia y a aquellos quienes, de una u otra forma, lo hicieron posible.

Agradecimientos

Deseo agradecer a mis padres por su paciencia y comprensión durante el desarrollo de este trabajo.

A mi tutor por su apoyo incondicional.

A mis colegas del Centro de Estudios CAD/CAM, gracias también por su apoyo incondicional.

Al Dios de toda sabiduría, y mi Señor, Jesucristo, sin su ayuda este trabajo no hubiera sido posible.

TABLA DE CONTENIDOS

ÍNDICE DE FIGURAS	vi
ÍNDICE DE TABLAS	vii
RESUMEN	viii
INTRODUCCIÓN	1
CAPÍTULO 1. ESTADO ACTUAL DE LA GENERACIÓN, PLANIFICACIÓN Y OPTIMIZACIÓN DE SECUENCIAS DE ENSAMBLE	6
1.1 El ensamble mecánico	6
1.2 Planificación de la Secuencia de Ensamble	8
1.2.1 Modelado del ensamble mecánico	10
1.2.2 Clasificación de las secuencias de ensamble	15
1.2.3 Representación de las secuencias de ensamble	17
1.2.4 Las restricciones topológicas y geométricas en la determinación de la factibilidad de la secuencia de ensamble	21
1.2.5 Generación de las secuencias de ensamble	25
1.2.6 Evaluación y Selección de las secuencias de ensamble	28
1.3 Técnicas de optimización avanzadas para la planificación de la secuencia de ensamble	31
1.3.1 Algoritmo genético	33
1.3.2 Optimización por Colonia de Hormigas	37
1.3.3 Optimización por Enjambre de Partículas	37
1.4 Conclusiones del capítulo	39

CAPÍTULO 2. MÉTODO ASISTIDO POR COMPUTADORA PARA LA OPTIMIZACIÓN DE LAS SECUENCIAS DE ENSAMBLE.....	40
2.1 Método para la obtención de la información de interferencia entre las piezas del ensamble	40
2.1.1 Descripción del método	41
2.1.2 Materiales y métodos	41
2.1.3 Resultados del método	45
2.1.4 Discusiones.....	48
2.1.5 Conclusiones parciales.....	48
2.2 Método de optimización para la planificación de las secuencias de ensamble basado en algoritmos genéticos.....	50
2.2.1 Modelo de Optimización	50
2.2.2 Método de optimización.....	57
2.2.3 Caso de estudio y discusiones	60
2.3 Conclusiones del capítulo.....	67
CONCLUSIONES GENERALES	68
RECOMENDACIONES	69
REFERENCIAS BIBLIOGRÁFICAS.....	70
ANEXOS	76

ÍNDICE DE FIGURAS

Figura 1.1. Desarrollo y Fabricación de un producto. Temas y soluciones relacionados con el ensamble.	7
Figura 1.2. Simple ejemplo axialmente simétrico: Bolígrafo de Bourjault.	11
Figura 1.3. Diferentes maneras de representar el mismo ensamble	14
Figura 1.4. Ensamblajes que no pueden ser construidos por un plan: (a) secuencial, (b) monótono, (c) coherente, (d) lineal.	16
Figura 1.5. Ensamble de una transmisión automática (AFI)	18
Figura 1.6. Piezas A y B compuestas por superficies complejas.	24
Figura 1.7. Frecuencia de los objetivos de ASP	32
Figura 2.1. Pasos para la extracción de la información geométrica de un fichero STEP	42
Figura 2.2. Aproximación triangular de las superficies de las piezas con el uso del mallado triangular (a) Pieza sólida (b) Malla triangular correspondiente a la pieza en (a)	43
Figura 2.3. Detección de las interferencias entre las piezas.	44
Figura 2.4. Algoritmo para obtener las relaciones de interferencias entre las piezas de un ensamble	47
Figura 2.5. Ensamble simplificado de un disco de freno	47
Figura 2.6. Ensamble de un reductor	47
Figura 2.7. Restricciones del proceso de ensamble relacionadas con la optimización de la secuencia de ensamble.	51
Figura 2.8. Ejemplo de un cromosoma	57
Figura 2.9. Cruce por emparejamiento parcial (PMX)	59
Figura 2.10. Mutación por inversión	59
Figura A1.1. Ensamble con piezas prismáticas y cilíndricas	76
Figura A1.2. Ensamble con piezas de forma libre	76
Figura A2.1. Modelo CAD 3D del ensamble de una transmisión automática (AFI). (a) vista isométrica, (b) vista de sección del ensamble con los elementos de fijación, (c) vista explosionada	77
Figura A3.1. Diagrama de flujo de la optimización de la secuencia de ensamble mediante GA.	78

ÍNDICE DE TABLAS

Tabla 1.1. Tipos de planes de ensamble.....	17
Tabla 2.1. Matriz de interferencia utilizada para inferir la dirección de ensamble +X.....	61
Tabla 2.2. Matriz de interferencia utilizada para inferir la dirección de ensamble –X.....	61
Tabla 2.3. Matriz de interferencia utilizada para inferir la dirección de ensamble +Y.....	62
Tabla 2.4. Matriz de interferencia utilizada para inferir la dirección de ensamble –Y.....	62
Tabla 2.5. Matriz de precedencia local del ensamble.....	63
Tabla 2.6. Matriz de estabilidad del ensamble	63
Tabla 2.7. Matriz de herramientas del ensamble.....	64
Tabla 2.8. Secuencias de ensamble optimizadas con el algoritmo propuesto GA-ASP.....	64
Tabla 2.9. Mejores secuencias de ensamble obtenidas con los algoritmos GA-ASP, MA-ASP y ACO-ASP respectivamente.....	65
Tabla A1.1. Diferencias entre el método propuesto para la determinación de las interferencias y el método descrito por Pan, Smith, Smith	76

RESUMEN

La Planificación de la secuencia de ensamble forma parte de la Planificación del ensamble. La secuencia de ensamble es la parte más importante de un plan de ensamble. El ensamble tiene una participación importante en los recursos y los tiempos de fabricación utilizados en la producción. Por lo tanto, su automatización y optimización mediante técnicas computacionales se hacen necesarias para asegurar la competitividad de los productos fabricados.

El objetivo de este trabajo es lograr la planificación asistida por computadora de las secuencias de ensamble de un producto mecánico. Se propone un método computacional denominado GA-ASP basado en un algoritmo genético que utiliza un modelo de optimización que incluye varias restricciones del proceso de ensamble como la restricción geométrica, la precedencia local del ensamble, la estabilidad del ensamble, el cambio de las direcciones de ensamble y el cambio de las herramientas de ensamble. Estas restricciones se representan en estructuras de datos de tipo matriz, de las cuales las matrices de interferencia se generan automáticamente a partir del modelo CAD mediante el uso de un nuevo método computacional que se describe. GA-ASP es implementado en un software y se aplica a un caso de estudio. Como resultado, las secuencias de ensamble obtenidas por GA-ASP y la comparación de estas con las producidas por dos métodos existentes, demuestran que al considerar las restricciones del proceso de ensamble utilizadas por el modelo de optimización, GA-ASP obtiene mejores soluciones para el ensamble del producto analizado.

INTRODUCCIÓN

En los años 50's, el mercado para los productos estadounidenses fue ocho veces mayor que el próximo mercado más grande que hubo. Sin embargo, Estados Unidos se aferraba a las técnicas de producción en masa tradicionales, mientras que al comienzo de los 50's, Japón y posteriormente Alemania adoptaron nuevas técnicas de fabricación: la ingeniería concurrente, la automatización de la fabricación y la fabricación flexible integrada por computadora. Como resultado, de 1960 a 1989, la participación de los Estados Unidos en el mercado mundial de los bienes fabricados cayó del 20% al 17% y del 1er al 3er lugar, detrás de Japón y Alemania. Las nuevas técnicas de fabricación les llevaron a los clientes productos más personalizados, con nuevos modelos en períodos de tiempo más cortos [1].

De forma general, en los años 60's y 70's, el mundo occidental cambió de una sociedad de mercado en masa –donde los gustos homogeneizados eran fácilmente satisfechos con unas pocas elecciones de productos– a una diversidad sin precedente. En un corto período la sociedad en masa unificada se fraccionó en muchos grupos diversos de personas con una gama amplia de gustos diferentes, surgió una sociedad de mercado segmentado y de mercado descentralizado. La sociedad existente se caracteriza por ser multiopcional donde “todo viene en al menos 31 sabores” [2].

En la actualidad, los clientes esperan productos de alta calidad, menos costosos, de forma más rápida, y más personalizados a los gustos individuales.

El ensamble tiene una participación importante en los recursos y los tiempos de fabricación utilizados en la producción, por lo que es un candidato importante si se pretende reducirlos. Se conoce que más del 40% del costo de fabricación se emplea en el ensamble, y el ensamble constituye del 20% al 70% de todo el trabajo de fabricación [3].

Existe una necesidad creciente para sistematizar y computarizar la generación de las secuencias de ensamble por varias razones [4, 5]:

- Aunque muchos ingenieros experimentados tienen la habilidad para idear maneras eficientes de cómo ensamblar un producto dado, se necesitan procedimientos sistemáticos para garantizar que ninguna secuencia de ensamble buena sea pasada por alto. Para los productos complejos, el número de secuencias de ensamble factibles puede ser tan elevado que incluso los ingenieros más habilidosos pueden fallar en notar muchas posibilidades. La disponibilidad de un procedimiento sistemático que sea correcto y completo garantizará que sean generadas todas las secuencias factibles.
- Las tareas de planificación y programación en la fabricación consumen tiempo y son propensas a errores. Para pequeños lotes de producción, el costo de planificación y programación puede influir considerablemente en el costo total de la producción. Además, el tiempo empleado en la planificación y en la programación puede retrasar la producción actual. La automatización de estas tareas agilizará su ejecución, reducirá su costo, y mejorará su calidad. Por lo tanto, se necesitan procedimientos sistemáticos para facilitar la automatización de la planificación y la programación de los sistemas de ensamble.
- En entornos donde esté presente la ingeniería concurrente, la automatización de la planificación de la secuencia ayudará al diseñador a evaluar los requerimientos del proceso de ensamble de las diferentes soluciones de diseño para un producto dado. Para algunos productos, los pequeños cambios en el diseño pueden tener un impacto considerable en las alternativas de ensamble.
- Un plan de ensamble se genera al considerar un número de factores que afectan su calidad, como la naturaleza y la estructura del producto a ensamblar, el tamaño del lote y el diseño de la instalación de ensamble. Si cualquiera de esos factores cambian, la calidad del plan de ensamble puede cambiar, de este modo el plan

podría ser cambiado (sujeto a consideraciones económicas). La automatización de la planificación permite la planificación rápida y eficiente cuando todos los elementos que contribuyen a la secuencia de ensamble son conocidos, y además admite la replanificación cuando esos elementos cambian.

Sin embargo, la planificación asistida por computadora de la secuencia de ensamble es un problema típico NP-completo bajo restricciones del proceso de ensamble, lo cual dificulta su solución. Se conoce que el tamaño del espacio de búsqueda es exponencialmente proporcional al número de piezas que componen los productos. Para satisfacer las demandas del desarrollo del producto, se utilizan métodos computacionales eficientes para abordar este problema con el objetivo de generar las secuencias de ensamble óptimas o cercanas a la óptima. Por lo que se propone el siguiente **problema científico**:

¿Cómo planificar las secuencias de ensamble para un producto mecánico al considerar las restricciones del proceso del ensamble de forma tal que se haga más eficiente y menos costoso?

De lo anterior se desprende la **necesidad** de disponer de un modelo que incluya las restricciones del proceso de ensamble y que esté apoyado por un método de optimización para obtener secuencias buenas u óptimas en la planificación de este proceso.

El **objeto** de investigación se centra en la planificación de las secuencias de ensamble y como **campo** los métodos asistidos por computadora para la planificación de las secuencias de ensamble.

El **objetivo general** es lograr la planificación asistida por computadora de las secuencias de ensamble de un producto mecánico.

Como **hipótesis** de esta investigación se plantea lo siguiente:

Un método de optimización basado en los algoritmos genéticos y que utilice un modelo que incluya las restricciones del proceso de ensamble, permitirá lograr la planificación asistida por computadora de las secuencias de ensamble de un producto mecánico.

Para dar cumplimiento al objetivo general se declaran las siguientes **tareas**:

1. Revisar la bibliografía referente al proceso de ensamble mecánico, con énfasis en la planificación asistida por computadora de la secuencia de ensamble.
2. Definir los procedimientos para obtención de las restricciones geométricas, de precedencia local del ensamble y de otras restricciones del proceso de ensamble, ya sea de forma automática o interactiva.
3. Aplicar un método basado en los algoritmos genéticos que permita la optimización de las secuencias de ensamble y que considere un modelo que incluya las restricciones del proceso de ensamble.
4. Elaborar el informe de la investigación.

Para llevar a cabo las tareas planteadas se emplearon **métodos** de investigación teóricos y empíricos:

1. El *análisis y síntesis* permitió descomponer mentalmente el procesamiento de la información en sus partes, las múltiples relaciones entre ellas y la simplificación de la información a procesar.
2. Mediante el método *histórico-lógico* se logró comprender la evolución histórica de la planificación de la secuencia de ensamble con la utilización de técnicas computacionales.
3. La *modelación* se utilizó al determinar el modelo de optimización para la planificación de las secuencias de ensamble.
4. La *observación* permitió conocer el comportamiento del objeto de investigación, tal y como este se manifiesta en la realidad.

Como **aportes** de este trabajo se encuentran:

1. La aplicación de un método de optimización basado en los algoritmos genéticos para la planificación de las secuencias de ensamble y que utiliza un modelo que considera las restricciones del proceso de ensamble siguientes:
 - El número de piezas que violan las restricciones geométricas
 - El número de piezas que violan la precedencia local del ensamble
 - El número de piezas inestables del ensamble
 - El cambio de las direcciones de ensamble

- El cambio de las herramientas
2. La creación de un método asistido por computadora que permite automatizar la obtención de la información de interferencia entre las piezas del ensamble a partir de un modelo geométrico tridimensional del producto.

La presente investigación se ha dividido en dos capítulos:

En el Capítulo 1 “Estado actual de la generación, planificación y optimización de secuencias de ensamble” se realiza una revisión del estado del arte de la planificación asistida por computadora de la secuencia de ensamble y su papel en la planificación del proceso del ensamble mecánico. Además se muestran las técnicas computacionales para la representación, generación, evaluación y optimización de las secuencias de ensamble.

El Capítulo 2 titulado “Método asistido por computadora para la optimización de las secuencias de ensamble” se compone de dos partes. La primera parte trata sobre un método para la obtención de la información de interferencia entre las piezas del ensamble, el cual tiene como objetivo obtener las matrices de interferencia correspondientes a las seis direcciones principales de desensamble, y además es automatizado con la creación de un software. La segunda parte es un método basado en los algoritmos genéticos que utiliza un modelo de costo del ensamble para la optimización de las secuencias de ensamble al considerar las restricciones del proceso.

Al finalizar el documento se exponen las conclusiones generales y las recomendaciones.

CAPÍTULO 1. ESTADO ACTUAL DE LA GENERACIÓN, PLANIFICACIÓN Y OPTIMIZACIÓN DE SECUENCIAS DE ENSAMBLE

En el presente capítulo se realiza una revisión de las principales investigaciones que componen la Planificación de la secuencia de ensamble –en idioma inglés Assembly Sequence Planning (ASP)– desde una perspectiva computacional. Se describen el modelado del ensamble y la representación, la generación, la evaluación y la selección de la secuencia de ensamble al considerar varias restricciones del proceso. Al finalizar, se mencionan propuestas de la literatura para solucionar el problema ASP al utilizar técnicas computacionales basadas en inteligencia artificial.

1.1 El ensamble mecánico

El ensamble es el proceso culminante en la fabricación. Agrupa todos los procesos anteriores de diseño, ingeniería, fabricación y logística para crear un objeto que realiza una función. El ensamble es diferente de los procesos unitarios tradicionales de fabricación como el fresado y el rectificado, porque es intrínsecamente integrador: sin duda agrupa las piezas, pero también agrupa (o debería agrupar) a las personas y empresas que diseñan y hacen esas piezas [6]. Según la definición anterior, el ensamble puede ser visto desde la ingeniería concurrente como el proceso integrador de todos los procesos que intervienen en la fabricación del producto. A excepción de estos procesos, el ensamble puede ser visto como un proceso, el cual es un movimiento de colecciones de piezas que comienza con cada pieza separada del resto, y termina con las piezas ensambladas en una configuración deseada [7].

El ensamble tiene una participación importante en los recursos y los tiempos de fabricación utilizados en la producción, por lo que es un candidato importante si se pretende reducirlos. Una respuesta para mantener el trabajo de ensamble en los países desarrollados, mientras se mantienen competitivos en una escala global, es la automatización del proceso del ensamble. La optimización del ensamble se convierte en una fuente de ahorros importante y sostenible. El ensamble está muy difundido a través del desarrollo y de las etapas de producción de los bienes fabricados [5]. En la figura 1.1 se muestra un esquema de temas relacionados con el ensamble, dónde y cómo emergen

estos y el alcance de la optimización en cada caso.

La optimización en el ensamble tiene que enfocarse en los aspectos de las tres etapas del desarrollo y la producción. Durante la concepción del producto y el diseño, el producto es optimizado desde el punto de vista del ensamble al aplicar los principios del Diseño para el Ensamble –en inglés Design for Assembly (DFA)–, el cual tiene como objetivo reducir el número de piezas y la complejidad de las operaciones, lo que permite que el resultado sea más fácil de fabricarse y ensamblarse.

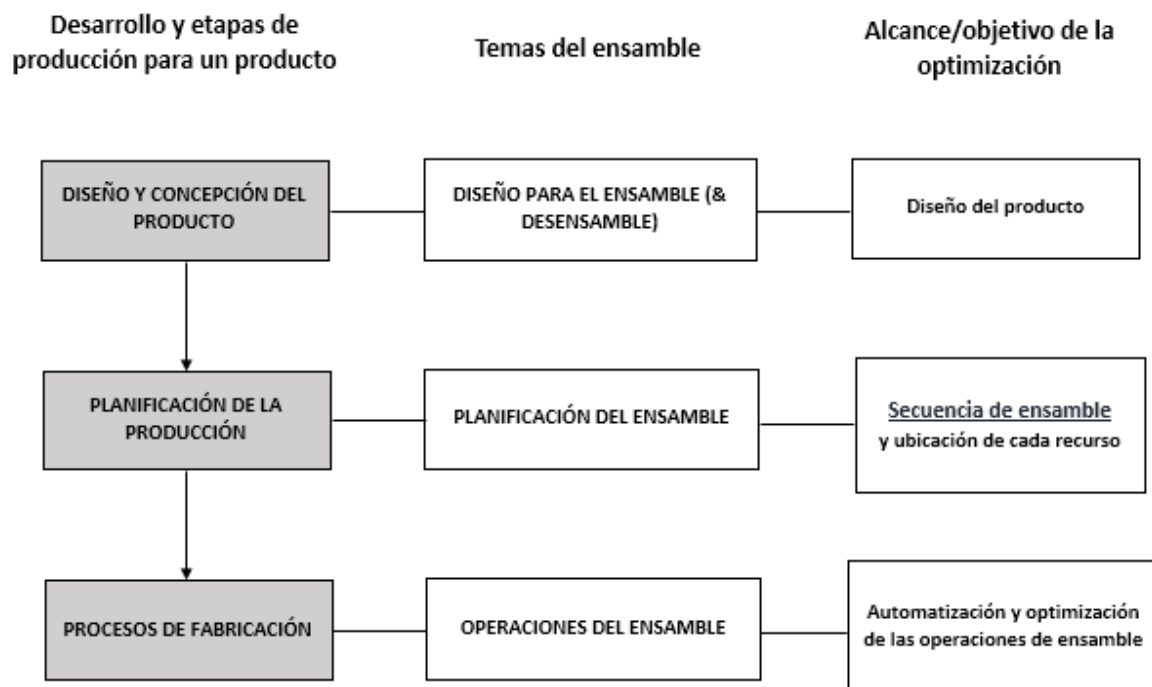


Figura 1.1. Desarrollo y Fabricación de un producto. Temas y soluciones relacionados con el ensamble. Fuente [5].

La optimización durante la planificación de la producción incluye la planificación del ensamble e intenta encontrar la secuencia óptima de las tareas de ensamble, el diseño de la línea de ensamble y la ubicación de cada recurso.

La optimización durante la etapa de producción está dirigida a los temas de automatización –la eficiencia y automatización de la línea/célula– y la automatización de las operaciones de ensamble –la organización de tareas específicas, el balanceo de las tareas, entre otras.

1.2 Planificación de la Secuencia de Ensamble

La planificación de los procesos es una tarea común en la fabricación de las piezas. Es la actividad responsable de la conversión de los datos del diseño a las instrucciones de fabricación o de trabajo.

Los planificadores de los procesos primeramente deben evaluar los datos del diseño y las especificaciones tales como las características o elementos geométricos, dimensiones, tolerancias, acabado superficial, y materiales para seleccionar una secuencia apropiada de operaciones de procesamiento y de máquinas específicas. Los detalles de la operación como la preparación y la planificación de corte, la planificación de las plantillas y los dispositivos, las velocidades, los avances, las herramientas, y los pasos de ensamble, son determinados y se calculan los tiempos normados y los costos. La planificación de los procesos representa el vínculo entre el diseño de ingeniería y la fabricación en la planta de producción. Es un principal determinante del costo de fabricación y la rentabilidad.

