



# UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MÉXICO

---

---



FACULTAD DE INGENIERÍA

DOCTORADO EN CIENCIAS DE LA INGENIERÍA

## T E S I S

# PROPUESTA DE METODOLOGÍA PARA SÍNTESIS ÓPTIMA DE MECANISMOS

TESIS PRESENTADA POR: MARIANA MORALES BENHUMEA  
PARA OBTENER EL GRADO DE DOCTORA EN CIENCIAS DE LA INGENIERÍA

Tutor Académico: Dr. Juan Carlos Ávila Vilchis  
Tutor Adjunto: Dr. Julio César Díaz Montes

2020

---

# Resumen

La síntesis dimensional de mecanismos planos de generación de trayectoria es ampliamente utilizada en diversos campos de la ingeniería y puede obtenerse de forma exacta u óptima.

En la síntesis dimensional exacta, aún cuando un punto del eslabón acoplador de un mecanismo de cuatro barras pasa por todos los puntos planteados, es posible que la trayectoria generada a partir de un número reducido de puntos, no corresponda a la que se desea que recorra el punto del acoplador, sin embargo, si se incrementa el número de puntos al plantear el problema, el sistema de ecuaciones para resolverlo, se vuelve inconsistente.

En estos casos, la síntesis óptima de mecanismos es capaz de obtener un eslabonamiento cuya trayectoria descrita por un punto del eslabón acoplador se aproxime la trayectoria ideal, con la ventaja de que a mayor número de puntos usados, la curva generada es más suave y describe mejor el recorrido completo del punto del acoplador.

En este trabajo se plantea la obtención de síntesis dimensional de mecanismos a través de una metodología consistente en el uso de algoritmos heurísticos capaz de adaptarse a problemas de diversas aplicaciones y número de puntos.

**Palabras clave:** *síntesis de mecanismos planos, optimización, algoritmos heurísticos, generador de trayectoria, mecanismos de cuatro barras.*

---

# Abstract

Dimensional synthesis of a four bar linkage for path generation is widely used in a variety of engineering fields and can be achieved in an exact or optimal form.

In exact dimensional synthesis, the coupler point pass through all the desired points, but the coupler path may not be a smooth one or more points may be required to describe properly a smooth path, resulting in linear equations systems that can not be solved.

This is when optimal dimensional synthesis is able to achieve a smooth coupler path, due to the increased number of points along the curve that result on a path very close to the desired one.

In this work, the dimensional synthesis of a planar mechanism is achieved by using a methodology that combines heuristic algorithms that aim to minimize the distance between the ideal coupler path and the actual one. This methodology is able to adapt itself to different application problems and variations in the number of points.

**Keywords:** *planar mechanism synthesis, optimization, heuristic algorithm, path generator synthesis, four bar planar mechanism.*

---

# Índice general

Resumen

Abstract

Índice de figuras

Índice de tablas

<b>1. Introducción</b>	<b>10</b>
1.1. Planteamiento del problema . . . . .	11
1.2. Justificación . . . . .	12
1.3. Hipótesis . . . . .	13
1.4. Objetivo . . . . .	13
1.4.1. Objetivos particulares . . . . .	13
1.5. Alcances y Limitaciones . . . . .	14
1.6. Metodología propuesta . . . . .	14
<b>2. Antecedentes</b>	<b>16</b>
2.1. Marco teórico: conceptos . . . . .	16
2.2. Estado del Arte . . . . .	21
2.2.1. Algoritmo de Recocido Simulado (SA) . . . . .	23
2.2.2. Algoritmos Evolutivos (EA) . . . . .	28
2.2.3. Lógica difusa (FL) . . . . .	36
2.2.4. Búsqueda armónica (HS) . . . . .	37
2.2.5. Mapeo cinemático (KM) . . . . .	38