El vacío entre el Diseño asistido por computadora –en inglés Computer Aided Design (CAD)– y la Fabricación asistida por computadora –en inglés Computer Aided Manufacturing (CAM)– puede ser acortado considerablemente mediante el desarrollo de mejores sistemas para la planificación de los procesos.

La planificación de los procesos puede hacerse de forma manual o con la asistencia de una computadora. La planificación manual de los procesos tiene sus desventajas. Está estrechamente ligada a la experiencia personal y al conocimiento del planificador de las instalaciones productivas, el equipamiento, sus capacidades, los procesos, y las herramientas; esto trae como resultado planes inconsistentes. La planificación manual de los procesos consume mucho tiempo, y es lenta. Además, es lenta al responder a los cambios en el diseño y la producción del producto [8].

Un plan de ensamble es un procedimiento escrito que describe un proceso de ensamble para un producto particular. Tales planes inicialmente pueden ser generados en una forma abstracta, al omitirse ciertos detalles. Todos esos detalles deben ser completados antes que se ejecute el proceso de ensamble correspondiente.

La generación de los planes de ensamble en algún nivel de detalle útil se llama la planificación del ensamble o la planificación de la secuencia de ensamble [7].

La planificación del ensamble es la ciencia de determinar la forma factible y óptima de juntar las piezas de un ensamble para producir el producto final [5]. La planificación del ensamble está caracterizada por los siguientes elementos:

1. Encontrar la secuencia óptima de tareas de ensamble (problema de la Planificación de la secuencia de ensamble o ASP)
2. Encontrar la ubicación óptima de cada recurso.

En el proceso del ensamble mecánico, la actividad de ASP juega un papel fundamental en el diseño del producto y del sistema de ensamble. La elección de la secuencia de ensamble puede impactar directamente en la eficiencia y el costo del proceso de ensamble. La secuencia en la cual las piezas son ensambladas puede influir en el número de los cambios en la orientación del ensamble y en las herramientas, en el número y la complejidad de los dispositivos de fijación, y en la complejidad de las operaciones de ensamble.

Al tratar estos temas surge la necesidad de los sistemas computarizados que permitirán que la planificación de los procesos sea realizada parcialmente o completamente por una computadora. La planificación de los procesos asistida por computadora –en idioma inglés Computer Aided Process Planning (CAPP)– ayuda a simplificar y a mejorar la actividad de la planificación de los procesos, proporciona al usuario planes de procesos óptimos de una manera consistente y rápida, y logra una utilización más efectiva de los recursos de fabricación.

La planificación del ensamble con frecuencia se enfrenta al problema de los estancamientos. Los estancamientos son secuencias de ensamble que no terminan en el producto completo. Un ejemplo sencillo es un recipiente con una tapa y otros componentes dentro de él. Si la tapa es enroscada al recipiente antes de poner los componentes dentro de él, los componentes no pueden ser insertados y el producto completo no se puede obtener. Una manera de planificar el ensamble y evitar los estancamientos es experimentar a partir del producto completo, es decir, ensamblado, y hacer un inventario de todas las secuencias posibles de desensamble, lo cual resulta en

todas las secuencias de ensamble posibles cuando el proceso es invertido. Este método es llamado planificación del ensamble hacia atrás. La suposición clave es que el ensamble es el opuesto del desensamble. Sin embargo, esta suposición es una simplificación de la realidad y solamente es verdadera bajo suposiciones estrictas, las cuales incluyen la ausencia de fuerzas y un enfoque de cuerpo rígido. Además en este método se considera que las operaciones del ensamble son las inversas de las operaciones del desensamble.

1.2.1 Modelado del ensamble mecánico

Un ensamble mecánico es una composición de piezas interconectadas que forman una unidad estable. Cada pieza es un objeto sólido. Las piezas están interconectadas ya sea que tengan una o más superficies en contacto [4].

El ensamble puede descomponerse en piezas individuales y subensambles. Un subensamble es un subconjunto de piezas que forman parte del ensamble, y debe tener las siguientes características [9]:

1. Cada pieza del subensamble es ensamblada al menos con una de las otras piezas del ensamble.
2. El subensamble es estable, lo que significa que las piezas interiores no pueden desprenderse del subensamble cuando el subensamble es manipulado como un todo.
3. Un subensamble no puede interferir con las otras piezas o subensambles en el proceso de ensamble.

Se requiere de una representación computacional de los ensambles mecánicos para poder generar automáticamente las secuencias de ensamble. Es por ello que el modelo de ensamble debe contener los datos esenciales necesarios para que el planificador de ensamble automático pueda generar todas las secuencias de ensamble.

El modelo del ensamble debe proveer no solo la información que permite al planificador generar todas las secuencias de ensamble sino también la información que permite a este razonar acerca de la factibilidad de las operaciones de ensamble que componen a una secuencia. Desde que Bourjault [10] presentó un método para la generación

algorítmica de todas las secuencias de ensamble para los productos mecánicos, los avances de las investigaciones en esta área han traído consigo la creación de sistemas que han progresado desde sistemas interactivos hasta sistemas completamente automatizados.

Bourjault [10] se propuso modelar el proceso de ensamble. Para lograr ese objetivo utilizó dos tipos de grafos para representar los productos. El grafo de contactos contiene un nodo para cada pieza en el ensamble, y un arco para cada contacto entre dos piezas. Entre cada par de piezas puede existir más de un contacto. A partir del grafo de contactos Bourjault definió el grafo de conexiones, el cual tiene un nodo por cada pieza en el ensamble y un arco por cada par de piezas que tienen al menos un contacto. Estos contactos pueden ser reales o virtuales.

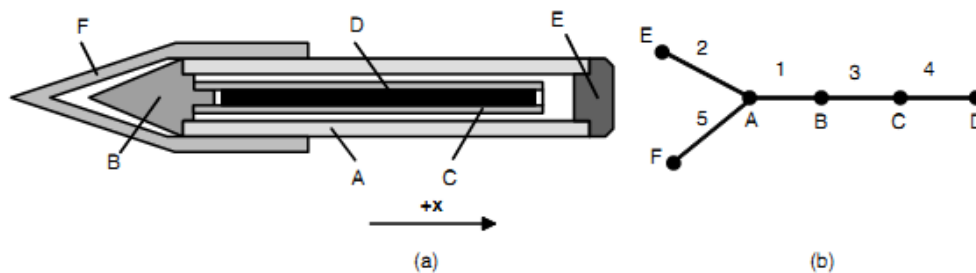


Figura 1.2. Simple ejemplo axialmente simétrico: Bolígrafo de Bourjault. (a) ensamble; (b) diagrama de conexión. Fuente [11].

La figura 1.2 muestra un ejemplo simple de un bolígrafo que fue extensamente examinado por Bourjault y se ha convertido en un ejemplo de prueba elemental para muchos investigadores. El bolígrafo está compuesto por seis componentes: cuerpo (A), cabeza (B), cartucho (C), tinta (D), botón (E), y tapa (F). En la figura 1.2 (b) se muestra un grafo de conexiones o diagrama de conexión.

Homem de Mello y Sanderson [4] desarrollaron un modelo relacional de ensambles, y plantearon que este modelo contiene toda la información mecánica y geométrica necesaria para calcular la factibilidad de las operaciones de ensamble sin requerir la intervención del diseñador. El modelo relacional consiste en un grafo relacional, más el conjunto de las funciones de atributos asociados. El grafo relacional extiende del grafo de conexiones para incorporar una representación de los dispositivos de sujeción que

fijan una pieza a otra. Ellos extendieron del grafo de conexiones para incluir los contactos virtuales entre las piezas del ensamble. Los contactos virtuales se definen entre las piezas que no contactan físicamente, pero que durante el desensamble interfieren entre ellas.

La figura 1.3 (c) muestra el grafo relacional para el ensamble simplificado de la linterna mostrado en la figura 1.3 (a). Las líneas representan arcos que conectan dos nodos, y estas se consideran relaciones entre las dos entidades representadas por los nodos. Los círculos y triángulos representan las entidades de los contactos y de los dispositivos de sujeción respectivamente. Las piezas se representan por círculos de color negro etiquetados por el nombre de la pieza según corresponda. En este grafo relacional, cada relación es de tipo contacto-pieza, es decir, representan un acople simple entre dos piezas, con excepción de las relaciones R5 y R13 que son relaciones de tipo sujeción que indican que los contactos C2 y C5 son fijados con algún elemento de sujeción, y además R6 y R14 las cuales son relaciones de tipo contacto-agente que indican que el agente del elemento de sujeción son los propios contactos, y no una pieza como por ejemplo un tornillo.

En su modelo relacional, Homem de Mello y Sanderson también definen funciones atributos que asocian las entidades (piezas, contactos y dispositivos de sujeción) o las relaciones entre estas entidades, a sus características como: el tipo de sujeción, las entidades relacionadas mediante una relación, y la forma de una pieza.

Delchambre [12] desarrolló un modelo que también distingue las relaciones de contacto a partir de las relaciones de los dispositivos de fijación. El grafo del producto es un grafo simple donde los nodos pueden ser piezas o dispositivos de fijación, y se especifica un conjunto de atributos asociados con cada arco. Estos atributos definen el tipo de contacto entre las piezas, las direcciones posibles de ensamble y los procesos de sujeción complementarios. Los contactos entre las piezas son clasificados como colocaciones, inserciones y sujeciones.

Ko y Lee [13] propusieron una representación de los ensambles al utilizar estructuras de datos jerárquicas, y obtuvieron un procedimiento para generar la jerarquía de componente a partir del grafo de acoplamiento. Modelaron el ensamble con el uso de vínculos virtuales entre las piezas acopladas. Cada vínculo virtual se caracteriza por las

condiciones de acoplamiento entre las dos piezas asociadas con el vínculo y por el tipo de vínculo. Una representación gráfica de los vínculos virtuales (grafo de acoplamiento) es similar al diagrama de conexión. Las piezas del ensamble son estructuradas en el “árbol jerárquico”. Este árbol jerárquico es una representación en forma de árbol de la jerarquía de pieza. La jerarquía de pieza se determina a partir del grafo de acoplamiento, del cual se utiliza la conectividad de cada pieza. Sin embargo, esta estructura tiene el inconveniente de que no es suficientemente general para representar todas las secuencias de ensamble factibles.

Modelar el ensamble de un producto compuesto por una docena de piezas es una tarea sencilla, no obstante cuando el producto posee un número elevado de componentes su modelado se hace bastante complejo, aún más si no se cuenta con una herramienta computacional que permita diseñar el ensamble al considerar la geometría original de las piezas individuales y los tipos de contactos que existen entre ellas, además de posicionar espacialmente cada pieza al ensamblar de una manera automática.

Mucho se ha avanzado en la creación de sistemas CAD que permitan estas ventajas. Sistemas CAD actuales como Solidworks, Inventor, Pro/Engineer y Catia, han sido equipados con la habilidad de crear el ensamble de un producto a partir de sus piezas individuales. Estos sistemas permiten un modelado del ensamble adecuado al considerar las posiciones espaciales y las relaciones jerárquicas entre las piezas, y las relaciones del ensamble (condiciones de acople) entre las piezas. Para crear el modelo del ensamble para un producto, primeramente los diseñadores crean las piezas individuales. Una vez que el diseño de las piezas se completa, los diseñadores pueden proceder a crear y a analizar el ensamble. Para crear el ensamble a partir de sus piezas se requiere especificar las relaciones espaciales y de acople entre las piezas. El análisis del ensamble puede incluir el chequeo de interferencia, las propiedades de masa, el análisis cinemático y dinámico, y el análisis por elementos finitos. Los sistemas CAD establecen un vínculo entre un ensamble y sus piezas individuales, de forma tal que los diseñadores solo necesitan cambiar las piezas individuales para modificar el diseño, y el sistema actualiza el modelo del ensamble de forma automática.

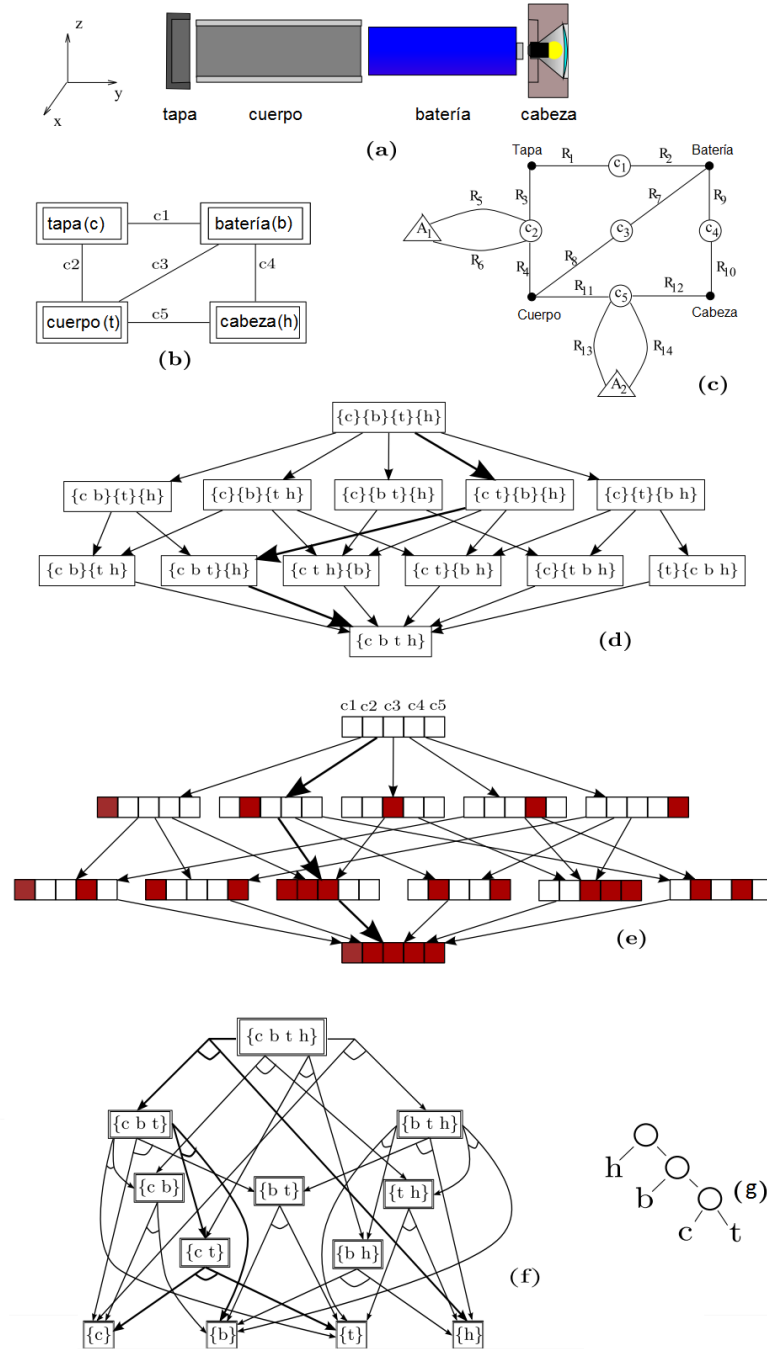


Figura 1.3. Diferentes maneras de representar el mismo ensamble (a) una linterna en su forma desensamblada. El grafo de conexiones (b) y el grafo relacional (c) para representar el modelo de ensamble. Para representar las secuencias de ensamble: el diagrama de estado (d) orientado a los componentes y (e) orientado a las conexiones, y (f) el grafo AND/OR. En (g) se representa un árbol de ensamble parcial para una secuencia de ensamble determinada. La misma secuencia está representada con líneas gruesas en los diagramas de estado y en el grafo AND/OR. Adaptado de [14].

1.2.2 Clasificación de las secuencias de ensamble

La mayoría de los sistemas para la planificación del ensamble restringen los tipos de planes o las secuencias que ellos pueden generar. En la figura 1.4 se muestran algunos ensambles 2D que no pueden ser contruidos por planes de ensamble de tipo secuencial, monótono, coherente y lineal. Estos planes pueden ser definidos de la siguiente manera [5, 7]:

Secuencial: Si puede ser descompuesto en un conjunto de operaciones tales que cada operación involucra el movimiento de un conjunto de piezas a lo largo de una trayectoria común. En la figura 1.4 (a) para el ensamble se requiere del movimiento de dos de sus piezas a lo largo de dos trayectorias diferentes, por lo que se necesita un plan de ensamble no secuencial.

Monótono: Si ninguna operación separa ningún par de piezas que estuvieran en sus posiciones relativas finales entre sí, y todas las operaciones dejan todas las piezas movidas en sus posiciones relativas finales en relación a alguna pieza no movida. De esta manera, se excluyen en los planes monótonos las operaciones que dejan las piezas en posiciones temporales, como se muestra en el tipo de plan no monótono de la figura 1.4 (b). C tiene que estacionarse en la ranura de A después que A y B han sido ensamblados.

Coherente: Si durante el proceso de ensamble, todas las operaciones añaden una pieza o subensamble a otro componente y se establece al menos una conexión entre ambos. La figura 1.4 (c) presenta un ejemplo donde C1, C2 y B1 no pueden ser ensamblados después que C y B han sido ensamblados a A. Como resultado, B-B1 y C-C1-C2 tienen que ensamblarse antes que B y C sean insertados en A en una secuencia de ensamble no coherente. Para realizar tal tipo de operaciones suele utilizarse un dispositivo de fijación que mantenga las posiciones relativas de los dos componentes a unir con el tercero.

Lineal: Si solo puede moverse una pieza en cada instante, o sea, cada operación añade una sola pieza a un subensamblé. Para ensamblar productos similares al de la figura 1.4 (d) se requiere de un plan no lineal.

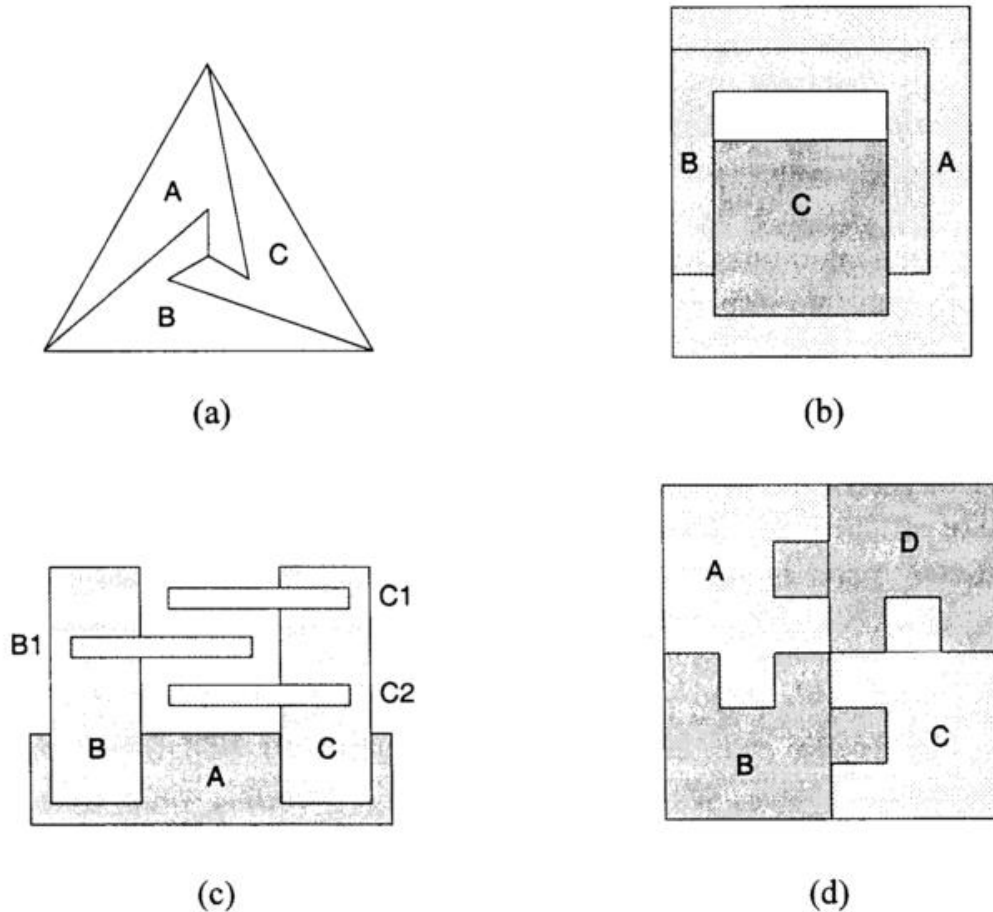


Figura 1.4. Ensamblés que no pueden ser contruidos por un plan: (a) secuencial, (b) monótono, (c) coherente, (d) lineal. Fuente [7].

Un plan de ensamble puede ser coherente o no coherente, lineal o no lineal, monótono o no monótono, y secuencial o no secuencial, o cualquier combinación de las situaciones anteriores, es decir, 16 combinaciones de planes como se muestra en la Tabla 1.1. En la figura 1.5 se muestra un ensamble que puede ser construido por un plan de ensamble coherente-secuencial-lineal-monótono.