2.2.6.	Optimización por enjambre de partículas (PSO) . . . . .	39
2.2.7.	Algoritmo del banco de krill modificado (MKH) . . . . .	43
2.2.8.	Optimización por algoritmo de luciérnagas (FA) . . . . .	44
2.2.9.	Optimización por algoritmo de colonia artificial de abejas (ABC) . . . . .	48
2.2.10.	Optimización por algoritmo de búsqueda del cuco (CS) . . . . .	49
<b>3.</b>	<b>Síntesis óptima de mecanismos de cuatro barras (SOM4B)</b>	<b>53</b>
3.1.	Función objetivo de un mecanismo de cuatro barras . . . . .	53
3.2.	Minimización de la función objetivo del mecanismo de cuatro barras . . . . .	56
3.2.1.	Uso de algoritmo SA . . . . .	56
3.2.2.	Uso de algoritmos genéticos . . . . .	58
3.2.3.	Uso de algoritmo PSO . . . . .	60
3.2.4.	Uso de algoritmo FA . . . . .	62
3.2.5.	Uso de algoritmo ABC . . . . .	65
3.2.6.	Uso de algoritmo CS . . . . .	66
<b>4.</b>	<b>Propuesta de metodología para SOM4B</b>	<b>69</b>
4.1.	Conjunto de soluciones iniciales . . . . .	69
4.2.	Algoritmos de rápida convergencia . . . . .	70
4.3.	Algoritmos de análisis detallado . . . . .	71
4.4.	Propuesta de combinación de algoritmos . . . . .	71
4.5.	Propuesta de metodología para síntesis óptima de mecanismos . . . . .	73
4.6.	Análisis de resultados . . . . .	75
4.6.1.	Combinación PSO-AG . . . . .	75
4.6.2.	Combinación PSO-FA . . . . .	75
4.6.3.	Combinación PSO-ABC . . . . .	76
4.6.4.	Combinación ABC-AG . . . . .	78
4.6.5.	Combinación ABC-FA . . . . .	78
4.6.6.	Combinación CS-AG . . . . .	79
4.6.7.	Combinación CS-FA . . . . .	80
4.6.8.	Comparación de las combinaciones evaluadas . . . . .	81
4.7.	Postprocesamiento de resultados . . . . .	82
4.7.1.	Análisis cinemático de mecanismos obtenidos . . . . .	82

4.7.2. Cumplimiento de la condición de Grashof . . . . .	82
<b>5. Conclusiones</b>	<b>84</b>
<b>Referencias</b>	<b>87</b>

---

# Índice de figuras

2.1. Tipos de eslabones por el número de nodos que presentan [10]. . . . .	18
2.2. Juntas cinemáticas de par inferior [10]. . . . .	18
2.3. Juntas cinemáticas de par superior [10]. . . . .	19
2.4. Diferentes métodos de síntesis de mecanismos. Modificada de [12]. . .	21
2.5. Algoritmo SA . . . . .	27
2.6. Tipo de cruce: puntual . . . . .	31
2.7. Trayectorias del acoplador de diferentes mecanismos sintetizados [20]	32
2.8. Comparación de efectividad entre AG y AG-FL [33]. . . . .	37
2.9. Tendencia de movimiento de una partícula en el algoritmo PSO [41]. .	41
2.10. Algoritmo PSO. . . . .	42
2.11. Algoritmo FA . . . . .	46
2.12. Algoritmo ABC. . . . .	50
2.13. Algoritmo CS. . . . .	52
3.1. Parámetros geométricos de un M4B [47]. . . . .	54
3.2. SM4B con el algoritmo de SA. Elaboración propia. . . . .	59
3.3. SM4B con algoritmos genéticos. Elaboración propia. . . . .	61
3.4. SM4B con algoritmo PSO. Elaboración propia. . . . .	62
3.5. SM4B con algoritmo FA. Elaboración propia. . . . .	64
3.6. SM4B con algoritmo FA, con espacio de búsqueda reducido. Elabora- ción propia. . . . .	64
3.7. SM4B con algoritmo ABC. Elaboración propia. . . . .	66
3.8. SM4B con algoritmo CS. Elaboración propia. . . . .	67
4.1. Etapas de la metodología propuesta. Elaboración propia. . . . .	74