	¿Coherente?	¿Secuencial?	¿Lineal?	¿Monótono?
1.	NO	NO	NO	NO
2.	NO	NO	NO	SÍ
3.	NO	NO	SÍ	NO
4.	NO	NO	SÍ	SÍ
5.	NO	SÍ	NO	NO
6.	NO	SÍ	NO	SÍ
7.	NO	SÍ	SÍ	NO
8.	NO	SÍ	SÍ	SÍ
9.	SÍ	NO	NO	NO
10.	SÍ	NO	NO	SÍ
11.	SÍ	NO	SÍ	NO
12.	SÍ	NO	SÍ	SÍ
13.	SÍ	SÍ	NO	NO
14.	SÍ	SÍ	NO	SÍ
15.	SÍ	SÍ	SÍ	NO
16.	SÍ	SÍ	SÍ	SÍ

Tabla 1.1. Tipos de planes de ensamble. Fuente [5].

1.2.3 Representación de las secuencias de ensamble

Muchas representaciones de las secuencias de ensamble utilizadas en la literatura acerca de la actividad de ASP pueden clasificarse en dos grupos [15]: listas ordenadas y representaciones gráficas.

La lista ordenada puede ser una lista de tareas, lista de estados de ensamble, o lista de subconjuntos de conexiones. En las listas ordenadas cada secuencia de ensamble está representada por un conjunto de listas. Aunque este conjunto de listas podría representar una descripción correcta y completa de todas las secuencias de ensamble factibles, no es necesariamente la representación más compacta y útil de las secuencias.

Los esquemas gráficos mapean las operaciones y los estados de ensamble en elementos esquemáticos específicos, y comparten gráficamente subsecuencias y estados comunes en muchas secuencias de ensamble. Esto permite crear representaciones más compactas y útiles que pueden abarcar todas las secuencias de ensamble factibles.

Entre los esquemas gráficos se encuentran: el diagrama de precedencia, el árbol invertido, el diagrama de estado y el grafo AND/OR.

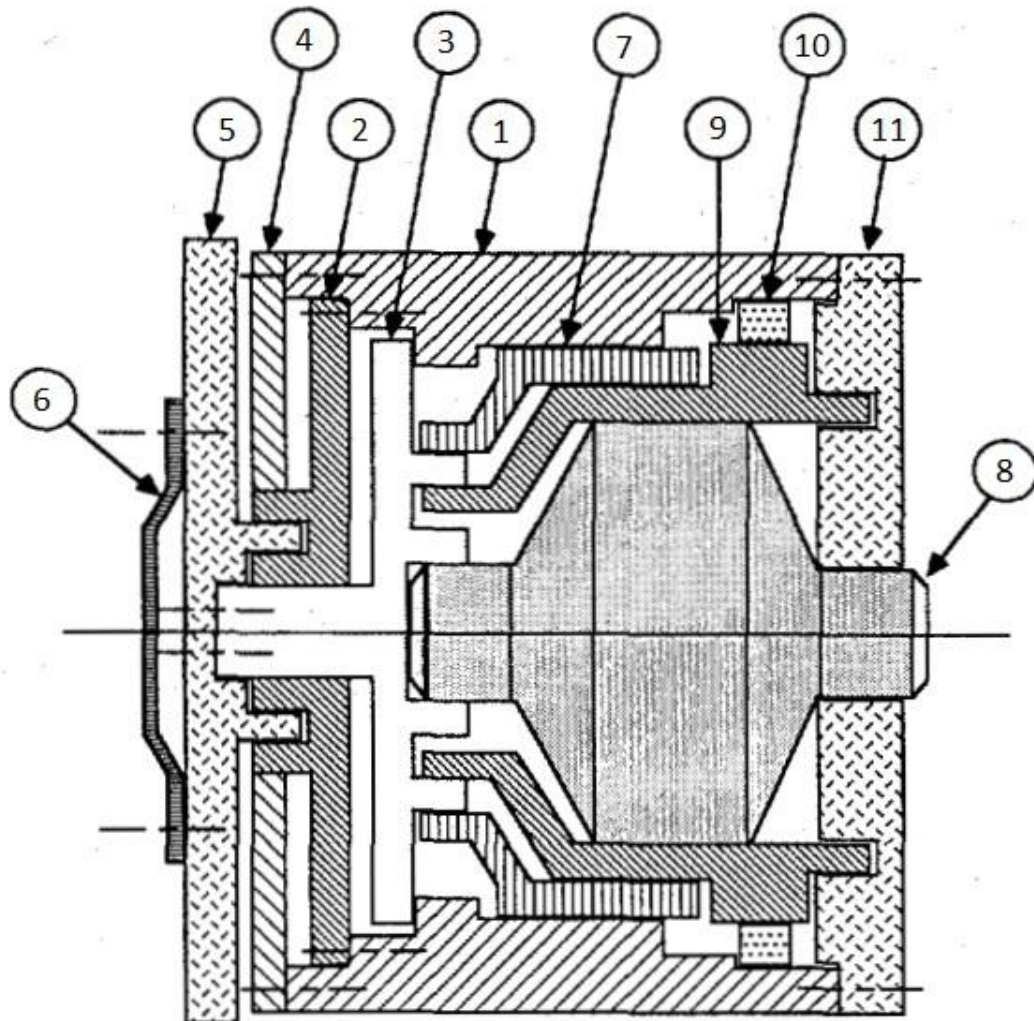


Figura 1.5. Ensamble de una transmisión automática (AFI) que puede ser ensamblado por un plan secuencial, monótono, coherente y lineal. Adaptado de [16].

Diagrama de precedencia

Un proceso puede ser desglosado en un conjunto de tareas. Estas tareas tienen que realizarse en un orden determinado (por ejemplo, la tarea C no puede realizarse antes de la tarea A y la tarea B). Este es expresado en relaciones de precedencia. El conjunto de relaciones de precedencia es representado gráficamente en la forma de un diagrama de precedencia.

Los diagramas de precedencia son representados como grafos dirigidos en los que los

nodos son operaciones y los arcos son las relaciones de precedencia entre las operaciones. Si un arco apunta de la operación A hacia la operación B, esto implica que la operación B solo puede ser realizada después que se haya realizado la operación A (aunque no necesariamente inmediatamente después de la operación A).

Las relaciones de precedencia son importantes en los problemas del Balanceo de la línea de ensamble –en idioma inglés Assembly Line Balancing (ALB)– y son utilizados para ejemplificar las relaciones entre las tareas u operaciones. Uno de los primeros algoritmos para asistir al balanceo de línea fue desarrollado en [17]. Este algoritmo toma como entrada un diagrama llamado grafo de precedencia que indica el orden en el cual las tareas de ensamble pueden realizarse. En principio, este grafo contiene todas las secuencias de ensamble factibles en la forma de una red. La desventaja de esta representación es la carencia de naturaleza algorítmica en el desarrollo del diagrama.

Naphade et al. [18] se enfocaron en los aspectos de satisfacción de restricciones del problema ASP, el cual involucra la determinación de un grafo de precedencia de ensamble correcto a partir de un conjunto arbitrariamente complejo de las restricciones del diseño, restricciones físicas y condiciones del diseño preestablecidas. En su investigación, ellos plantean que el grafo de precedencia tiene un mayor mérito como un plan de ensamble que una secuencia simple especificada completamente.

Árbol invertido

Bourjault [10] representó todas las secuencias de ensamble válidas en la forma de un árbol invertido que describe los órdenes posibles del ensamble. El árbol invertido solo da las secuencias de conexiones, y no las secuencias de ensamble existentes. Además, este no contiene ninguna información sobre los subensambles.

Diagrama de estado

El diagrama de estado puede representar las secuencias de ensamble en dependencia del enfoque que se utilice, ya sea orientado a las conexiones u orientado a los componentes.

Grafo de secuencia de conexiones

La representación del grafo de secuencia de conexiones está orientado a las conexiones entre las piezas del ensamble. Por lo tanto, este esquema da las secuencias de las conexiones en vez de las secuencias de ensamble. Sin embargo, los estados no representan un conjunto de piezas, pero sí un conjunto de relaciones entre las piezas. En este grafo, cada nodo representa un estado de ensamble y las conexiones establecidas son marcadas con un color como se muestra en la figura 1.3 (e). Los arcos representan los movimientos de ensamble correspondientes al establecimiento de las conexiones. Una secuencia de ensamble se define como un camino o trayectoria al partir de un estado completamente desensamblado a un estado completamente ensamblado.

Grafo dirigido de las secuencias de ensamble

Esta representación está orientada a los componentes o las piezas del ensamble. En este grafo los nodos corresponden al conjunto de los subensambles estables, y los arcos corresponden a las tareas de ensamble como lo muestra la figura 1.3 (d). Un grafo dirigido podría contener secuencias de ensamble no factibles.

Grafo AND/OR

Homem de Mello y Sanderson describieron la representación de las secuencias de ensamble basada en el grafo AND/OR [19]. Cada subensamble representado por un nodo en este grafo se corresponde a un subconjunto del conjunto de piezas en el producto, y el nodo raíz corresponde al producto completamente ensamblado.

Los nodos representan los subensambles estables y los hiper-arcos representan las operaciones de ensamble factibles. Cada operación de ensamble es representada como un hiper-arco que apunta desde los subensambles hijos hasta el subensamble padre (relación AND). Ya que un subensamble puede ser ensamblado de varias formas, esto resulta en diferentes hiper-arcos que apuntan hacia el subensamble (relación OR). Figura 1.3 (f).

En comparación con el diagrama de estado, el grafo AND/OR tiene las ventajas que para ensambles compuestos por más de cuatro piezas el número de nodos es menor, y además este grafo puede representar la posibilidad de la ejecución paralela de las

operaciones de ensamble ejecutadas en subensambles separados.

1.2.4 Las restricciones topológicas y geométricas en la determinación de la factibilidad de la secuencia de ensamble

Las restricciones topológicas del ensamble incluyen los contactos y las conexiones entre las piezas o los componentes, y se pueden representar como vínculos en el diagrama de conexiones. Las restricciones geométricas se incluyen en el modelo de ensamble 2D o 3D, tales como la estructura integral de la geometría, los elementos geométricos como los planos, cilindros, conos y superficies curvas. Las posiciones espaciales relativas de las piezas o componentes son determinadas por las restricciones geométricas [9].

La restricción geométrica en el ensamble trata el ensamble de los componentes sin que ocurra ninguna colisión. Al acoplarse dos piezas, al menos debe existir una trayectoria libre de colisión que permita acoplar los componentes. La obtención de las interferencias entre las piezas de un producto contribuye a determinar la factibilidad de las secuencias de ensamble según la geometría de las piezas involucradas, luego a partir de esta factibilidad se pueden deducir las relaciones de precedencia del ensamble para generar el conjunto de las secuencias.

Algunos autores han representado la información de la relación entre las piezas de un ensamble en forma matricial. Dini y Santochi [20] describieron un procedimiento para generar las secuencias posibles de ensamble, el cual estaba basado en un modelo matemático del producto, el cual se obtiene a través de la definición de tres matrices: matriz de interferencia, matriz de contacto y matriz de conexión. En este estudio, la matriz de interferencia representa las interferencias entre las piezas de un producto en los ejes de coordenadas principales +x, +y, +z. Huang y Huang [21] presentaron un método para almacenar los datos de las matrices de interferencia en la matriz de precedencia del desensamble, donde las posibles direcciones de desensamble están representadas por un número binario. Sinanoglu y Borklu [22] describieron un método para determinar los estados de la factibilidad geométrica mediante las matrices de intersección, las cuales son utilizadas para representar la interferencia entre las piezas durante la operación de ensamble. Estas matrices se definen a lo largo de los seis ejes principales del sistema de

coordenadas cartesianas y contienen elementos booleanos que representan la información de interferencia. Sin embargo, la representación matricial de Dini y Santochi [20] ha sido la de mayor utilización en investigaciones posteriores.

Otros estudios tratan algoritmos para determinar la interferencia entre las piezas del ensamble. Baldwin [23] propuso varios métodos y herramientas informáticas para la generación de las secuencias de ensamble mecánico. Estos métodos solo consideraban las interferencias de las piezas del ensamble causadas por condiciones de contacto, con las cuales la herramienta informática generaba posibles subensambles y realizaba consultas de la forma sí/no a un usuario para registrar su factibilidad. Sin embargo, Baldwin [23] recomienda enriquecer sus métodos para automatizar completamente la generación de las secuencias de ensamble, de tal manera que las consultas planteadas al usuario fueran respondidas automáticamente al considerar la geometría del ensamble. Como resultado, propone comprobar la interferencia geométrica causada por la interferencia global de la pieza por medio de tres técnicas diferentes: prueba de interferencia volumétrica, prueba de interferencia de emisión de rayos y prueba de interferencia de proyección de superficies.

Sambhoos, Koc y Nagi [24] utilizaron un algoritmo de emisión de rayos para identificar las relaciones de contacto entre piezas clasificadas en dos tipos: indirecta e interferencia. La información geométrica era obtenida de un fichero con un formato basado en el Estándar para el intercambio de los datos del modelo del producto –en idioma inglés Standard for the Exchange of Product model data (STEP) [25-30]. La Organización Internacional de Normalización (ISO) ha desarrollado el STEP para habilitar el intercambio de los datos de un producto a través de su ciclo de vida. La especificación de los datos de STEP está representada por los Protocolos de Aplicación –en inglés Application Protocol (AP). Entre los APs para el intercambio de los ficheros CAD se encuentran el AP203 y AP214 los cuales están centrados principalmente en la representación de la información geométrica en formato de Representación de frontera – en inglés Boundary Representation (BRep)– de las piezas y los ensambles mecánicos. Además, STEP Part 21 especifica un formato de intercambio con el uso de una codificación en texto plano de los datos del producto.

Pan, Smith y Smith [31] propusieron un método computacional para extraer la información

geométrica de un fichero STEP, analizar las relaciones de interferencia entre las piezas del ensamble, y generar seis matrices donde se representaban las relaciones de interferencia entre las piezas en las direcciones de ensamble de los ejes principales de coordenada (+x, -x, +y, -y, +z, -z). Sin embargo, este método solo manipula piezas prismáticas y cilíndricas, y no considera piezas con superficies de forma libre.

El método que ha sido muy utilizado en el contexto de la detección de las colisiones en tiempo real entre objetos se denomina Detección de interferencia múltiple, y permite reducir el problema de detección de colisiones a múltiples llamadas a pruebas de interferencia estática [32]. La interferencia entre los objetos se comprueba en un lapso de tiempo determinado. Jiménez, Thomas, y Torras [32] apuntan que si este lapso de tiempo es muy grande podría no detectarse una colisión, y si es muy pequeño, podría ser costoso computacionalmente, pues se generarían demasiadas comprobaciones de interferencia. Idealmente, el próximo lapso de tiempo debe ser el menor tiempo en el que puede ocurrir una colisión. Existen varias estrategias [32] que estiman este tiempo de diferentes maneras, lo que disminuiría la cantidad de comprobaciones de interferencia.

Pan, Smith y Smith [31] utilizan un método diferente para detectar las colisiones entre dos piezas. En su estudio, la existencia de una interferencia es comprobada si las proyecciones de dos piezas en un plano perpendicular a una dirección principal de ensamble se solapan. Además, para analizar el problema de planificación del ensamble, afirman que no es necesario utilizar el método de Detección de interferencia múltiple, pues fue diseñado para detectar las colisiones en tiempo real de objetos en movimiento en un entorno de simulación física. Sin embargo, en la vida real, el proceso de ensamble está sujeto a restricciones técnicas originadas por la estabilidad.

Las restricciones geométricas son frecuentemente estudiadas al asumirse que las fuerzas no están presentes, como es el caso de Pan, Smith y Smith [31]. Sin embargo, la utilización del método de Detección de interferencia múltiple podría utilizarse para permitir el análisis de las relaciones de interferencia en ambientes donde las piezas o subensambles estén sometidos a la acción de varias fuerzas para determinar su estabilidad en el ensamble.

Las matrices de interferencia representan la restricción del movimiento de un componente por otros componentes en una dirección específica.



Figura 1.6. Piezas A y B compuestas por superficies complejas.

La matriz de interferencia correspondiente a la figura 1.6 al comprobarse las relaciones de interferencia en la dirección de desensamble +z es la siguiente:

$$F_{+z} = \begin{array}{c|cc} & A & B \\ \hline A & 0 & 1 \\ B & 0 & 0 \end{array}$$

En esta representación, F_{+z} significa que el movimiento de la pieza A está restringida en la dirección +z por la pieza B; el movimiento del componente B no está restringido por la pieza A en la dirección +z. Esta matriz contiene la información completa, porque la matriz de interferencia que se refiere a la dirección opuesta es la transpuesta de la original [11]. Una investigación más detallada y abarcadora relacionada con la factibilidad geométrica del ensamble es la desarrollada por Yu y Wang [33]. En su estudio, ellos propusieron la matriz de interferencia extendida para determinar la factibilidad geométrica en las direcciones de (des)ensamble, donde las direcciones son extendidas a los ejes del sistema de coordenada local en cada componente. Para analizar la factibilidad geométrica del componente que debe ser ensamblado con una trayectoria multidireccional, propusieron la matriz de interferencia de giro compuesta por una estructura lo

suficientemente compacta como para describir la información de interferencia desde un punto de giro. Además describen los métodos para la obtención automática de las dos matrices, y presentaron un algoritmo para discriminar la factibilidad geométrica en la planificación de la secuencia de (des)ensamble.

1.2.5 Generación de las secuencias de ensamble

Existen numerosos métodos desarrollados para generar múltiples secuencias de ensamble.

El método de Sekiguchi [34] genera una única secuencia de ensamble factible a partir del dibujo de ensamble y un fichero de entrada que contiene la pieza y la clasificación de los features. Ko y Lee [13] extienden el método de Sekiguchi para generar múltiples secuencias de ensamble de una base de datos CAD. Frommherz y Hornberger [35] también desarrollaron un método para generar múltiples secuencias de ensamble. Su método utiliza la información básica de la geometría del ensamble en formato BRep sin esquemas de clasificación. Wolter [36] propuso otro método para generar múltiples secuencias de ensamble similar al de Frommherz y Hornberger. Miller y Hoffman [37] propusieron un método para generar una sola secuencia de ensamble que está basado en la utilización de los dispositivos de sujeción para determinar una secuencia de ensamble factible.

Aunque los métodos anteriores permiten generar una o varias secuencias de ensamble, aún son incompletos, por lo que se hace necesario obtener el conjunto de todas las secuencias de ensamble factibles.

Bourjault [10] propuso un método para generar todas las secuencias de ensamble factibles, el cual utiliza un sistema de dos preguntas en la forma de “sí o no” y toma como base el grafo de conexiones del ensamble del producto. Las preguntas se hacen por pares y tienen la siguiente forma:

P1) ¿Puede la conexión L_i establecerse si las conexiones L_j, L_k, \dots han sido establecidas anteriormente?

P2) ¿Puede la conexión L_i establecerse si las conexiones L_j, L_k, \dots no han sido establecidas anteriormente?

donde L_i, L_j, L_k, \dots son las conexiones (reales o virtuales, denominadas “liaisons”) en el grafo de conexiones.

Con una cantidad L de conexiones existe un máximo teórico de 2^L estados de conexiones. Como promedio, solo la mitad de las conexiones se establecen en un estado, lo que conlleva a un máximo teórico de $L \cdot 2^{L-1}$ preguntas al diseñador.

A partir de las respuestas a estas preguntas se obtiene como resultado las condiciones de acoplamiento para todas las conexiones entre las piezas del producto y de las cuales se generan el conjunto de todas las secuencias de ensamble factibles.

Aunque las respuestas a estas preguntas sea relativamente sencilla, aún es una tarea que consume mucho tiempo y puede ser casi imposible de realizar si se incrementa el número de conexiones L entre las piezas del ensamble.

De Fazio y Whitney [16] desarrollaron un método que también utiliza el diagrama de conexiones para modelar el ensamble. Formularon dos nuevas preguntas con el propósito de disminuir a $2 \cdot L$ la cantidad de preguntas realizadas. Estas tienen la siguiente forma:

P1) ¿Qué conexiones deben hacerse antes de hacer la conexión L_i ?

P2) ¿Qué conexiones quedan por hacerse después de hacer la conexión L_i ?

Las respuestas a estas preguntas son expresadas en la forma de relaciones de precedencia entre las conexiones o combinaciones lógicas de las conexiones, a partir de las cuales se generan las secuencias de ensamble.

Sin embargo, como desventaja del método, la reducción de las preguntas es a expensas de una mayor dificultad al responderlas.

Homem de Mello y Sanderson [4] propusieron un método diferente para la generación de las secuencias de ensamble basado en un nuevo modelo para representar el producto mecánico. Ellos transformaron el problema de generar las secuencias de ensamble en el problema de generar las secuencias de desensamble al considerar que las tareas de desensamble son las inversas de las tareas de ensamble factibles.

El método determina las secuencias mediante la descomposición del ensamble al usar

los “cutsets” del grafo de conexiones con cada descomposición factible que corresponde a las ramificaciones en una representación de grafo AND/OR de las secuencias de ensamble.

La descomposición del ensamble comienza con los “cutsets” del grafo de conexiones completo. Cada cutset representa una descomposición a la cual se le comprueba su factibilidad. Esto equivale a realizar una pregunta sí/no acerca del desensamble de dos subensambles. Los dos subensambles resultantes de cada descomposición son candidatos a ser descompuestos al utilizar los cutsets de cada subgrafo de subensamble. La descomposición termina cuando se obtienen las piezas individuales.

Como desventaja del método está el tiempo y la dificultad que requiere el usuario para definir el grafo de conexiones con los tipos de contactos, los tipos de dispositivos de sujeción y los contactos virtuales; estos últimos constituyen un gran esfuerzo para el usuario, pues tiene que encontrar en el ensamble todos los pares de piezas que no contactan pero que interfieren entre sí durante el desensamble.

Baldwin [23] propuso tres nuevos métodos para la generación de las secuencias de ensamble centrados principalmente en reducir drásticamente el número de preguntas realizadas al usuario, lo que resulta en una mayor automatización. El primero es el “método de Bourjault modificado” basado en el trabajo de Bourjault [10]. El segundo es el método Cut-set y el tercero es el método “onion skin”, ambos basados en el trabajo de Homem de Mello y Sanderson [4].

La idea de la reducción de las preguntas está basada fundamentalmente en la formulación de dos reglas. La primera es la regla del *superconjunto* (en inglés, *superset*) que plantea que si un componente particular no puede ser extraído de un subensamble dado, la adición de más componentes a ese subensamble no cambia su situación. La segunda regla es la del *subconjunto* (en inglés, *subset*), la cual plantea que si un subensamble hijo puede ser extraído de un subensamble padre, este subensamble hijo también puede ser extraído de un subensamble compuesto por un subconjunto del ensamble padre, el cual debe contener todos los componentes del subensamble hijo.

Para ilustrar esas reglas se hace el siguiente análisis.

El componente *batería* en la figura 1.3 (a) no puede ser extraído del subensamble *tapa-cuerpo-batería* en la dirección Z+ debido a razones geométricas. Por lo tanto, este

tampoco puede ser extraído en esa dirección de *tapa-cuerpo-batería-cabeza*, el cual es un *superconjunto* de *tapa-cuerpo-batería*.

El componente *cabeza* puede ser extraído de *tapa-cuerpo-batería-cabeza* en la dirección Y+. Por lo tanto, este también puede ser extraído en esa dirección de los subensambles *subconjuntos cuerpo-batería-cabeza* o *cuerpo-cabeza* o *batería-cabeza* o *tapa-cuerpo-cabeza* o *tapa-batería-cabeza*.

Los métodos anteriores para generar todas las secuencias de ensamble factibles constituyeron un gran avance teórico a partir del cual otros investigadores se inspiraron para desarrollar nuevos métodos.

Para obtener los planes de ensamble detallados, Ben-Arieh y Kramer [38] presentaron una metodología y algoritmos para generar de forma consistente todas las secuencias de ensamble factibles al considerar las diferentes combinaciones de las operaciones de subensamble. Latombe et al. [39] expandieron el problema de la secuenciación del ensamble de una pieza con geometría nominal a las piezas que tienen geometría con tolerancia.

Kunica y Vranjes [40] describieron un planificador de ensamble automatizado que incluye la generación de las secuencias de desensamble. Este planificador comienza con una secuencia de desensamble inicial basada en reglas heurísticas, la cual es subsecuentemente modificada después de realizar experimentos de simulación con un prototipo virtual.

1.2.6 Evaluación y Selección de las secuencias de ensamble

Los métodos mencionados están dirigidos fundamentalmente a la generación de las secuencias de ensamble factibles al considerar las restricciones topológicas y geométricas entre las piezas en el ensamble.

Ye y Urzi [41] clasifican las restricciones del ensamble en fuertes y suaves. Las restricciones *fuertes* están basadas en las características topológicas y geométricas del ensamble. Las restricciones *suaves* se pueden obtener de la información referente a: el peso de la pieza, el tamaño de la pieza, la complejidad de la pieza, el material de la pieza,

la orientación de la pieza, el cambio de la herramienta, el costo, entre otros.

Basado en los criterios para las restricciones *suaves* propuestos por [41], Kunica y Vranjes [40] plantearon que la secuencia de (des)ensamble puede ser generada al utilizar diferentes criterios como:

- los volúmenes, las superficies o las masas de la pieza (si el volumen, superficie o masa de una pieza es mayor, esta es una mejor candidata para ser ensamblada primero),
- el número de relaciones entre las piezas (si la pieza tiene más relaciones, esta es una mejor candidata para ser desensamblada después),
- las distancias entre los centros de gravedad de las piezas y el centro de gravedad del producto (si el centro de gravedad de la pieza está a una mayor distancia del centro de gravedad del producto, esta es una mejor candidata para ser desensamblada primero),
- los valores de la componente y del centro de gravedad de la pieza (si la componente y del centro de gravedad de la pieza es la más baja, esta pieza es una mejor candidata para ser desensamblada después).

Aún son muchas las alternativas generadas para ensamblar un producto y además se hace necesario evaluarlas al considerar otras restricciones del proceso de ensamble con el objetivo de obtener secuencias óptimas que contribuyan de forma general a disminuir los tiempos y los costos del ensamble. Entre estas restricciones pueden estar las direcciones y herramientas del ensamble, la accesibilidad, la estabilidad y la seguridad [42].

Los cambios de las direcciones de ensamble y las herramientas de ensamble en el proceso del ensamble tienen un gran efecto en el costo y la eficiencia del ensamble.

La accesibilidad puede estar presente cuando el ensamble de una pieza o subensamble al resto del conjunto ensamblado no esté imposibilitado a causa de la interferencia entre las piezas y las herramientas. Por ejemplo, en la reparación de un ensamble es necesario poder acceder a la pieza defectuosa, en esta operación se tiene en cuenta la herramienta utilizada y la pieza que se quiere extraer.

También la estabilidad se considera como uno de los factores cruciales para la operación

del ensamble de la mayoría de los productos mecánicos. El número de los dispositivos de fijación y de las herramientas se reducen si los subensambles ensamblados son estables. Como resultado, se disminuyen el costo o el tiempo del ensamble. Se puede definir un subensamble como estable si sus componentes mantienen su posición relativa y no rompen sus contactos de forma espontánea [43], es decir, cuando las piezas que se ensamblan para formar un subensamble al finalizar la operación de ensamble mantienen su posición final. Boneschanscher [44] define tres tipos de estabilidad del ensamble: la estabilidad gravitacional, la estabilidad del ensamble y la estabilidad de movimiento. La estabilidad del ensamble es causada por la gravedad y las fuerzas de apoyo o las fuerzas del engrampaje sin considerar las fuerzas adicionales resultantes de la aceleración de la operación de ensamble. Basados en el trabajo de Lee [45], Wang y Liu [46] categorizaron la estabilidad fortalecida por los conectores del ensamble en tres niveles de acuerdo a la fortaleza de la conexión:

- Fuerte: el ajuste forzado o apretado, el ajuste por interferencia, el atornillado o enroscado, el remachado y la soldadura.
- Débil: el ajuste de despejo, el pegado y la fijación por enganche.
- Conexión inestable: el acople de superficie, y otros ajustes sin la aplicación de fuerzas.

Si un subensamble es estable, o sea, si las posiciones relativas de sus piezas permanecen incambiables cuando estas son manipuladas, se puede garantizar la seguridad de las operaciones de ensamble.

Ciertas investigaciones han estado enfocadas a generar planes de ensamble que cumplan con algunas de las restricciones anteriores.

Homem de Mello [47] propuso tres criterios: (1) está basado en una combinación de la complejidad de las operaciones de ensamble y en la estabilidad de los subensambles intermedios, (2) seleccionar los planes de ensamble mediante la maximización de la flexibilidad de la secuenciación de las operaciones de ensamble y (3) seleccionar los planes de ensamble mediante la minimización del tiempo de ensamble a través de la ejecución paralela de las operaciones de ensamble.

Laperriere y ElMaraghy [48] propusieron cuatro criterios: reorientaciones, paralelismo, estabilidad y clustering; en su sistema GAPP. Motavalli e Islam [49] emplearon dos criterios, el tiempo de ensamble total y el número de reorientaciones, en la evaluación de la secuencia de ensamble. Dos criterios fueron combinados al utilizar la teoría de utilidad multiatributo para derivar una única función objetivo. Su et al. [50] propusieron cuatro criterios de evaluación: el ángulo de ensamble, la dirección de ensamble, la reorientación y la estabilidad. Para seleccionar la secuencia óptima, se construyó una función de evaluación comprensiva mediante la integración de los cuatro criterios y los pesos son cuantitativamente ubicados al tener en cuenta la teoría del conjunto difuso, el análisis de clustering, y la teoría de la entropía.

Mohd et al. [51] realizaron un resumen acerca de la optimización de ASP y ALB con la utilización de enfoques basados en algoritmos de inteligencia artificial en el período 2000-2010. Ellos identificaron la frecuencia de los objetivos utilizados para la optimización del ASP en 39 investigaciones realizadas en ese período (Figura 1.7). El objetivo de optimización más utilizado es minimizar los cambios de dirección del ensamble. Luego le siguen minimizar el número de los cambios de herramienta, minimizar el cambio de tipo de ensamble, minimizar la complejidad del ensamble, minimizar la similitud del conector, maximizar la estabilidad del ensamble, minimizar la penalidad de la restricción geométrica, minimizar el costo del ensamble, minimizar el tiempo del ensamble y minimizar la distancia de desplazamiento de la herramienta del ensamble.

1.3 Técnicas de optimización avanzadas para la planificación de la secuencia de ensamble

De forma general las investigaciones hasta finales del pasado siglo estuvieron enfocadas principalmente en la obtención de planes de ensambles óptimos para productos con un número reducido de componentes. La estrategia principal empleada estaba basada en algoritmos de búsqueda/optimización sobre las representaciones de grafos de ensamble. No obstante, realizar una búsqueda exhaustiva es la estrategia más simple que asegura

la exploración de todo el espacio de búsqueda, pero no se hace práctica para ensambles con estructuras complejas aunque se utilicen técnicas heurísticas debido a la explosión combinatoria cuando el número de piezas se incrementa.

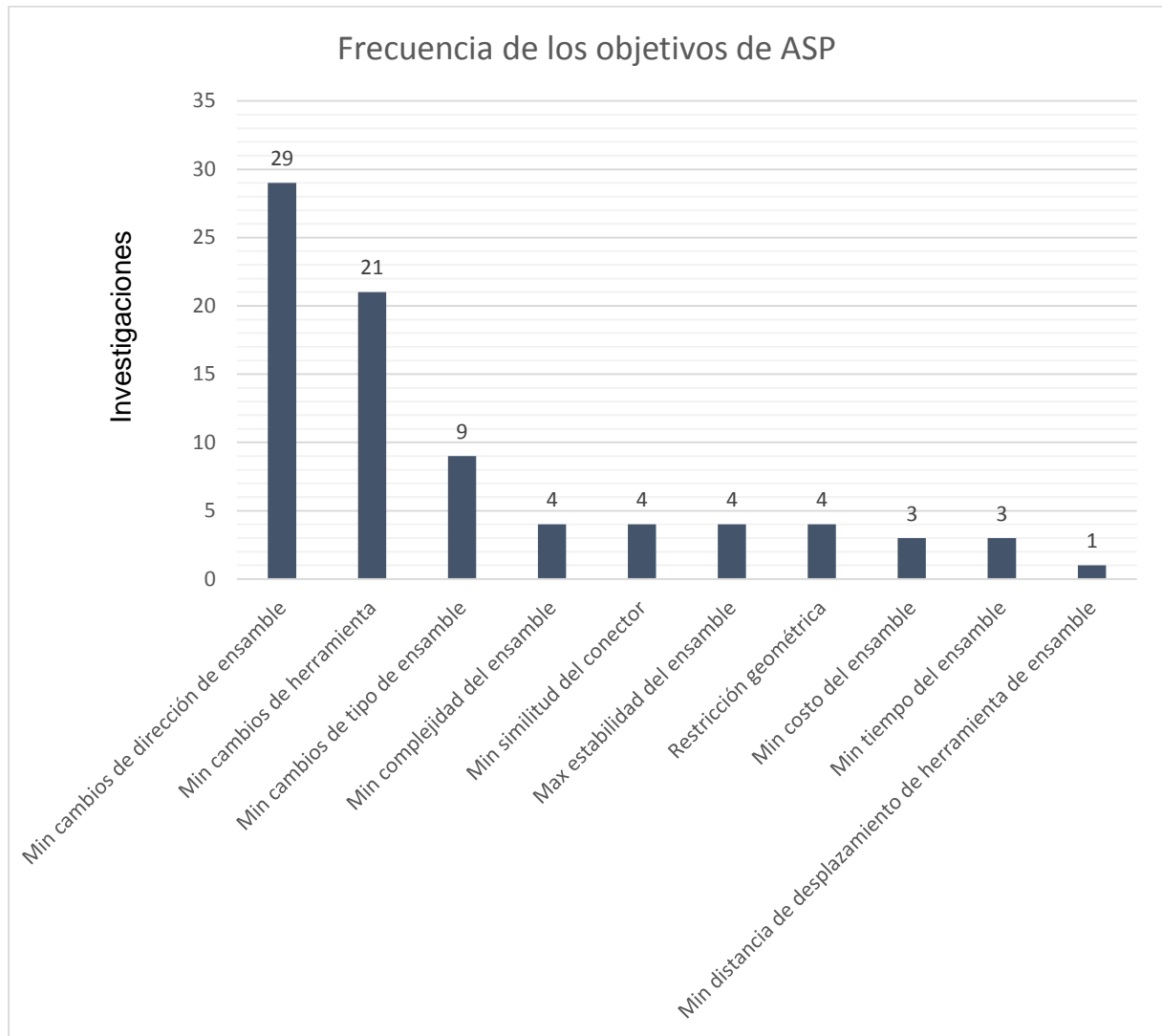


Figura 1.7. Frecuencia de los objetivos de ASP. Fuente [51].

La ASP está clasificada como un problema NP-Duro [12, 16] . En el caso de un ensamble compuesto por cinco piezas, el número posible de soluciones o variantes es de $5!$, o sea, 120 soluciones. Cuando el número de piezas aumenta a diez el número de soluciones se incrementa considerablemente a $10! = 3\,628\,800$. Además existen ciertas restricciones del problema que se deben tener en cuenta cuando se generan las secuencias de ensamble.

En las últimas dos décadas los algoritmos de optimización basados en inteligencia artificial han sido aplicados con éxito al problema ASP. Los de mayor utilización han sido los algoritmos genéticos –en idioma inglés Genetic Algorithm (GA)–, la optimización por colonia de hormigas –en idioma inglés Ant Colony Optimization (ACO)– y la optimización por enjambre de partícula –en idioma inglés Particle Swarm Optimization (PSO).

1.3.1 Algoritmo genético

Los GAs son una clase muy popular de los Algoritmos evolutivos. Diferentes escuelas de los Algoritmos evolutivos han surgido de forma independiente durante los pasados 40 años: los algoritmos genéticos, las estrategias de evolución, desarrollada por I. Rechenberg y H-P. Schwefel, y la programación evolutiva por L. Fogel. Más tarde, al final de la década del 80, la programación genética fue propuesta por J. Koza. Cada una de ellas constituye un enfoque diferente, no obstante, están inspiradas por los mismos principios de la teoría de la evolución natural.

Los GAs fueron desarrollados por J. Holland en los 1970s (Universidad de Michigan, EUA) para entender el proceso adaptativo de los sistemas naturales.

Un GA típicamente aplica un operador de cruzamiento a dos soluciones que juegan un papel fundamental, más un operador de mutación que aleatoriamente modifica la información del individuo para promover la diversidad. Los GAs utilizan una selección probabilística que originalmente es la selección proporcional. El reemplazamiento (selección del superviviente) es generacional, es decir, los padres son reemplazados sistemáticamente por la descendencia. El operador de cruzamiento se basa en el cruzamiento uniforme mientras que la mutación es de tipo bit-flipping. Se aplican las probabilidades p_m y p_c para la mutación y el cruzamiento respectivamente.

Los componentes de búsqueda principales para diseñar un algoritmo evolutivo son los siguientes [52]:

1. Representación: En la comunidad de los algoritmos evolutivos (especialmente los GAs), la solución codificada se refiere al cromosoma mientras que las variables de decisión dentro de una solución (cromosoma) son los genes. Los valores posibles de las variables (genes) son los alelos y la posición de un elemento (gen) dentro

de un cromosoma se llama locus.

2. Inicialización de la población: es un componente de búsqueda común para todas las metaheurísticas poblacionales.
3. Función objetivo: Todas las metaheurísticas tienen en común este componente de búsqueda. En la comunidad de los algoritmos evolutivos, el término aptitud se refiere a la función objetivo.
4. Estrategia de selección: La estrategia de selección está dirigida a la siguiente pregunta: ¿Qué padres son seleccionados para la próxima generación con preferencia hacia una mejor aptitud?
5. Estrategia de reproducción: La estrategia de reproducción consiste en diseñar adecuados operadores de cruzamiento y mutación para generar nuevos individuos (descendencia).
6. Estrategia de reemplazo: Los nuevos descendientes compiten con los individuos viejos por su lugar en la próxima generación (supervivencia del más apto).
7. Criterio de parada: Es un componente común para todas las metaheurísticas. Algunos criterios de parada son específicos para las metaheurísticas poblacionales.

En el problema ASP, las secuencias de ensamble son codificadas en cromosomas. Cada cromosoma corresponde a un individuo de una población de secuencias de ensamble. Los operadores genéticos se aplican a estos cromosomas para producir la descendencia más apta, de acuerdo a algún criterio de optimización. Los GAs se ejecutan sobre una población inicial de planes arbitrarios de ensamble, y finalizan –después de un número de generaciones– con un conjunto de buenos planes –óptimos o cercanos al óptimo. El proceso es estocástico: elecciones aleatorias, con ciertas probabilidades fijas, pueden ser hechas al aplicar los operadores genéticos (el gen que muta, los padres seleccionados para el cruzamiento) o en el orden que esas elecciones se aplican [14].

Bonneville et al. [53] fueron los primeros que introdujeron el concepto del GA en la ASP. En este estudio, los expertos de ensamble generan las secuencias de ensamble factibles para un conjunto inicial de una secuencia de ensamble. El cruzamiento se aplica sistemáticamente al par de individuos más aptos, pero la mutación se aplica aleatoriamente a la descendencia. Mientras que en Sebaaly et al. [54] la reproducción, el cruzamiento y la mutación se aplican en una manera cíclica, pero los cónyuges para el

cruzamiento se seleccionan aleatoriamente.

Chen y Liu [1] propusieron un planificador de ensamble multinivel basado en GA. La configuración de la probabilidad del operador genético, generada en un GA del primer nivel, es actualizada dinámicamente en un GA de segundo nivel cuyos cromosomas codifican la probabilidad de aplicar cada operador en el GA de primer nivel. Esto evita el problema de la selección de la frecuencia de mutación adecuada, la cual es un compromiso entre la convergencia prematura al mínimo local (calidad de solución final baja) y la frecuencia de convergencia general (tiempos de ejecución extensos).

En Marian et al. [55] se incluyen tipos de planes de ensamble adicionales, típicos de algunos procesos de ensamble, como los no secuenciales, no lineales, no monótonos y pseudo no coherentes, mediante la codificación en los cromosomas de no sólo las secuencias de ensamble sino de genes denominados Entidades Significativas para la Secuencia de Ensamble (EMAS), las cuales incluyen piezas únicas, subensambles completos, conjuntos de piezas para ser posicionadas simultáneamente, operaciones sin la adición de una nueva pieza, dispositivos de fijación auxiliares, etc. La población inicial de ensambles factibles se genera mediante la búsqueda guiada en la tabla de conexiones. Esta tabla codifica las restricciones de precedencia intrínsecas, las extrínsecas (derivadas a partir de las restricciones del proceso y de la accesibilidad) se utilizan para seleccionar entre las conexiones candidatas. En cuanto al proceso de optimización, igualmente en los operadores de cruzamiento y de pseudo-mutación la búsqueda guiada es utilizada de nuevo para garantizar la factibilidad de los cromosomas resultantes.

Un método basado en GA fue propuesto por Choi et al. [56] para optimizar las secuencias de ensamble. El objetivo es generar la secuencia de ensamble óptima para minimizar el tiempo de producción y el costo con un conjunto de tiempos de ensamble y un número de reorientaciones dados.

Tseng y Tang [57] utilizaron un GA para buscar los tipos de estaciones adecuados para el orden secuencial de las tareas de ensamble basadas en el conector. Tseng et al. [58] presentaron un modelo de planificación de secuencia de ensamble multi-planta con el uso de un método de GA para integrar la planificación de secuencia de ensamble con la asignación de planta. El método GA tiene el objetivo de minimizar el total de los costos

operacionales de ensamble y los costos de multi-planta.

Tseng et al. [59] propusieron un modelo integrado para la planificación de la secuencia de ensamble y desensamble. Se presentó un método de GA de dos bucles para buscar las soluciones buenas de las secuencias de ensamble y desensamble integradas. El bucle externo encuentra las secuencias de ensamble, mientras el bucle interior utiliza el orden inverso de la solución de la secuencia de ensamble como la solución inicial de la secuencia de desensamble para encontrar buenas secuencias integradas. Se formuló una función de aptitud mediante la integración de los costos de ensamble y los costos de desensamble.

Jin et al. [60] presentaron un método de optimización de la secuencia de ensamble para los productos mecánicos complejos mediante el empleo de un grafo dirigido y una matriz de ensamble para representar la relación de ensamble. La calidad dimensional de ensamble para diferentes secuencias fue evaluada al utilizar un método GA para obtener las secuencias de ensamble óptimas.