4.2. SM4B con una combinación de algoritmos PSO y ABC. . . . .	77
--	----



---

# Índice de tablas

2.1. Relación entre número de puntos de precisión, ecuaciones e incógnitas para síntesis exacta de mecanismos [10]. . . . .	22
4.1. Combinaciones de algoritmos de optimización a evaluar para la metodología de síntesis de mecanismos. . . . .	72
4.2. Parámetros de los algoritmos en la combinación PSO-AG. . . . .	75
4.3. Parámetros de los algoritmos en la combinación PSO-FA. . . . .	76
4.4. Parámetros de los algoritmos en la combinación ABC-AG. . . . .	79
4.5. Parámetros de los algoritmos en la combinación ABC-FA. . . . .	79
4.6. Parámetros de los algoritmos en la combinación CS-AG. . . . .	80
4.7. Parámetros de los algoritmos en la combinación CS-FA. . . . .	81
4.8. Comparación de tiempo de ejecución y valor de error estructural obtenido . . . . .	82
4.9. Causas de detención de la ejecución del algoritmo utilizado para la SOM4B. . . . .	83

---

# Capítulo 1

## Introducción

En este proyecto de investigación se propone estudiar algunos de los algoritmos de optimización existentes actualmente y aplicarlos a la síntesis de mecanismos, para determinar las ventajas y desventajas de cada uno y aplicarlos al desarrollo de una metodología que sea capaz de obtener la síntesis óptima de un mecanismo plano.

A partir de las coordenadas de puntos por los que el mecanismo debe pasar y que forman parte de la trayectoria que se desea que alcance un punto del mecanismo, se busca determinar las coordenadas que permitan identificar los pasadores y punto acoplador de todos los elementos del mecanismo en una posición inicial, realizando posteriormente un análisis cinemático para verificar los resultados obtenidos.

En el capítulo 2 se presentan los conceptos teóricos relativos a la síntesis de mecanismos, así como el panorama general de la investigación acerca de síntesis óptima de mecanismos.

En el capítulo 3 se analizan algunos algoritmos de optimización, aplicados al problema de síntesis de mecanismos planos, para observar su desempeño bajo diferentes valores iniciales en la búsqueda de un mecanismo cuyo punto del acoplador sea capaz de alcanzar de manera cercana la trayectoria definida por un grupo de puntos dados.

En el capítulo 4 se realiza una propuesta de metodología para el problema de

síntesis de mecanismos planos, con base en los resultados observados en la implementación de los algoritmos presentados en el capítulo 3, así mismo, se presenta el análisis de los resultados obtenidos al realizar la síntesis al seguir esta metodología, en comparación con la aplicación única de los algoritmos de forma individual.

## 1.1. Planteamiento del problema

Entre los criterios de optimización más utilizados en la síntesis de trayectoria de mecanismos se encuentran el mínimo error estructural, la restricción de dimensiones del mecanismo, el cumplimiento de la condición de Grashof y el ángulo mínimo de transmisión.

En algunos casos las dimensiones del mecanismo se restringen ya que es necesario que el mecanismo tenga un determinado tamaño. El cumplimiento de la condición de Grashof se refiere a que al menos un eslabón del mecanismo sea capaz de girar de forma continua  $360^\circ$ , debido a que en muchos casos a dicho eslabón se conecta un motor. El ángulo mínimo de transmisión es aquel formado entre el eslabón acoplador y el eslabón de salida y debe estar dentro de cierto intervalo para evitar puntos muertos o puntos de cambio, es decir, posiciones en las que es posible que el mecanismo deje de girar normalmente, presentándose una singularidad.

Otras características que podrían utilizarse como criterios particulares de optimización son la capacidad de generar presión uniforme (para sujeción de objetos en mecanismos de prótesis de mano, por ejemplo), la estabilidad, el equilibrio y la transmisión de fuerza en mecanismos utilizados en elementos de rehabilitación [1][2].