El GA en su forma básica tiende a generar descendencia no factible que viola la restricción de precedencia a causa de los operadores de cruzamiento y de mutación. Para manejar esta restricción, los investigadores han utilizado diferentes métodos como la penalidad y la estrategia de reparación. La penalidad está dada a los cromosomas que no son factibles debido a la violación de las restricciones de precedencia, lo que resulta en un valor de aptitud reducido. Por lo tanto, la oportunidad de seleccionar en la próxima generación un cromosoma no factible se reduce. La estrategia de reparación se utiliza en [61] para manipular la restricción de precedencia. En este enfoque, el cromosoma no factible es reparado y transformado en el cromosoma factible al utilizar un paso adicional en el GA.

Algunos investigadores han combinado el GA con otros algoritmos de optimización para mejorar su rendimiento.

Li et al. [62] combinaron la fortaleza de los GAs y la búsqueda tabú y propusieron un algoritmo genético mejorado con tabú para optimizar la secuencia de ensamble, mientras que la búsqueda tabú es utilizada como búsqueda local. Shan et al. [63] propusieron una combinación de GA con el recocido simulado para optimizar la secuencia de ensamble. Hui et al. [64] propusieron un método eficiente al combinar el sistema de colonia de

hormigas con un algoritmo genético. En Gao et al. [65] el algoritmo memético (MA-ASP) fue utilizado para resolver el problema ASP mediante la combinación de la naturaleza de la búsqueda global paralela de los algoritmos evolutivos con la búsqueda local para mejorar las soluciones de los individuos.

1.3.2 Optimización por Colonia de Hormigas

La idea básica de los algoritmos de optimización por colonia de hormigas (ACO) es imitar el comportamiento cooperativo de las hormigas reales para solucionar problemas de optimización. Las metaheurísticas ACO fueron propuestas por M. Dorigo en 1992. Estos algoritmos pueden ser vistos como sistemas multiagentes en los cuales cada agente está inspirado por el comportamiento de una hormiga real.

Failli and Dini [66] fueron los primeros en introducir ACO en el estudio de ASP y de la Planificación de la secuencia de desensamble –en idioma inglés Disassembly Sequence Planning (DSP)– y analizaron la influencia de cada parámetro del algoritmo.

El enfoque de Wang et al. [67] (ACO-ASP) está basado en la idea de que la reversión de las secuencias de ensamble son las secuencias de ensamble del producto. Los investigadores usaron ACO para generar la secuencia de ensamble y el objetivo de optimización utilizado está basado en las veces de los cambios de dirección. El ACO utilizado genera la secuencia de desensamble a partir de la cual se obtiene la secuencia de ensamble.

Yu y Wang [68] propusieron un método mejorado basado en ACO para solucionar el ASP para productos complejos que combina las ventajas del sistema de colonia de hormigas (ACS) y el sistema de hormiga max-min (MMAS) e integra algunas medidas de optimización.

1.3.3 Optimización por Enjambre de Partículas

La optimización por enjambre de partículas (PSO) se originó en 1995 por Kennedy y Eberhart. Es otra metaheurística poblacional estocástica inspirada a partir de la inteligencia colectiva. Esta imita el comportamiento social de los organismos naturales tales como las bandadas de pájaros y los cardúmenes de peces para encontrar un lugar

con suficiente alimento.

Al igual que el GA, el PSO original no es adecuado para ser aplicado directamente al problema ASP. Aparte del tema de la restricción de precedencia, el PSO original está diseñado para el problema continuo, donde la solución está en el espacio de valor real mientras que las soluciones de ASP residen en el espacio entero discreto.

Otro tema importante del PSO original es que fácilmente queda atrapado en el óptimo local. Para solucionar este problema, [69] introdujo un nuevo mecanismo de actualización de la velocidad mediante la utilización aleatoria de una de dos fórmulas en vez de una única fórmula. Además [46] introdujo un operador caótico para diversificar la posición de la partícula actualizada, el cual ayuda a reducir la convergencia prematura.

Xing y Wang [70] presentaron un híbrido de PSO y GA para generar las secuencias de ensamble. Como PSO está limitado por su convergencia pobre y cae fácilmente en la solución óptima local, y los operadores de cruzamiento y mutación del GA son útiles para converger a la solución óptima global, el método híbrido propuesto combina las ventajas de PSO y GA.

1.4 Conclusiones del capítulo

Existe una necesidad creciente para sistematizar y computarizar la generación de las secuencias de ensamble, por lo que se utilizan las tecnologías del CAD/CAM con las cuales se logran buenos resultados en la automatización de la planificación del ensamble.

Cuando se analizan productos compuestos por varios componentes y sujetos a restricciones del proceso de ensamble, es complicado determinar una secuencia factible u óptima, por lo que es necesaria la utilización de técnicas algorítmicas basadas en inteligencia artificial.

El método de optimización de los Algoritmos Genéticos es adecuado para lograr la planificación asistida por computadora de la secuencia de ensamble.

CAPÍTULO 2. MÉTODO ASISTIDO POR COMPUTADORA PARA LA OPTIMIZACIÓN DE LAS SECUENCIAS DE ENSAMBLE

El presente capítulo tiene como objetivo fundamental lograr la planificación asistida por computadora de las secuencias de ensamble de un producto mecánico. Para llevar a cabo este propósito, se hace necesario automatizar la obtención de los datos iniciales que constituyen las restricciones de un modelo de costo para la optimización de las secuencias de ensamble. Entre los datos iniciales considerados en el proceso se encuentran: la interferencia entre las piezas del producto en las seis direcciones principales de ensamble, la precedencia local del ensamble, la relación de estabilidad entre las piezas del ensamble y las herramientas de ensamble. Este capítulo se divide en dos partes. En la primera se propone un método asistido por computadora para la obtención automática de la información de interferencia entre las piezas del ensamble. En la segunda parte se describe un método de optimización de la secuencia de ensamble basado en los algoritmos genéticos que utiliza un modelo que incluye las restricciones del proceso de ensamble, y se aplica a un caso de estudio donde se obtienen buenos resultados que permiten validar el método.

2.1 Método para la obtención de la información de interferencia entre las piezas del ensamble

En esta parte se describe un método para la obtención de la información de interferencia entre las piezas del ensamble. Este método incluye las técnicas para la extracción de la información geométrica del ensamble contenida en un fichero CAD 3D con formato estándar STEP, la detección de las colisiones entre las piezas en las direcciones de desensamble y la generación de las matrices de interferencia. Para demostrar la eficacia del método se analizan dos ejemplos de ensambles de productos mecánicos y se obtienen como resultado las matrices de interferencia correspondientes a las seis direcciones principales de desensamble.

2.1.1 Descripción del método

El método propuesto está basado en la técnica de Detección de interferencia múltiple descrita en la sección 1.2.4 para detectar las colisiones entre las piezas. El método abordado manipula piezas al considerar cualquier superficie que las caracterice con el uso de una aproximación triangular según su descripción geométrica. En el anexo 1 se compara el método propuesto con otro descrito en la sección 1.2.4.

2.1.2 Materiales y métodos

Para el desarrollo de este método se contó con la información de las piezas pertenecientes a un ensamble, la cual previamente ha sido salvada en un fichero de formato ISO-10303, también conocido como formato STEP. Este formato ya es común en la mayoría de los sistemas CAD comerciales existentes hoy en día.

La información mencionada se procesó en una computadora Pentium IV con las siguientes características: procesador dual core con velocidad de 2.0 GHz, memoria RAM de 2GB, espacio de disco duro necesario para su ejecución de 500 MB. El sistema operativo en que se ejecutó la herramienta informática desarrollada fue Ubuntu Linux 12.04 LTS a 64 bits, aunque también se realizaron pruebas en el sistema operativo Windows XP a 32 bits.

El módulo desarrollado se programó en el lenguaje Python, versión 2.7.3, y se utilizaron las bibliotecas de OpenCascade v6.5, por medio de PythonOCC v0.6 como plataforma de desarrollo CAD/CAE/PLM 3D. Las funciones de intercambio de los datos CAD que implementa PythonOCC para extraer la información geométrica de los ficheros STEP Part 21 son utilizadas en una herramienta informática desarrollada con el lenguaje Python.

A continuación se presenta la metodología para la extracción de la información geométrica de las piezas relativas a un ensamble. Luego se describe el proceder para obtener las interferencias entre dichas piezas en las direcciones de los tres ejes principales de desensamble en la forma de matrices de interferencia que servirán como datos de entrada a una herramienta de planificación de (des)ensamble.

Extracción de la información geométrica de las piezas contenidas en un fichero STEP

Actualmente, varias herramientas CAD generan ficheros con el formato STEP Part 21 para garantizar que el intercambio de sus datos se ajuste a este estándar. Por esta razón, el método propuesto utiliza la información geométrica para detectar la interferencia entre las piezas, la cual es extraída de un fichero con este formato.

Al cargarse un fichero STEP con las funciones de PythonOCC hay que tener en cuenta que el ensamble es representado como una entidad topológica de tipo *Compound* y cada pieza es de tipo *Solid*. Estos tipos pertenecen a un tipo genérico denominado *Shape*.

Una vez que se haya cargado en la aplicación el fichero STEP, los datos extraídos se mapean automáticamente en estructuras de datos del OCE representadas con formato BRep. Entonces, cada pieza del ensamble representada por una forma de tipo *Solid* es almacenada en una lista de piezas P por la herramienta informática implementada. En la figura 2.1 se muestran de forma general los pasos para la extracción de la información geométrica (piezas) de un fichero STEP.



Figura 2.1. Pasos para la extracción de la información geométrica de un fichero STEP

Generación de la aproximación triangular de las superficies de las piezas

Una vez que se tenga la lista de piezas (P), se generan las aproximaciones triangulares de sus superficies (Figura 2.2). La técnica utilizada en este momento es el mallado de los sólidos (que representan las piezas) con el constructor de mallas para estructuras BRep del OCE. Hay que tener cuenta que una mayor precisión de la malla permite una aproximación más exacta a la superficie original de la pieza, pero influye en el aumento del costo computacional. Posteriormente se transforman estos datos a una estructura denominada *TrimeshData*. Para llevar a cabo esta tarea, se utiliza una función similar a `_compute_trimesh()` que se encuentra implementada en la clase *DynamicShape* del módulo DYN (simulación de cuerpos rígidos) del PythonOCC.

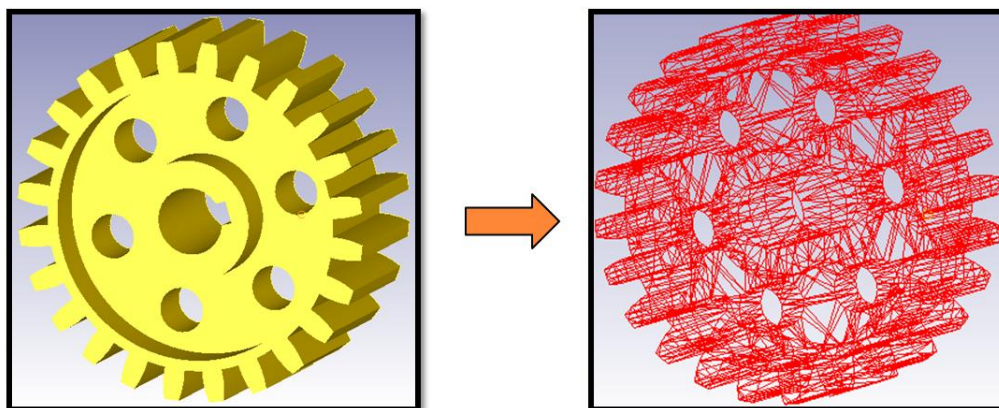


Figura 2.2. Aproximación triangular de las superficies de las piezas con el uso del mallado triangular (a) Pieza sólida (b) Malla triangular correspondiente a la pieza en (a)

Hay que destacar que la malla resultante de cada pieza está representada por una estructura que se utiliza en el próximo paso y que es resultado de aplicar transformaciones geométricas al considerar el centro de masa de cada pieza. Al terminarse el proceso de mallado de todas las piezas, las mallas resultantes se almacenan en la lista de mallas M .

Detección de las interferencias entre las piezas

La mayoría de las piezas son ensambladas a lo largo de los ejes principales en los sistemas de ensamble automatizados. Por lo que se hace necesario determinar si una pieza tiene una trayectoria libre de colisión al ser ensamblada o desensamblada en una dirección determinada. El método que se presenta analiza la interferencia entre las piezas a lo largo de los tres ejes principales de desensamble (+x, +y, +z). A partir de la lista M , que contiene el mallado superficial de la geometría de cada pieza, se comprueban las relaciones de interferencia entre cada par de mallas. Si una colisión existe entre las mallas M_i y M_j , donde M_i, M_j pertenecen a M , con $M_i \neq M_j$ (es decir, no se analiza la interferencia de una malla consigo misma), mientras M_i se mantiene estática (fija) y M_j se mueve en una de las direcciones principales de desensamble, entonces se registra la información de interferencia entre las dos piezas involucradas. Al finalizar, la información de interferencia entre todas las piezas es almacenada por cada dirección de desensamble en un conjunto de matrices de interferencia, para mantener la compatibilidad con algunas herramientas que utilizan esta información para la planificación de las secuencias de (des)ensamble.

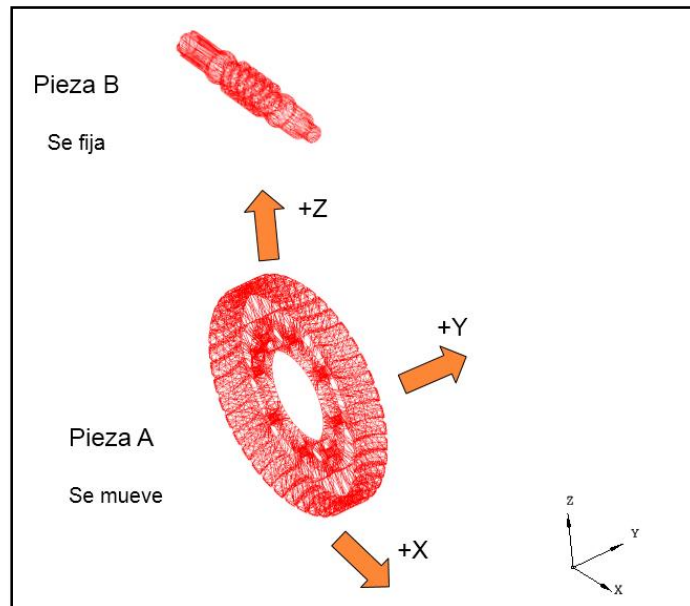


Figura 2.3. Detección de las interferencias entre las piezas

El módulo DYN del PythonOCC fue utilizado para la detección de las interferencias entre las piezas. Contiene una clase llamada *DynamicSimulationContext* que permite detectar la ocurrencia de colisiones entre las mallas triangulares. En este caso, se fija un objeto malla y se le aplica un movimiento (velocidad) a otro objeto malla en una dirección de desensamble. Se repite este proceso por cada par de mallas de las piezas en las tres direcciones principales de desensamble (figura 2.3). De esta manera se detecta la información de interferencia entre las dos piezas analizadas y es almacenada en las matrices de interferencia.

Creación de las matrices de interferencia

Para el caso de la matriz F_{+z} (sección 1.2.4), se analiza la interferencia entre las piezas A y B, al mantener fija la pieza B y mover la pieza A en la dirección de desensamble $+z$. Al detectarse una colisión, entonces se almacena el valor 1 en la segunda columna de la primer fila, sino se pone el 0 como valor. Los elementos de la diagonal principal de la matriz siempre van a ser igual a 0, porque no se analiza la relación de interferencia de una pieza consigo misma. Los pasos del método propuesto son resumidos en formato de pseudocódigo en la figura 2.4.

2.1.3 Resultados del método

El método presentado para la creación de las seis matrices de interferencia a partir de la información geométrica extraída de un fichero con formato STEP fue comprobado para dos ejemplos de ensamble que contienen piezas con superficies complejas.

Una herramienta informática fue desarrollada al utilizar la plataforma PythonOCC con la que se implementó el método propuesto. La herramienta informática requiere como dato de entrada un fichero STEP que pudo haber sido generado por una herramienta CAD comercial, y el resultado son las matrices de interferencia correspondientes a las seis direcciones de desensamble principales. La figura 2.5 muestra un ensamble de un disco de freno. La pieza 1 corresponde al rotor del disco de freno, que contacta con la pieza 2 que es el centro del disco de freno. Las piezas 3, 4, 5 son pernos que fijan el centro del disco de freno al rotor del disco de freno. Este ensamble fue simplificado para una mejor comprensión del análisis del caso. Primeramente se determinan las relaciones de

interferencia entre las piezas mientras cada una se desplaza en las tres direcciones de desensamble y luego se obtienen las siguientes matrices de interferencia:

+x	-x	+y	-y	+z	-z
[[0 0 1 1 1]	[[0 0 1 1 1]	[[0 0 1 1 1]	[[0 0 1 1 1]	[[0 1 1 1 1]	[[0 0 0 0 0]
[0 0 1 1 1]	[0 0 1 1 1]	[0 0 1 1 1]	[0 0 1 1 1]	[0 0 1 1 1]	[1 0 0 0 0]
[1 1 0 0 0]	[1 1 0 0 1]	[1 1 0 0 0]	[1 1 0 0 0]	[0 0 0 0 0]	[1 1 0 0 0]
[1 1 0 0 0]	[1 1 0 0 0]	[1 1 0 0 0]	[1 1 0 0 0]	[0 0 0 0 0]	[1 1 0 0 0]
[1 1 1 0 0]]	[1 1 0 0 0]]	[1 1 0 0 0]]	[1 1 0 0 0]]	[0 0 0 0 0]]	[1 1 0 0 0]]

Las matrices que representan a las direcciones -x, -y, -z fueron generadas al tener en cuenta que se corresponden con las transpuestas de las matrices +x, +y, +z. La información de la matriz de interferencia para la dirección de +z sugiere que la pieza 1 al desensamblarse en esa dirección está obstruida por las piezas 2, 3, 4 y 5, el movimiento de la pieza 2 está restringido por las piezas 3, 4 y 5, y ninguna pieza interfiere la extracción de las piezas 3, 4 y 5.

La figura 2.6 (b) muestra un segundo ejemplo de un ensamble de un reductor, el cual también fue simplificado. Este ensamble está compuesto por siete piezas. La pieza 1 es el tornillo de izaje o cáncamo, la pieza 2 es el cuerpo del reductor y la pieza 3 representa la tapa superior. La pieza 4 representa la rueda dentada conducida, la pieza 5 al árbol de salida del movimiento, la pieza 6 corresponde al piñón o rueda dentada conductora y la pieza 7 es el árbol de entrada del movimiento al piñón.

En este modelo 3D de ensamble puede notarse la complejidad superficial de la geometría que caracteriza a las piezas del reductor, y al analizarse con la herramienta informática que implementa el método propuesto se obtienen las matrices de interferencia de desensamble siguientes:

+x	-x	+y	-y	+z	-z
[[0 0 1 0 0 0 0]	[[0 0 1 0 0 0 0]	[[0 0 1 0 0 0 0]	[[0 0 1 0 0 0 0]	[[0 0 0 0 0 0 0]	[[0 1 1 1 1 0 0]
[0 0 0 1 1 1 1]	[0 0 0 1 1 1 1]	[0 0 0 1 0 1 0]	[0 0 0 1 0 1 0]	[1 0 1 1 1 1 1]	[0 0 0 0 0 0 0]
[1 0 0 1 1 1 1]	[1 0 0 1 1 1 1]	[1 0 0 1 0 1 0]	[1 0 0 1 0 1 0]	[1 0 0 0 0 0 0]	[0 1 0 1 1 1 1]
[0 1 1 0 1 1 1]	[0 1 1 0 1 0 0]	[0 1 1 0 0 0 0]	[0 1 1 0 1 0 0]	[1 0 1 0 1 1 0]	[0 1 0 0 1 1 0]
[0 1 1 1 0 1 1]	[0 1 1 1 0 0 0]	[0 0 0 1 0 0 0]	[0 0 0 0 0 0 0]	[1 0 1 1 0 0 0]	[0 1 0 1 0 0 0]
[0 1 1 0 0 0 1]	[0 1 1 1 1 0 1]	[0 1 1 0 0 0 1]	[0 1 1 0 0 0 0]	[0 0 1 1 0 0 1]	[0 1 0 1 0 0 1]
[0 1 1 0 0 1 0]]	[0 1 1 1 1 1 0]]	[0 0 0 0 0 0 0]]	[0 0 0 0 0 1 0]]	[0 0 1 0 0 1 0]]	[0 1 0 0 0 1 0]]

ALGORITMO: Relaciones de interferencia entre piezas del ensamble

ENTRADA: Modelo de ensamble CAD 3D con formato STEP

SALIDA: Matrices de interferencia F_{+x} , F_{-x} , F_{+y} , F_{-y} , F_{+z} , F_{-z}

COMENZAR:

/** Extracción de la información geométrica de las piezas **/

Extraer piezas del fichero STEP

Almacenar piezas en la lista P , número de piezas n

/** Generar la aproximación triangular de cada pieza **/

FOR $k \leftarrow 1$ to n {

$M[k] = \text{obtenerAproxTriangular}(P[k])$

}

/** Obtención de las interferencias entre las piezas **/

FOR $i \leftarrow 1$ to n {

FOR $j \leftarrow 1$ to n {

IF $i < j$ {

/** $F_{-x} = F_{+x}^T$, $F_{-y} = F_{+y}^T$, $F_{-z} = F_{+z}^T$ **/

IF M_i colisiona con M_j al moverse en la dirección de desensamble $+x$ {

$F_{+x}[i][j] = 1$

}ELSE { $F_{+x}[i][j] = 0$ }

IF M_i colisiona con M_j al moverse en la dirección de desensamble $+y$ {

$F_{+y}[i][j] = 1$

}ELSE { $F_{+y}[i][j] = 0$ }

IF M_i colisiona con M_j al moverse en la dirección de desensamble $+z$ {

$F_{+z}[i][j] = 1$

}ELSE { $F_{+z}[i][j] = 0$ }

}

}

}

FINALIZAR

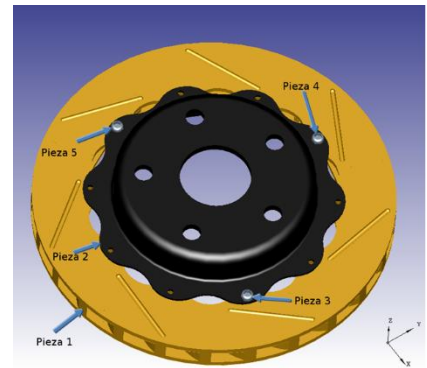
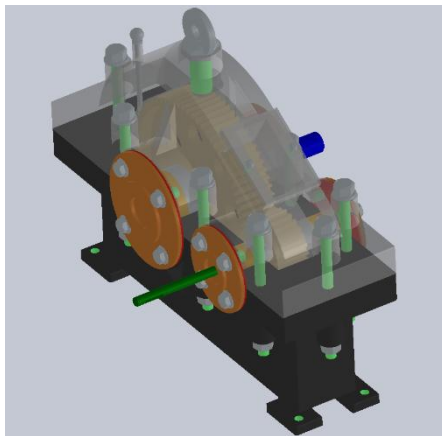
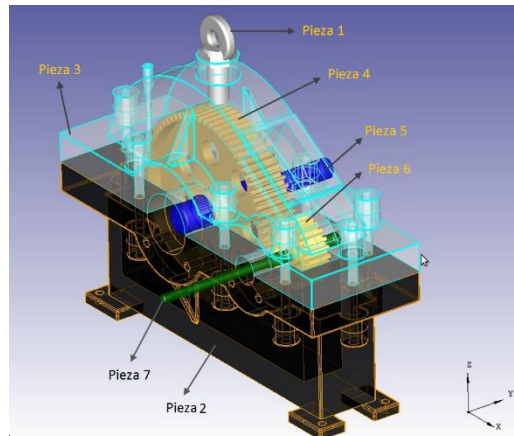


Figura 2.5. Ensamble simplificado de un disco de freno

Figura 2.4. Algoritmo para obtener las relaciones de interferencias entre las piezas de un ensamble



(a)



(b)

Figura 2.6. Ensamble de un reductor (a) Ensamble del reductor con todos sus componentes (b) Ensamble del reductor simplificado

2.1.4 Discusiones

Para detectar las interferencias entre todas las piezas de un ensamble, primeramente se realiza una aproximación triangular de la superficie de cada pieza mediante el mallado, y posteriormente se realizan pruebas de interferencias mientras la malla de una pieza se mantiene estática y la malla de otra se mueve en una dirección de desensamble hasta que ocurra o no una colisión. Las relaciones de interferencias entre las piezas se almacenan en matrices con una estructura ampliamente utilizada en la literatura, y sirven como base para la generación automatizada del conjunto de las secuencias de (des)ensamble al considerarse la geometría.

Los resultados consideran las relaciones de interferencia entre piezas de superficies prismáticas, cilíndricas y además con superficies de forma libre, lo que implica que se puede extender el análisis de interferencia a otros ensambles de productos con geometrías más complejas. Sin embargo, solo se toman en cuenta las direcciones de desensamble de los ejes principales.

La información contenida en las matrices de interferencia es importante para la generación automática de las reglas de precedencia de desensamble, y al considerar estas reglas se puede realizar la selección de las secuencias de (des)ensamble factibles de un producto para lograr su posterior planificación. Sin embargo, a partir de la información de estas matrices solo se puede discriminar si una secuencia de (des)ensamble es factible si se toma en cuenta la geometría de las piezas involucradas.

2.1.5 Conclusiones parciales

Se presentó un método para la obtención de las relaciones de interferencia entre las piezas con superficies prismáticas, cilíndricas y de forma libre de un ensamble a partir de la geometría contenida en un fichero con formato STEP. Para automatizar el método descrito, fue programada una herramienta informática para leer ficheros STEP, y generar las matrices de interferencia correspondientes a las direcciones de desensamble de los seis ejes principales. Los resultados obtenidos muestran la eficacia del método al

analizarse ensambles de productos compuestos por piezas con superficies complejas. La herramienta informática fue desarrollada sobre la plataforma libre y de código abierto PythonOCC, lo que implica que es una alternativa a los que se pueden crear al utilizarse herramientas privativas de código cerrado.

2.2 Método de optimización para la planificación de las secuencias de ensamble basado en algoritmos genéticos

En esta parte se presenta un método para la optimización de las secuencias de ensamble. Primeramente se describe un modelo de optimización que incluye varias restricciones que son frecuentes en el proceso de ensamble de un producto mecánico. Luego, se detalla el método GA-ASP que utiliza el modelo de optimización anterior y que está basado en los GAs. Finalmente, el método es aplicado a un caso de estudio y se obtiene como resultado las secuencias de ensamble optimizadas. Los resultados son comparados con los obtenidos por otros dos métodos y se concluye que el método propuesto produce secuencias de ensamble mejores bajo las restricciones del proceso de ensamble consideradas.

2.2.1 Modelo de Optimización

Se supone lo siguiente para simplificar el modelo de optimización:

1. Todas las piezas son rígidas, y no se considera el efecto de la tolerancia del ensamble.
2. Las direcciones del ensamble se restringen a +X, -X, +Y, -Y, +Z, -Z en los tres ejes de coordenadas ortogonales.
3. Una pieza se ensambla en una dirección y una herramienta se utiliza en un proceso de operación de ensamble.

Restricciones del proceso de ensamble

La idea de la optimización de la secuencia de ensamble es generar las secuencias factibles con un costo mínimo de ensamble mientras se satisfacen las restricciones del proceso de ensamble (figura 2.7). Al considerar las restricciones geométricas se eliminan las secuencias no factibles, por lo que se reduce considerablemente el espacio de búsqueda. Otras restricciones que disminuyen el costo del ensamble son la precedencia local del ensamble, la estabilidad del ensamble, el cambio de las direcciones de ensamble y el cambio de las herramientas de ensamble.

Modelo de costo del ensamble

Las secuencias de ensamble buenas u óptimas deben satisfacer las restricciones del proceso de ensamble. Aparte de satisfacer las restricciones geométricas, las secuencias deben tener el número mínimo de piezas que violen la precedencia local del ensamble, el número mínimo de piezas inestables, el número mínimo de los cambios de dirección de ensamble y de las herramientas de ensamble. El modelo utilizado permite evaluar cada secuencia al considerar las cinco restricciones del proceso del ensamble anteriores.

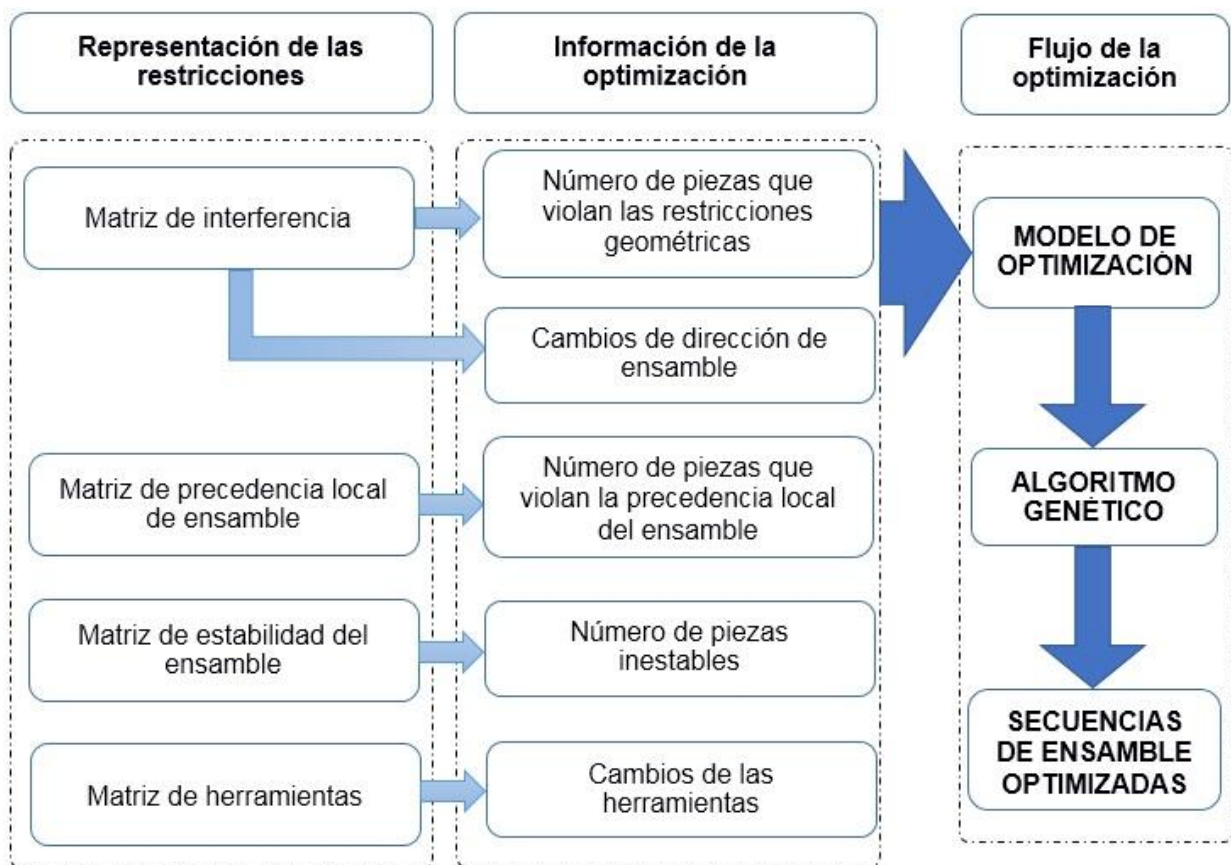


Figura 2.7. Restricciones del proceso de ensamble relacionadas con la optimización de la secuencia de ensamble.

El modelo de costo constituye la función de costo del ensamble y se puede definir como la siguiente función de aptitud:

$$\text{Min } F(s) = W_d * N_d(s) + W_p * N_p(s) + W_s * N_s(s) + W_t * N_t(s) + W_g * M \quad (1)$$

Donde

N_d : cambio de las direcciones de ensamble

N_p : número de piezas que violan la precedencia local del ensamble

N_s : número de piezas inestables

N_t : cambio de herramientas de ensamble

M : número de piezas que violan las restricciones geométricas

W_d, W_p, W_s, W_t, W_g : los pesos de las cinco restricciones del proceso de ensamble.

s : representa la secuencia de ensamble.

Para un ensamble compuesto por n piezas $s = (p_1, p_2, \dots, p_n)$ y p_i ($i=1, 2, \dots, n$) representa a cada pieza del ensamble.

Detección de las piezas que violan las restricciones geométricas

Del conjunto de las secuencias posibles en las que se puede ensamblar un producto mecánico, existe un gran número de ellas que no son factibles debido a las interferencias entre las piezas cuando son ensambladas en una dirección determinada. Lo anterior podría producir planes de ensamble inconsistentes, por lo que es necesario determinar las secuencias que solo son factibles geométricamente. Para representar las restricciones geométricas del ensamble se utiliza la matriz de interferencia. A cada dirección de ensamble que se considere para un producto, le corresponde una matriz de interferencia. En la sección 2.1 se describe un método asistido por computadora para la obtención de las matrices de interferencia al tomar como fuente de datos el modelo geométrico tridimensional del producto. Una vez que estas matrices son generadas automáticamente, se utilizan para determinar la factibilidad geométrica de cada secuencia de ensamble.

Para inferir si es posible ensamblar una pieza a un subensamble en la dirección +X, al tomar como fuente de información la matriz de interferencia IM_{+x} , se utilizan los siguientes pasos:

- 1) Se asume que los componentes $\{p_1, p_2, \dots, p_m\}$ han sido ensamblados antes que la pieza p_i .
- 2) Se define la siguiente fórmula:
$$V(p_i) = \sum_{j=1}^m IM_{p_j p_i}$$
- 3) Determinar si es factible o no ensamblar la pieza p_i en la dirección +X:
si: $V(p_i) = 0$, el ensamble de p_i al conjunto $\{p_1, p_2, \dots, p_m\}$ en la dirección +X es factible geométricamente; sino, no es factible.

Los pasos anteriores se repiten para cada una de las direcciones presentes en el ensamble del producto, y de esta manera se calculan las piezas que violan las restricciones geométricas en una secuencia de ensamble.

Detección de las piezas que violan la precedencia local del ensamble

Aunque algunas secuencias sean geométricamente factibles, pueden existir piezas que violen la precedencia local del ensamble de un producto mecánico. La cantidad de las piezas que violan la precedencia local del ensamble se clasifica como una restricción suave, en la que se pueden considerar los volúmenes, las superficies o las masas de la pieza, y la distancia entre el centro de gravedad del producto y el centro de gravedad de la pieza, entre otros criterios. Para determinar la precedencia local del ensamble se utiliza el criterio de la distancia entre el centro de gravedad de la pieza y el centro de gravedad del producto. Usualmente, las piezas que están ubicadas más al interior del producto deben ser ensambladas antes que las piezas que están posicionadas más hacia al exterior.

Para un ensamble compuesto por n piezas, la precedencia local del ensamble se representa como una matriz de precedencia $PM = [v_{ij}]_{n \times n}$.

Los valores de los elementos son definidos de la siguiente manera:

$v_{ij} = +1$, cuando la pieza p_i debe ser ensamblada antes de la pieza p_j .

$v_{ij} = -1$, cuando la pieza p_i debe ser ensamblada después de la pieza p_j .

$v_{ij} = 0$, cuando no existe precedencia local de ensamble entre la pieza p_i y la pieza p_j .

Cuando se ensambla el producto de la figura 1.5 puede ocurrir el siguiente escenario: Primero se ensambla la pieza 1. Luego la pieza 9 o la pieza 10 pueden ser ensambladas en el próximo paso. Se puede notar que al ensamblarse cualquiera de estas dos piezas en la dirección $-X$ no existe ninguna restricción geométrica entre estas dos piezas que invalide la operación de ensamble al moverse cualquiera de ellas hasta su posición final. Sin embargo, la pieza 9 debe ensamblarse antes de la pieza 10 debido a que existe una restricción de precedencia local. En este caso, la distancia del centro de gravedad de la pieza 9 al centro de gravedad del producto es menor que la distancia a la que se encuentra el centro de gravedad de la pieza 10 con respecto al centro de gravedad del producto. Esto permite conocer que la pieza 10 en su posición final de ensamble está más al exterior del producto con respecto a la pieza 9 que se encuentra más al interior, por lo que $v_{9\ 10} = 1$.

La matriz PM puede ser generada automáticamente al tomar como fuente de datos el modelo CAD del producto. Esto permite automatizar la obtención de la precedencia local del ensamble basado en el criterio mencionado, y a su vez, constituye una herramienta útil cuando se analizan productos más complejos y con una mayor cantidad de componentes.

A partir de la matriz PM se obtiene el número de piezas que violan la precedencia local del ensamble para cada secuencia. El número de piezas que violan las restricciones locales de precedencia en una secuencia de ensamble es calculado por $Np = \sum_{k=1}^m pr_k$, donde m son los pares de restricciones de precedencia entre dos piezas; $pr_k = 0$, si el par de piezas k th ($1 \leq j \leq m$) satisface la precedencia local del ensamble, sino $pr_k = 1$.

Detección de las piezas inestables

En el ensamble de un producto, frecuentemente se utilizan dispositivos de fijación para sujetar algunas piezas debido a la inestabilidad de estas, lo que permite que los subensambles permanezcan estables en el proceso de ensamble. Los dispositivos

auxiliares son reducidos si la estabilidad de las piezas de la secuencia de ensamble se mantiene, lo que trae como resultado una disminución del costo del ensamble.

La estabilidad por los conectores del ensamble se cataloga en tres niveles de acuerdo a la fortaleza de la conexión: fuerte, débil y conexión inestable. Los conectores del ensamble se consideran como dispositivos de fijación para ensamblar las piezas y son vistos como piezas no restrictivas y se excluyen de la secuencia de ensamble.

La estabilidad de una pieza p_j en su posición final de ensamble se determina si esta tiene un apoyo estable que puede ser otra pieza p_i . Si la pieza p_j está encima de la pieza p_i y la fuerza de gravedad de p_j pasa a través de la superficie de contacto de las dos piezas, se puede concluir que p_i es un apoyo estable para p_j . Por lo que al realizar la operación de ensamble de p_j a la pieza previamente ensamblada (p_i), no se necesita ningún dispositivo de fijación para sujetar la pieza p_j a p_i , y p_j permanece estable después de la operación. En la figura 1.5, la pieza 1 es un apoyo o soporte estable para la pieza 7.

Para un ensamble compuesto por n piezas, la estabilidad del ensamble entre dos piezas determinada por el apoyo estable o los conectores de ensamble se representa como una matriz de estabilidad $SM = [S_{ij}]_{n \times n}$, y los valores de S_{ij} se definen como:

$S_{ij} = 2$: cuando la pieza p_i es soportada por p_j o estas piezas están unidas por conexiones fuertes.

$S_{ij} = 1$: cuando las piezas p_i y p_j están unidas por conexiones débiles y no existe un apoyo estable entre ellas.

$S_{ij} = 0$: cuando las piezas p_i y p_j están unidas por acople de superficies sin la aplicación de fuerzas y débiles y no existe un apoyo estable entre ellas.

Se puede representar un subensamble estable temporal, compuesto por una cantidad de piezas m ya ensambladas, como $sub = \{p_1, p_2, \dots, p_m\}$. Si p_i se va a ensamblar en el próximo paso, su estabilidad con respecto a sub puede determinarse al tomar como fuente la información contenida en la matriz de estabilidad SM . El valor de estabilidad S_{ij}

$(1 \leq j \leq m)$ puede deducirse como $\{S_{i1}, S_{i2}, \dots, S_{im}\}$. En caso que algún valor de $S_{ij} = 2$ con $(1 \leq j \leq m)$, se concluye que la pieza p_i es estable al ensamblarse al conjunto *sub*, en cualquier otro caso ($S_{ij} = 1$ o $S_{ij} = 0$) se necesitan dispositivos de fijación para poder llevar a cabo el proceso de ensamble. De esta manera se puede determinar el número de piezas inestables de una secuencia de ensamble.

Los cambios de dirección y los cambios de las herramientas

Para mejorar la eficiencia del ensamble y disminuir su costo, es recomendable ensamblar en grupos las piezas en la misma dirección y con la misma herramienta.

Como ejemplo se puede mostrar que una vez ensamblada la pieza 1 (figura 1.5) las piezas 3, 2, 4, 5 se pueden ensamblar consecutivamente en la dirección +X y con la misma herramienta.

La información relativa a las direcciones de ensamble es determinada a partir de las matrices de interferencia, y como resultado se infiere el número de los cambios de dirección de ensamble para la secuencia evaluada.

El ensamble de una pieza puede realizarse con una o varias herramientas. Cuando se va a ensamblar una pieza, es usual tratar de seleccionar la misma herramienta utilizada en la operación de ensamble anterior, lo que permite que el tiempo de los cambios de herramientas disminuya en el proceso de ensamble. Las herramientas de ensamble se representan en una matriz de herramienta $TM = [t_{ij}]_{n \times m}$, donde n es el número de piezas y m es el número de herramientas que se utilizan para ensamblar la pieza correspondiente. Después de que una secuencia de ensamble es generada se determinan las herramientas que se van a utilizar y se calcula el número de los cambios de herramienta de ensamble.

2.2.2 Método de optimización

El método basado en los GAs fue elegido para optimizar el problema ASP. Esta metaheurística poblacional ha probado ser efectiva al solucionar problemas NP-Duros y se ha aplicado al ensamble y a la planificación de procesos.

A continuación se definen los componentes de búsqueda utilizados, los cuales están basados en el diseño clásico de los GAs y apoyados en las propuestas encontradas en la literatura citada para resolver el problema ASP.