Si bien se encuentra publicada bastante información sobre la síntesis óptima de trayectoria de mecanismos, hasta ahora no se han encontrado estudios en los que se analicen las diversas soluciones obtenidas de forma simultánea para verificar que cumplan con los criterios de optimización mencionados en el párrafo anterior, lo que representa un área poco explorada para los dispositivos de rehabilitación [3][4].

Por lo anterior, se buscará establecer una MSM (Metodología de Síntesis de Mecanismos) que cumpla con varios criterios de optimización de forma simultánea, tales como la generación de trayectoria, la estabilidad, la transmisión de fuerza, el equilibrio y la condición de Grashof, por ejemplo. Los mecanismos obtenidos podrán ser utilizados en diferentes áreas, entre ellas, los dispositivos de rehabilitación.

## 1.2. Justificación

La síntesis exacta de un mecanismo se refiere al proceso de establecer ecuaciones para que un punto del mecanismo alcance todos los puntos deseados, que se conocen como puntos de precisión. Sin embargo, cuando el número de puntos de precisión es elevado, se obtienen sistemas de ecuaciones inconsistentes, por lo que no es posible utilizar síntesis exacta y es por ello que se utiliza síntesis óptima ya que esta no está restringida al número de puntos a alcanzar, además de que entre mayor sea la cantidad de puntos de precisión se puede definir mejor la trayectoria que se desea que el mecanismo siga [5][6].

Dentro de la síntesis óptima, los métodos heurísticos existentes tienen ciertas limitantes, como la dependencia de la solución inicial generada de forma aleatoria, en el caso de los algoritmos genéticos, o bien la dificultad de programación de los métodos PSO o el Algoritmo de Competición Imperialista (ICA, del inglés *Imperialist Competitive Algorithm*) [7][8].

Se propone utilizar técnicas existentes de síntesis de mecanismos, con el fin de establecer una metodología que integre las ventajas y elimine las desventajas de diferentes métodos de optimización aplicados a la síntesis de generación de trayectoria de mecanismos planos para que los mecanismos generados satisfagan múltiples criterios, además del mínimo error estructural.

## 1.3. Hipótesis

Es posible elaborar una metodología que determine, a partir de una función objetivo apropiada (aquella que contenga los elementos que definen un mecanismo plano como posiciones por alcanzar, dimensiones, distancia entre apoyos y localización de los mismos) el mecanismo óptimo que satisface, además del criterio de mínimo error estructural, al menos un criterio secundario adicional.

## 1.4. Objetivo

Proponer y validar una metodología que permita realizar un proceso de síntesis óptima de mecanismos para diferentes aplicaciones.

### 1.4.1. Objetivos particulares

- Establecer los criterios que se considerarán para integrar la metodología de síntesis óptima de mecanismos.
- Evaluar la conveniencia de los métodos de optimización más representativos para determinar las ventajas y desventajas de su uso aplicado a síntesis de mecanismos.
- Realizar simulaciones numéricas de implementación de los métodos revisados para determinar su convergencia.
- Determinar la flexibilidad que los métodos de optimización revisados presentan con respecto a la adición de restricciones a la función objetivo.
- Identificar los métodos de optimización que presentan mejores resultados de convergencia y adaptabilidad a los cambios en los parámetros iniciales de cada problema individual de optimización y determinar la manera en que pueden ser mejorados, ajustando las etapas de tal forma que se obtenga una metodología más robusta de síntesis de mecanismos.

- Propuesta de metodología para síntesis óptima de mecanismos y validación de la misma a través de su comparación con metodologías existentes, de acuerdo con los criterios de optimización determinados, partiendo de la función objetivo del mecanismo y sus restricciones.
- Evaluar la metodología propuesta, comparando los resultados obtenidos por las síntesis de mecanismos reportadas para distintas aplicaciones y determinar el impacto o mejora obtenida.

## 1.5. Alcances y Limitaciones

El estudio se orientará a mecanismos planos de lazo cerrado. Los datos iniciales para realizar la síntesis consistirán en puntos de precisión con los que sea posible generar la trayectoria de un punto del mecanismo.