Codificación de la secuencia de ensamble

En los GAs la codificación o representación de la solución juega un papel fundamental en su eficiencia y efectividad. La codificación debe ser apropiada para el problema de optimización abordado. Las soluciones se codifican en lo que se denomina cromosomas, y un cromosoma está compuesto por genes. Para ASP, un cromosoma debe poder representar una secuencia de ensamble de una manera sencilla y cada gen es un número natural que se corresponde con el identificador de cada pieza de la secuencia. Esta codificación es una representación clásica de tipo permutación, donde cada elemento (pieza) solo aparece una vez. La figura 2.8 muestra la estructura de un cromosoma para una secuencia compuesta por 7 piezas. La posición de un gen en el cromosoma representa la precedencia del ensamble de una pieza. Se puede notar que el ensamble de la pieza 2 precede al ensamble de la pieza 3, es decir, el componente 2 debe ser ensamblado antes que el componente 3. El primer componente en ser ensamblado es el 0 y el último es el 1. La población inicial de los cromosomas es generada de forma aleatoria.

0	2	3	6	5	4	1
---	---	---	---	---	---	---

Figura 2.8. Ejemplo de un cromosoma

Función de aptitud

La función de aptitud o de costo permite evaluar cada cromosoma (secuencia de ensamble) al considerar las restricciones del proceso de ensamble y calcular para cada solución un valor de aptitud que el GA utiliza para seleccionar los mejores cromosomas en cada generación. En esta investigación la función de aptitud va a ser la función de costo del ensamble definida en la fórmula 1.

Estrategia de selección

La estrategia de selección determina cuáles individuos son elegidos como padres para reproducirse y cuánta descendencia es producida por los individuos seleccionados. Las secuencias de ensamble que se van a “cruzar” se eligen al considerar su aptitud mediante la estrategia de selección por torneo, la cual permite obtener un número de secuencias padres a partir de un subconjunto de individuos elegidos al azar de la población.

Estrategia de reproducción

Al seleccionarse los padres del conjunto de individuos de la población, se lleva a cabo la reproducción, la cual tiene el papel de aplicar los operadores de variación como el cruzamiento y la mutación.

El cruzamiento es un operador binario y su papel es heredar algunas características de los dos padres para generar los descendientes. El cruce por emparejamiento parcial –en idioma inglés Partially Matched Crossover (PMX)– se utiliza por el GA propuesto. El PMX es un operador de reordenamiento, y cuando es aplicado a dos cromosomas padres, se seleccionan dos puntos de cruzamiento, entonces los valores entre estos dos puntos son intercambiados, y finalmente se eliminan los números repetidos de cada cromosoma descendiente.

De forma general los pasos de PMX (figura 2.9) son:

- 1) Seleccionar dos padres de la población;
- 2) Generar aleatoriamente las dos posiciones de cruzamiento y determinar el bloque que representa el rango entre los puntos para cada padre;
- 3) Intercambiar los dos bloques de los padres, generar dos cromosomas temporales;

- 4) Eliminar los genes duplicados, obtener los dos cromosomas descendientes.

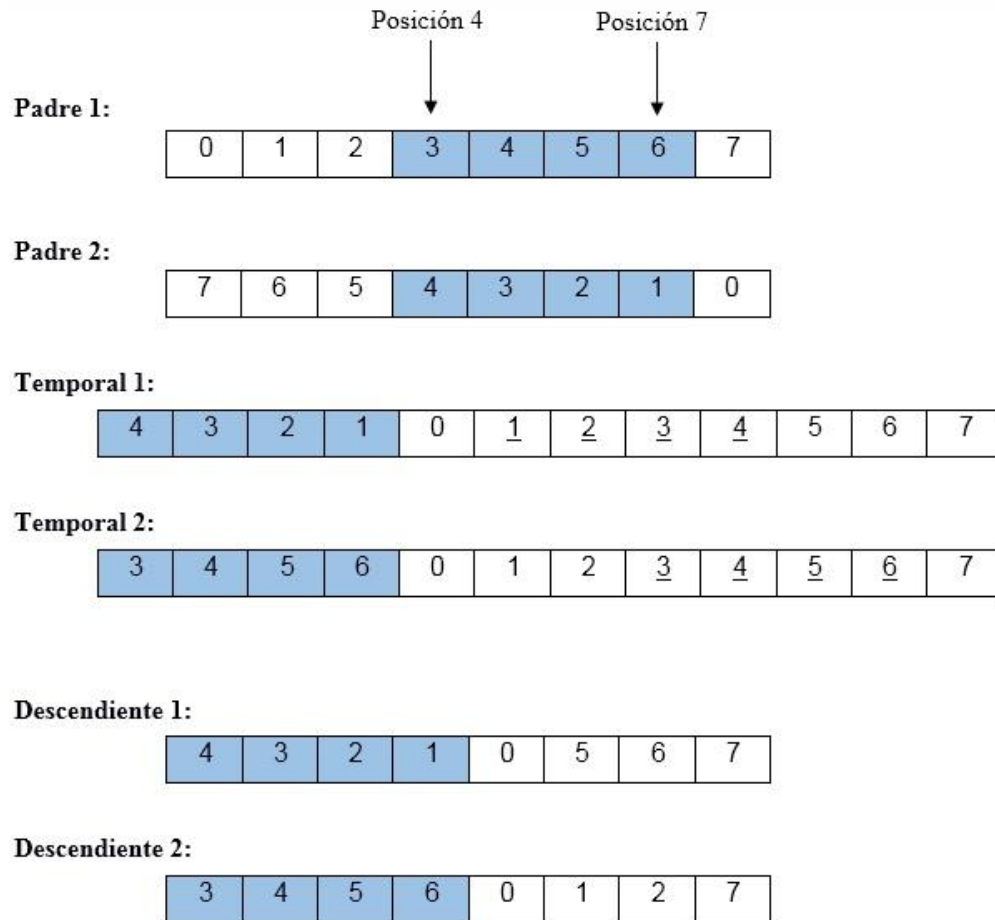


Figura 2.9. Cruce por emparejamiento parcial (PMX)

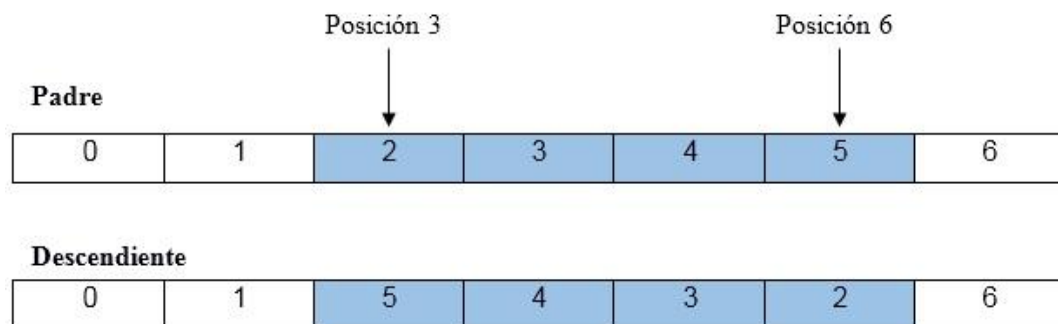


Figura 2.10. Mutación por inversión

La mutación es un operador unario que actúa sobre un único individuo y representa cambios pequeños del individuo seleccionado de la población. El GA usa la mutación por inversión, la cual consiste en la elección aleatoria de dos posiciones del cromosoma seleccionado e invierte los valores que están dentro de ese rango (figura 2.10).

Estrategia de reemplazo

Finalmente, la estrategia de reemplazo permite actualizar la población con los descendientes generados. El GA utiliza el reemplazo de estado estable –en inglés steady-state– que consiste en que cuando se producen los descendientes en cada generación del GA, estos reemplazan los peores individuos de la población de los padres.

Pasos fundamentales del GA propuesto

El GA está compuesto por los siguientes pasos:

- 1) Generar la población inicial de las secuencias de ensamble, obtener aleatoriamente el número decimal que representa a cada pieza;
- 2) Seleccionar dos padres de la población con el uso de la selección por torneo;
- 3) Utilizar el cruzamiento PMX y la mutación por inversión como operadores de variación para generar los nuevos cromosomas;
- 4) Actualizar la población al aplicar la estrategia de reemplazo de estado estable, y el criterio de parada es el número máximo de generaciones con un valor de 300.

El diagrama de flujo del anexo 3 muestra las etapas del GA que se propone.

2.2.3 Caso de estudio y discusiones

Caso de estudio

El GA propuesto (GA-ASP) para la optimización de la secuencia de ensamble fue programado en el lenguaje Python con la biblioteca inspyred, la cual está dedicada a la creación de algoritmos de inteligencia computacional bioinspirados. Una laptop con un procesador Intel Core i3-380M y 4GB de memoria RAM fueron las características básicas de hardware utilizadas.

Piezas	1	2	3	4	5	6	7	8	9	10	11
1	0	0	0	0	0	0	1	0	0	1	1
2	1	0	1	0	0	0	1	1	1	1	1
3	1	0	0	0	0	0	1	1	1	1	1
4	1	1	1	0	0	0	1	1	1	1	1
5	1	1	1	1	0	0	1	1	1	1	1
6	0	1	1	1	1	0	1	1	1	0	1
7	0	0	0	0	0	0	0	1	1	0	1
8	0	0	0	0	0	0	0	0	0	0	1
9	0	0	0	0	0	0	0	1	0	0	1
10	0	0	0	0	0	0	0	0	0	0	1
11	0	0	0	0	0	0	0	0	0	0	0

Tabla 2.1. Matriz de interferencia utilizada para inferir la dirección de ensamble +X

Se seleccionó como caso de estudio el ensamble de una transmisión automática mostrado en la figura 1.5, el cual está compuesto por 11 piezas. En el anexo 2 se ilustra el modelo CAD 3D del ensamble analizado a partir del cual se generan las matrices de interferencia con el uso del método descrito en la sección 2.1.

Piezas	1	2	3	4	5	6	7	8	9	10	11
1	0	1	1	1	1	0	0	0	0	0	0
2	0	0	0	1	1	1	0	0	0	0	0
3	0	1	0	1	1	1	0	0	0	0	0
4	0	0	0	0	1	1	0	0	0	0	0
5	0	0	0	0	0	1	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0
7	1	1	1	1	1	1	0	0	0	0	0
8	0	1	1	1	1	1	1	0	1	0	0
9	0	1	1	1	1	1	1	0	0	0	0
10	1	1	1	1	1	0	0	0	0	0	0
11	1	1	1	1	1	1	1	1	1	1	0

Tabla 2.2. Matriz de interferencia utilizada para inferir la dirección de ensamble -X

Las direcciones de ensamble consideradas son +X, -X, +Y, y -Y, las cuales están representadas en cuatro matrices de interferencia, las que se muestran en las tablas 2.1, 2.2, 2.3 y 2.4. La matriz de precedencia local del ensamble (tabla 2.5), la matriz de estabilidad (tabla 2.6), y la matriz de herramientas (tabla 2.7), están basadas en la estructura y el proceso de ensamble del caso de estudio.

Piezas	1	2	3	4	5	6	7	8	9	10	11
1	0	1	1	0	0	0	1	1	1	1	1
2	1	0	1	1	1	0	0	0	0	0	0
3	1	1	0	1	1	0	1	1	1	0	0
4	0	1	1	0	1	0	0	0	0	0	0
5	0	1	1	1	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0
7	1	0	1	0	0	0	0	1	1	0	0
8	1	0	1	0	0	0	1	0	1	1	1
9	1	0	1	0	0	0	1	1	0	1	1
10	1	0	0	0	0	0	0	1	1	0	0
11	1	0	0	0	0	0	0	1	1	0	0

Tabla 2.3. Matriz de interferencia utilizada para inferir la dirección de ensamble +Y.

Piezas	1	2	3	4	5	6	7	8	9	10	11
1	0	1	1	0	0	0	1	1	1	1	1
2	1	0	1	1	1	0	0	0	0	0	0
3	1	1	0	1	1	0	1	1	1	0	0
4	0	1	1	0	1	0	0	0	0	0	0
5	0	1	1	1	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0
7	1	0	1	0	0	0	0	1	1	0	0
8	1	0	1	0	0	0	1	0	1	1	1
9	1	0	1	0	0	0	1	1	0	1	1
10	1	0	0	0	0	0	0	1	1	0	0
11	1	0	0	0	0	0	0	1	1	0	0

Tabla 2.4. Matriz de interferencia utilizada para inferir la dirección de ensamble -Y.

Los parámetros de configuración de la función de costo son $W_g = 50$, $W_d = 0.4$, $W_p = 0.15$, $W_s = 0.15$, $W_t = 0.3$, y los del algoritmo GA se establecen como sigue: población inicial = 100, generaciones = 300, probabilidad de cruzamiento = 0.9, y probabilidad de mutación = 0.1. La pieza base es el componente 1 y es seleccionada según el criterio de que es la pieza que tiene el mayor número de conexiones con otras piezas en el ensamble. El conjunto de las secuencias de ensamble iniciales (población inicial en el GA) fue generado de forma aleatoria.

Piezas	1	2	3	4	5	6	7	8	9	10	11
1	0	1	1	1	1	1	1	1	1	1	1
2	-1	0	-1	1	1	1	-1	-1	-1	1	1
3	-1	1	0	1	1	1	-1	-1	-1	1	1
4	-1	-1	-1	0	1	1	-1	-1	-1	-1	1
5	-1	-1	-1	-1	0	1	-1	-1	-1	-1	-1
6	-1	-1	-1	-1	-1	0	-1	-1	-1	-1	-1
7	-1	1	1	1	1	1	0	1	1	1	1
8	-1	1	1	1	1	1	-1	0	-1	1	1
9	-1	1	1	1	1	1	-1	1	0	1	1
10	-1	-1	-1	1	1	1	-1	-1	-1	0	1
11	-1	-1	-1	-1	1	1	-1	-1	-1	-1	0

Tabla 2.5. Matriz de precedencia local del ensamble

Piezas	1	2	3	4	5	6	7	8	9	10	11
1	0	0	2	2	0	0	2	0	0	0	2
2	2	0	2	0	2	0	2	2	2	0	0
3	2	2	0	0	0	0	0	0	0	0	0
4	2	0	2	0	0	0	0	0	0	0	0
5	0	2	0	0	0	2	0	0	0	0	0
6	0	0	0	0	2	0	0	0	0	0	0
7	2	0	0	0	0	0	0	0	2	0	0
8	0	0	0	0	0	0	0	0	2	0	2
9	0	0	0	0	0	0	2	2	0	2	2
10	2	0	0	0	0	0	0	0	2	0	0
11	2	0	0	0	0	0	0	2	2	0	0

Tabla 2.6. Matriz de estabilidad del ensamble

Piezas	T1	T2
1	1	0
2	1	0
3	1	1
4	0	1
5	0	1
6	0	1
7	1	0
8	1	0
9	1	0
10	1	0
11	1	1

Tabla 2.7. Matriz de herramientas del ensamble

Se realizaron varios experimentos con el algoritmo GA desarrollado y se seleccionaron tres de ellos que representan las mejores secuencias de ensamble obtenidas, los cuales se muestran en la tabla 2.8. La primera fila para cada experimento muestra la secuencia de ensamble optimizada, las restricciones del proceso de ensamble y el costo de ensamble (valor de aptitud). Las dos filas siguientes son las direcciones de ensamble y las herramientas utilizadas para ensamblar las piezas correspondientes.

Experimento	Secuencia de ensamble	M	Np	Ns	Nd	Nt	F(s)
1	Secuencia Direcciones Herramientas 1 → 7 → 9 → 8 → 10 → 11 → 3 → 2 → 4 → 5 → 6 -X -X -X -X -X -X +X +X +X +X +X T1 T1 T1 T1 T1 T1 T1 T1 T2 T2 T2	0	1	1	1	1	1.00
2	Secuencia Direcciones Herramientas 1 → 7 → 9 → 10 → 8 → 11 → 3 → 2 → 4 → 5 → 6 -X -X -X -X -X -X +X +X +X +X +X T1 T1 T1 T1 T1 T1 T1 T1 T2 T2 T2	0	2	1	1	1	1.15
3	Secuencia Direcciones Herramientas 1 → 3 → 2 → 4 → 5 → 6 → 7 → 9 → 8 → 10 → 11 +X +X +X +X +X +X -X -X -X -X -X T1 T1 T1 T2 T2 T2 T1 T1 T1 T1 T1	0	1	1	1	2	1.30

Tabla 2.8. Secuencias de ensamble optimizadas con el algoritmo propuesto GA-ASP.

Discusiones

La tabla 2.9 muestra la comparación entre los métodos de optimización GA-ASP, MA-ASP y ACO-ASP. Los dos últimos métodos fueron mencionados en la sección 1.3 como técnicas avanzadas para la optimización de la secuencia de ensamble. El caso de estudio examinado por estos métodos es el mismo que el ensamble de la figura 1.5.

Experimento		Mejor secuencia de ensamble	M	Np	Ns	Nd	Nt	F(s)
GA-ASP	Secuencia	1 → 7 → 9 → 8 → 10 → 11 → 3 → 2 → 4 → 5 → 6	0	1	1	1	1	1.00
	Direcciones	-X -X -X -X -X -X +X +X +X +X +X						
	Herramientas	T1 T1 T1 T1 T1 T1 T1 T1 T2 T2 T2						
MA-ASP	Secuencia	1 → 3 → 2 → 4 → 5 → 6 → 7 → 10 → 9 → 8 → 11	0	2	1	1	2	1.45
	Direcciones	+X +X +X +X +X +X -X -X -X -X -X						
	Herramientas	T1 T1 T1 T2 T2 T2 T1 T1 T1 T1 T1						
ACO-ASP	Secuencia	11 → 8 → 9 → 7 → 10 → 1 → 3 → 2 → 4 → 5 → 6	0	4	1	0	1	1.05
	Direcciones	+X +X +X +X +X +X +X +X +X +X +X						
	Herramientas	T1 T1 T1 T1 T1 T1 T1 T1 T2 T2 T2						

Tabla 2.9. Mejores secuencias de ensamble obtenidas con los algoritmos GA-ASP, MA-ASP y ACO-ASP respectivamente.

La mejor secuencia de ensamble de cada uno de estos métodos fue evaluada por la función de costo del ensamble propuesta, por lo que los objetivos de la optimización son los mismos, y por lo tanto la comparación entre los métodos es válida. Sin embargo, se debe considerar que en la literatura citada MA-ASP y ACO-ASP tienen como objetivo de optimización la reducción de los cambios de dirección de ensamble, lo que limita su campo de acción al no considerar otras restricciones del proceso de ensamble. Si solo se toma en cuenta la reducción de los cambios de dirección de ensamble, la solución obtenida por GA-ASP y MA-ASP son similares, pues $N_d = 1$, y es superada por ACO-ASP con $N_d = 0$. No obstante, ACO-ASP tiene como desventaja que requiere de operaciones de desensamble (ej. (6, +Y), (11, +X), (6, -X), y (6, -Y)) utilizadas en la inicialización del algoritmo, y además genera la secuencia de desensamble y la invierte

para obtener la secuencia de ensamble, ya que considera que las secuencias de desensamble son las inversas de las secuencias de ensamble del producto, lo cual puede no ocurrir en el ensamble de un artículo mecánico.

El método GA-ASP requiere de una mayor cantidad de iteraciones (300 generaciones) en comparación con las 100 que necesita MA-ASP, para converger a las soluciones mostradas, lo que resulta en un aumento del costo computacional. Sin embargo, cuando se consideran otras restricciones del proceso de ensamble como la precedencia local del ensamble, las piezas inestables y los cambios de las herramientas, GA-ASP obtiene la mejor secuencia de ensamble en comparación con las que resultan de MA-ASP y ACO-ASP, ya que estos últimos están especializados solamente en la reducción de los cambios de dirección del ensamble.

2.3 Conclusiones del capítulo

El presente capítulo describió un método asistido por computadora denominado GA-ASP basado en un algoritmo genético para la optimización y planificación de las secuencias de ensamble de un producto mecánico. El GA-ASP usó un modelo de optimización que incluyó varias restricciones del proceso de ensamble. Estas restricciones fueron representadas en estructuras de datos de tipo matriz, de las cuales las matrices de interferencia y la matriz de precedencia local del ensamble fueron generadas automáticamente. Fue descrito e implementado un método computacional para la generación automática de las matrices de interferencia a partir del ensamble CAD del producto.

El GA propuesto fue satisfactoriamente implementado en un software en el lenguaje de programación Python y fue aplicado a un caso de estudio. Los resultados obtenidos por GA-ASP y la comparación de las soluciones de este con las de dos métodos existentes, demostraron que al considerar las restricciones del proceso de ensamble utilizadas por el modelo de optimización descrito, GA-ASP obtuvo mejores soluciones para el ejemplo de ensamble analizado.

CONCLUSIONES GENERALES

Se creó un método asistido por computadora que permitió automatizar la obtención de la información de interferencia entre las piezas del ensamble al tomar como fuente de datos un modelo geométrico tridimensional del producto.

Se aplicó un método de optimización basado en los algoritmos genéticos que utilizó un modelo que incluye las restricciones del proceso de ensamble, lo cual permitió lograr la planificación asistida por computadora de las secuencias de ensamble de un producto mecánico, por lo que queda demostrada la hipótesis de la investigación.

RECOMENDACIONES

1. El método propuesto para la obtención de la información de interferencia entre las piezas del ensamble es restrictivo en cuanto a las direcciones de desensamble que contempla, por lo que deben analizarse en próximos estudios productos que contengan piezas ubicadas en los ejes de coordenada diferentes a los ejes principales.
2. Aumentar el alcance del modelo de optimización al considerar:
 - Otras direcciones de ensamble
 - El efecto de la tolerancia del ensamble
 - Otros tipos de planes de ensamble
3. Incluir en el modelo de optimización otras restricciones del proceso de ensamble como:
 - Minimizar los cambios de los tipos de conexión
 - Minimizar los tiempos del ensamble
 - Minimizar los costos del ensamble
4. La automatización de la obtención de los datos de entrada para el método de optimización como la información de la estabilidad y de las herramientas de ensamble.
5. Desarrollo de un módulo basado en el método de optimización propuesto que esté vinculado a un sistema CAD.

REFERENCIAS BIBLIOGRÁFICAS

- [1] C. Shiang-Fong, L. Yong-Jin, "The Application of Multi-Level Genetic Algorithms in Assembly Planning," *Journal Of Industrial Technology*, vol. 17, no. 4, pp. 1–9, 2001.
- [2] J. Naisbitt, "Megatrends - Ten New Directions Transforming Our Lives." 1982.
- [3] H. B. Shan, S. X. Li, D. G. Gong, P. Lou, "Genetic simulated annealing algorithm-based assembly sequence planning," in *International Technology and Innovation Conference 2006 (ITIC 2006)*, pp. 1573–1579, 2006.
- [4] L. S. Homen de Mello, A. C. Sanderson, "Automatic Generation of Mechanical Assembly Sequences," Carnegie Mellon University, the Robotics Institute, 1988.
- [5] R. M. Marian, "Optimisation of assembly sequences using genetic algorithms," PhD thesis, University of South Australia, Adelaide, Australia. 2003.
- [6] D. E. Whitney, *Mechanical Assemblies: Their Design, Manufacture, and Role in Product Development*. Oxford University Press, 2004, p. 573. ISBN 0-19-515782-6.
- [7] J. D. Wolter, "A Combinatorial Analysis of Enumerative Data Structures for Assembly Planning," in *International Conference on Robotics & Automation*, 1991, pp. 611–618.
- [8] I. Zeid, *Mastering CAD/CAM*. McGraw-Hill, 2004.
- [9] Y. Wang, J. Liu, "Subassembly identification for assembly sequence planning," *International Journal of Advanced Manufacturing Technology*, pp. 781–793, 2013.
- [10] A. Bourjault, "Contribution à une approche méthodologique de l'assemblage automatisé: élaboration automatique des séquences opératoires, (Contribution to a systematic approach of automatic assembly: automatic determination of operation sequences)," PhD thesis, Besançon, France: Université de Franche-Comté, 1984. (en Francés).
- [11] A. J. D. Lambert, S. M. Gupta, *Disassembly Modeling for Assembly, Maintenance, Reuse, and Recycling*. CRC press, Boca Raton, Florida, 2005, p. 446. ISBN 1-57444-334-8.
- [12] A. Delchambre, "Computer-Aided Assembly Planning," London: Chapman and Hall, 1992.
- [13] H. Ko, K. Lee, "Automatic assembly procedure generation from mating conditions," *Computer-Aided Design*, vol. 19, no. 1, pp. 3–10, 1987.

- [14] P. Jimenez, "Survey on assembly sequencing: a combinatorial and geometrical perspective," *Journal of Intelligent Manufacturing*, vol. 24, no. 2, pp. 235-250. 2013.
- [15] L. S. Homen de Mello, A. C. Sanderson, "Representation of mechanical assembly sequences," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 2, pp. 211–227, 1991.
- [16] T. L. De Fazio, D. E. Whitney, "Simplified generation of all mechanical assembly sequences," *IEEE Journal of Robotics and Automation*, vol. 3, no. 6, pp. 640–658, 1987.
- [17] T. Prenting, R. Battaglin, "The precedence diagram: A tool for analysis in assembly line balancing," *Journal of Industrial Engineering*, vol. 15, no. 4, pp. 208–13, 1964.
- [18] K. S. Naphade, R. H. Storer, S. D. Wu, "Graph-Theoretic Generation of Assembly Plans, Part I: Correct Generation of Precedence Graphs." Lehigh University, Bethlehem, Pennsylvania, 1999.
- [19] L. S. Homen de Mello, A. C. Sanderson, "AND/OR graph representation of assembly plan," *IEEE Transactions on Robotics and Automation*, vol. 6, no. 2, pp. 188–99, 1990.
- [20] G. Dini, M. Santochi, "Automated sequencing and subassembly detection in assembly planning," *Annals of the CIRP*, vol. 41, no. 3, pp. 1–4, 1992.
- [21] Y. M. Huang, C. Huang, "Disassembly matrix for disassembly processes of products," *International Journal of Production Research*, vol. 40, no. 2, 2002.
- [22] C. Sinanoglu, H. R. Borklu, "An approach to determine geometric feasibility to assembly states by intersection matrices in assembly sequence planning," *Journal of Intelligent Manufacturing*, vol. 15, no. 4, pp. 543–559, 2004.
- [23] D. F. Baldwin, "Algorithmic Methods and Software Tools For The Generation Of Mechanical Assembly Sequences," MsC Thesis, Massachusetts Institute of Technology, 1990.
- [24] K. Sambhoos, B. Koc, R. Nagi, "Extracting Assembly Mating Graphs for Assembly Variant Design," *Journal of Computing and Information Science in Engineering*, vol. 9, no. 3, pp. 1–9, 2009.
- [25] INTERNATIONAL ORGANIZATION FOR STANDARDIZATION, "Industrial automation systems and integration -- Product data representation and exchange. Part 1: Overview and fundamental principles." pp. 1–28. ISO 10303-1: 1994.
- [26] INTERNATIONAL ORGANIZATION FOR STANDARDIZATION, "Industrial automation systems and integration -- Product data representation and exchange.

- Part 41: Integrated generic resources: Fundamentals of product description and support." pp. 1–201. ISO 10303-41: 1994.
- [27] INTERNATIONAL ORGANIZATION FOR STANDARDIZATION, "Industrial automation systems and integration -- Product data representation and exchange. Part 42: Integrated generic resources: Geometric and topological representation." pp. 1–245. ISO 10303-42: 1994.
 - [28] INTERNATIONAL ORGANIZATION FOR STANDARDIZATION, "Industrial automation systems and integration -- Product data representation and exchange. Part 43: Integrated generic resources: Representations structures." pp. 1–38. ISO 10303-43: 1994.
 - [29] INTERNATIONAL ORGANIZATION FOR STANDARDIZATION, "Industrial automation systems and integration -- Product data representation and exchange. Part 21: Implementation methods: Clear text encoding of the exchange structure." pp. 1–69. ISO 10303-21: 1994.
 - [30] INTERNATIONAL ORGANIZATION FOR STANDARDIZATION, "Industrial automation systems and integration -- Product data representation and exchange. Part 203: Application protocol: Configuration controlled design." pp. 1–529. ISO 10303-203: 1994.
 - [31] C. Pan, S. S.-F. Smith, G. C. Smith, "Determining Interference Between Parts in CAD STEP Files for Automatic Assembly Planning," *Journal of Computing and Information Science in Engineering*, vol. 5, no. 1, pp. 56–62, 2005.
 - [32] P. Jimenez, F. Thomas, C. Torras, "3D Collision Detection: A Survey," *Computers & Graphics*, vol. 25, no. 2, pp. 269–285, 2001.
 - [33] J. Yu, C. Wang, "Method for discriminating geometric feasibility in assembly planning based on extended and turning interference matrix," *International Journal of Advance Manufacturing Technology*, vol. 67, pp. 1867–1882, 2013.
 - [34] H. Takeyama, H. Sekiguchi, T. Kojima, K. Inoue, T. Honda, "Study on Automatic Determination of Assembly Sequence," *Annals of the CIRP*, vol. 32, no. 1, pp. 371–374, 1983.
 - [35] B. Frommherz, J. Hornberger, "Automatic Generation of Precedence Graphs," Publications of the University of Karlsruhe, Federal Republic of Germany, 1988.
 - [36] J. D. Wolter, "On the Automatic Generation of Assembly Plans," Center for Research on Integrated Manufacturing, Robot Systems Division, University of Michigan, September, 1988.

- [37] J. M. Miller, R. L. Holffman, "Automatic Assembly Planning with Fasteners," in *the proceedings of the IEEE Conference on Robotics and Automation*, 1989, pp. 69–74.
- [38] D. Ben-Arieh, B. Kramer, "Computer-aided process planning for assembly: generation of assembly operations sequence," *International Journal of Production Research*, vol. 32, no. 3, pp. 643–656, 1994.
- [39] J. Latombe, R. H. Wilson, F. Cazals, "Assembly Sequencing with Toleranced Parts," *Journal of Computer Aided Design*, vol. 29, no. 2, pp. 159–174, 1997.
- [40] Z. Kunica, B. Vranjes, "Towards automatic generation of plans for automated assembly," *International Journal of Production Research*, vol. 37, no. 8, pp. 1817–1836, 1999.
- [41] N. Ye, D. A. Urzi, "Heuristic rules and strategies of assembly planning: experiment and implications in the design of assembly decision support system," *International Journal of Production Research*, vol. 34, no. 8, pp. 2211–2228, 1996.
- [42] X. F. Zha, S. Y. Lim, S. C. Fok, "Integrated intelligent design and assembly planning: a survey," *International Journal of Advanced Manufacturing Technology*, vol. 14, no. 10, pp. 1–15, 1998.
- [43] C. K. Choi, X. F. Zha, T. L. Ng, W. S. Lau, "On the automatic generation of product assembly sequences," *International Journal of Production Research*, vol. 36, no. 3, pp. 617–633, 1998.
- [44] N. Boneschanscher, "Plan generation for flexible assembly systems," PhD thesis, Delft University of Technology, The Netherlands. 1993.
- [45] S. Lee, "Subassembly Identification and Evaluation for Assembly Planning," *IEEE Transactions on System, Man and Cybernetics*, vol. 24, no. 3, pp. 493–503, 1994.
- [46] Y. Wang, J. H. Liu, "Chaotic particle swarm optimization for assembly sequence planning," *Robotics and Computer Integrated Manufacturing*, vol. 26, no. 2, pp. 212–222, 2010.
- [47] L. S. Homen de Mello, "Task Sequence Planning for Robotic Assembly," PhD thesis, Carnegie-Mellon University, Pittsburgh, 1989.
- [48] L. Laperriere, H. A. Elmaraghy, "GAPP : A Generative Assembly Process Planner," *Journal of Manufacturing Systems*, vol. 15, no. 4, pp. 282–293, 1996.
- [49] S. Motavalli, A-U. Islam, "Multi-criteria assembly sequencing," *Computers & Industrial Engineering*, vol. 32, no. 4, pp. 743–51, 1997.

- [50] Q. Su, S. Lai, J. Liu, "Geometric computation based assembly sequencing and evaluating in terms of assembly angle, direction, reorientation, and stability," *Computer-Aided Design*, vol. 41, pp. 479–489, 2009.
- [51] F.F.R. Mohd, H. Windo, T. Ashutosh, "A review on assembly sequence planning and assembly line balancing optimisation using soft computing approaches," *International Journal of Advanced Manufacturing Technology*, pp. 335–349, 2012.
- [52] E. Talbi, *Metaheuristics: from design to implementation*, John Wiley & Sons, Hoboken, New Jersey, 2009, p. 618. ISBN 978-0-470-27858-1.
- [53] F. Bonneville, J. Henrioud, A. Bourjault, "Generation of assembly sequences with ternary operations," in *IEEE International Symposium on Assembly and Task Planning*, 1995, pp. 245–249.
- [54] M. Sebaaly, H. Fujimoto, F. Mrad, "Linear and non-linear assembly planning: fuzzy graph representation and GA search," in *IEEE International Conference on Robotics and Automation*, 1996, pp. 1533–1538.
- [55] R. Marian, L. Luong, K. Abhary, "A genetic algorithm for the optimisation of assembly sequences," *Computers & Industrial Engineering*, pp. 503–527, 2006.
- [56] Y-K. Choi, D.M. Lee, Y.B. Cho, "An approach to multi-criteria assembly sequence planning using genetic algorithms," *International Journal of Advance Manufacturing Technology*, vol. 42, pp. 180–188, 2009.
- [57] H. Tseng, C. Tang, "A sequential consideration for assembly sequence planning and assembly line balancing using the connector concept," *International Journal of Production Research*, vol. 44, no. 1, pp. 97–116, 2006.
- [58] Y-J. Tseng, J-Y. Chen, F-Y. Huang, "A multi-plant assembly sequence planning model with integrated assembly sequence planning and plant assignment using GA," *International Journal of Advance Manufacturing Technology*, vol. 48, pp. 333–345, 2010.
- [59] Y. Tseng, H. Kao, F. Huang, "Integrated assembly and disassembly sequence planning using a GA approach," *International Journal of Production Research*, vol. 48, no. 20, pp. 5991–6013, 2010.
- [60] S. Jin, W. Cai, X. Lai, Z. Lin, "Design automation and optimization of assembly sequences for complex mechanical systems," *International Journal of Advance Manufacturing Technology*, vol. 48, pp. 1045–1059, 2010.
- [61] P. DeLit, P. Latinne, B. Rekiek, A. Delchambre, "Assembly planning with an ordering genetic algorithm," *International Journal of Production Research*, vol. 39, no. 16, pp. 3623–3640, 2001.

- [62] J. Li, L. Khoo, S. Tor, "A Tabu-enhanced genetic algorithm approach for assembly process planning," *Journal of Intelligent Manufacturing*, vol. 14, no. 2, pp. 197–208, 2003.
- [63] H. Shan, S. Li, D. Gong, P. Lou, "Genetic simulated annealing algorithm-based assembly sequence planning," *IET conference publications*, vol. 524, pp. 1573–1579, 2006.
- [64] C. Hui, L. Yuan, and Z. Kai-Fu, "Efficient method of assembly sequence planning based on GAAA and optimizing by assembly path feedback for complex product," *International Journal of Advance Manufacturing Technology*, vol. 42, pp. 1187–1204, 2009.
- [65] L. Gao, W. Qian, X. Li, J. Wang, "Application of memetic algorithm in assembly sequence planning," *International Journal of Advanced Manufacturing Technology*, vol. 49, pp. 1175–1184, 2009.
- [66] F. Failli, G. Dini, "Ant colony systems in assembly planning: a new approach to sequence detection and optimization," in *Proceedings of the 2nd CIRP Intelligent Computation in Manufacturing Engineering*, 2000, pp. 227–232.
- [67] J. Wang, J. Liu, X. Zhong, "A novel ant colony algorithm for assembly sequence planning," *International Journal of Advanced Manufacturing Technology*, vol. 25, pp. 1137–1143, 2005.
- [68] J. Yu, C. Wang, "A max – min ant colony system for assembly sequence planning," *International Journal of Advanced Manufacturing Technology*, vol. 67, pp. 2819–2835, 2013.
- [69] H. Yu, J. Yu, W. Zhang, "An particle swarm optimization approach for assembly sequence planning," *Appl Mech Mater*, pp. 16–19, 2009.
- [70] Y. Xing, Y. Wang, "Assembly sequence planning based on a hybrid particle swarm optimisation and genetic algorithm," vol. 50, no. 24, pp. 7303–7312, 2012.

ANEXOS

Anexo 1. Diferencias entre dos métodos para la determinación de las interferencias entre las piezas del ensamble

Método descrito por Pan, Smith, Smith [31]	Método propuesto
Extracción de la información geométrica de las piezas contenidas en un fichero STEP.	Extracción de la información geométrica de las piezas contenidas en un fichero STEP.
	Generación de la aproximación triangular de las superficies de las piezas
Obtención de las interferencias entre las piezas con superficies prismáticas y cilíndricas.	Obtención de las interferencias entre las piezas con superficies prismáticas y cilíndricas, y de forma libre.
Se detectan las interferencias entre cada par de piezas al utilizar un algoritmo de proyecciones de dos piezas en un plano perpendicular a una dirección principal de ensamble.	Se detectan las interferencias entre cada par de piezas al utilizar un algoritmo de Detección de interferencia múltiple.
Creación de las matrices de interferencia.	Creación de las matrices de interferencia.

Tabla A1.1. Diferencias entre el método propuesto para la determinación de las interferencias y el método descrito por Pan, Smith, Smith

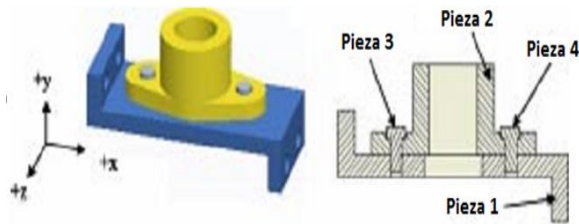


Figura A1.1. Ensamble con piezas prismáticas y cilíndricas

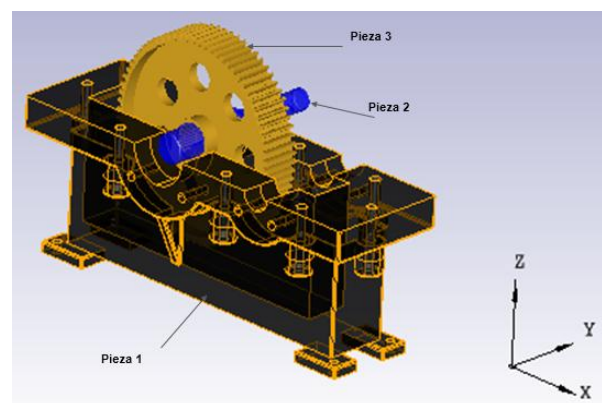


Figura A1.2. Ensamble con piezas de forma libre

Anexo 2. Modelo CAD 3D del ensamble de una transmisión automática (AFI).

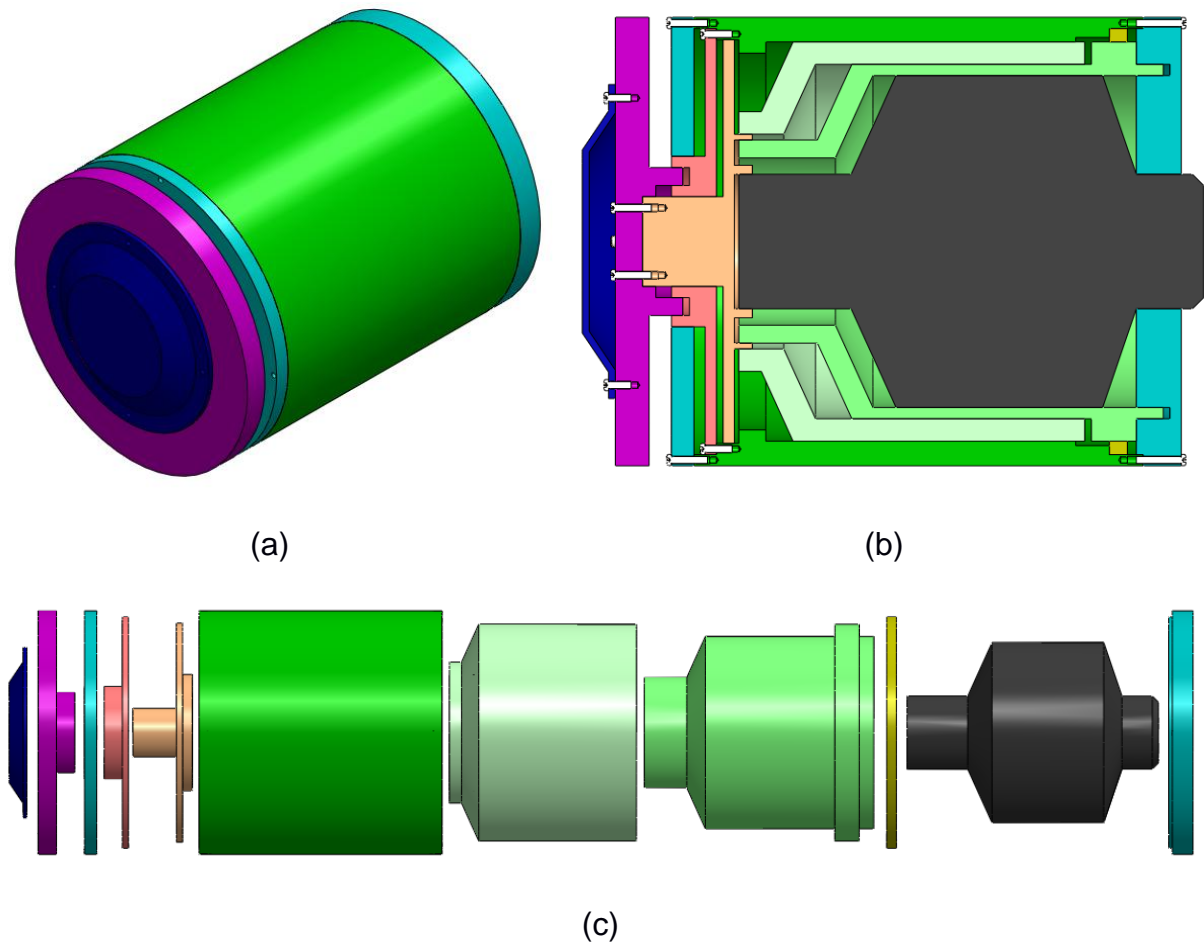


Figura A2.1. Modelo CAD 3D del ensamble de una transmisión automática (AFI). (a) vista isométrica, (b) vista de sección del ensamble con los elementos de fijación, (c) vista explosionada

Anexo 3. Diagrama de flujo de la optimización de la secuencia de ensamble mediante GA.

Figura A3.1. Diagrama de flujo de la optimización de la secuencia de ensamble mediante GA