La metodología propuesta será capaz de adaptarse a cambios en los puntos de precisión que definen la trayectoria del mecanismo.

La evaluación de la metodología se realizará comparando los resultados de los mecanismos obtenidos en esta última con aquellos obtenidos mediante los métodos de optimización revisados.

## 1.6. Metodología propuesta

1. Revisión permanente y actualización continua del estado del arte de los métodos de optimización de síntesis de trayectoria de mecanismos planos para que el resultado de la investigación sea un producto actualizado que pueda tener aplicación en diferentes ámbitos.
2. Evaluar el funcionamiento de los algoritmos de síntesis óptima y determinar cómo incluir más restricciones en la función objetivo, debido a que los problemas de síntesis óptima de mecanismos deben considerar ciertas limitaciones

desde su planteamiento, por ejemplo, el espacio total donde se ubicará el mecanismo, la longitud mínima o máxima de los eslabones, la ubicación de los puntos de apoyo, entre otros.

3. Analizar el desempeño de los métodos de optimización existentes y verificar qué ventajas de cada uno podrían utilizarse para que formen parte de la MSM propuesta y en caso de tener desventajas, analizar cómo podrían disminuirse o eliminarse.
4. Proponer una MSM, que partiendo de una función objetivo genere mecanismos que satisfagan varios criterios de optimización o que se acerquen lo más posible a ellos, de mejor forma que los métodos existentes.
5. Obtener mecanismos por medio de métodos de síntesis ya definidos, observando si las soluciones obtenidas satisfacen los criterios de optimización establecidos.
6. Validar la MSM propuesta comparando la solución obtenida con aquellas generadas por otros métodos de síntesis para mecanismos de cuatro barras (SM4B), verificando que la solución sea mejor, de acuerdo con los criterios previamente establecidos.

---

# Capítulo 2

## Antecedentes

En este capítulo se presentan los conceptos relativos a optimización y síntesis de mecanismos. Se revisan y evalúan las ventajas y desventajas de algunos métodos de optimización y su aplicación a la síntesis de trayectoria de mecanismos planos.

### 2.1. Marco teórico: conceptos

#### *Optimización*

En términos matemáticos, la optimización es la minimización o maximización de una función sujeta a restricciones en las variables que la componen [9]. En dicho proceso se utiliza la siguiente notación, dada en la ecuación (2.1):

$$\min_{x \in R^n} f(x) \text{ sujeto a } \begin{cases} c_i(x) = 0 & i \in \varepsilon \\ d_k(x) \leq 0, & k \in j \end{cases} \quad (2.1)$$

Donde  $x$  es la variable independiente de la función  $f$ , que es la función que se busca minimizar,  $c$  y  $d$  indican restricciones a considerar en el proceso de minimización,  $\varepsilon$  y  $j$  son conjuntos de índices para las restricciones de igualdad y desigualdad, respectivamente.

La optimización se refiere al proceso de modificar la entrada o características de



un dispositivo o proceso matemático para obtener un mínimo o un máximo como salida [9].

### *Mecanismo*

Es una combinación de eslabones conectados entre sí por medio de pares cinemáticos de tal forma que se mueven uno sobre el otro con un movimiento relativo definido. Los mecanismos se han empleado desde la antigüedad con diferentes fines, debido a que hacen posible cambiar la dirección o magnitud de la fuerza aplicada. La relación entre la fuerza ejercida por el mecanismo y la fuerza aplicada al mismo se conoce como ventaja mecánica [10].

El propósito de cualquier mecanismo es transferir movimiento y trabajo mecánico de una o más fuentes de entrada a uno o más elementos de salida [10].

Los elementos que componen un mecanismo se conocen como eslabones, ligados unos con otros mediante pares o juntas cinemáticos que permiten ciertos movimientos. El número y tipo de sólidos junto al número y tipo de pares condicionan el tipo de movimiento que el mecanismo puede realizar (grado de movilidad).

Los mecanismos pueden clasificarse, de acuerdo con el movimiento que realizan sus componentes, como lineales, rotatorios, reciprocantes y oscilantes [10].

Un mecanismo plano es aquel en el que todos sus componentes se mueven en planos paralelos. Si se selecciona cualquier punto de este tipo de mecanismos y se dibuja su trayectoria, ésta es una curva en dos dimensiones que se encuentra en un plano paralelo a aquellos que contienen las trayectorias del resto de los puntos del mecanismo.

### *Grados de libertad (GDL)*

El número de GDL es el número de parámetros independientes que se requieren para definir completamente la posición del mecanismo en cualquier instante [10].

*Eslabón*

Un eslabón es un cuerpo rígido que tiene al menos dos nodos, es decir, puntos de conexión con otros eslabones. De acuerdo con el número de nodos que presentan, los eslabones pueden ser binarios, ternarios y cuaternarios, para dos, tres y cuatro nodos, respectivamente. En la Figura 2.1 se muestran los diferentes tipos de eslabones.

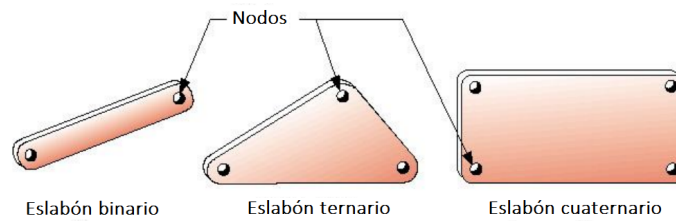


Figura 2.1: Tipos de eslabones por el número de nodos que presentan [10].

El término junta o par cinemático es una conexión entre dos o más eslabones que permite su movimiento relativo. Los pares cinemáticos se dividen en pares inferiores y pares superiores. En la Figura 2.2 se muestran las juntas cinemáticas de par inferior.

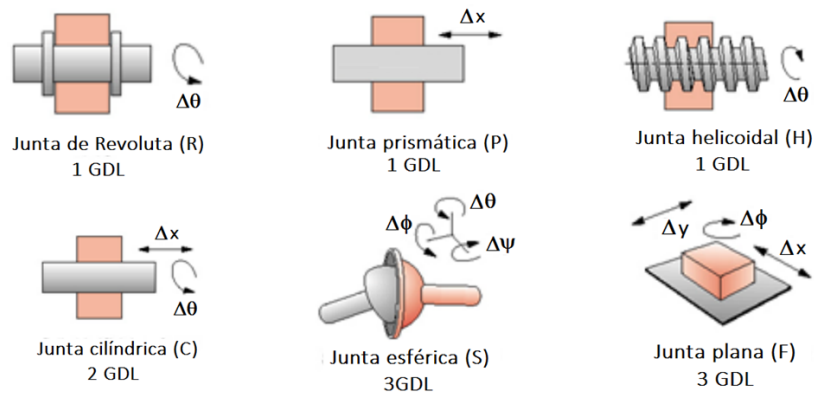


Figura 2.2: Juntas cinemáticas de par inferior [10].

La junta giratoria o revoluta sólo permite rotación relativa y, por consiguiente, posee un grado de libertad. Con frecuencia, este par se denomina articulación de pasador.

La junta prismática sólo permite movimiento relativo de deslizamiento o traslación y, por ende, se denomina casi siempre articulación de deslizamiento. También posee un solo grado de libertad.

El par de tornillo o par helicoidal cuenta con un solo grado de libertad porque los movimientos de deslizamiento y rotación están relacionados por el ángulo de hélice de la rosca.

El par cilíndrico permite tanto rotación angular como un movimiento de deslizamiento independiente. Por consiguiente, el par cilíndrico tiene dos grados de libertad.

El par esférico es una articulación de rótula y posee tres grados de libertad, una rotación alrededor de cada uno de los ejes coordenados.

Las juntas cinemáticas de par superior son aquellas que presentan un contacto de punto o de línea y pueden apreciarse en la Figura 2.3.

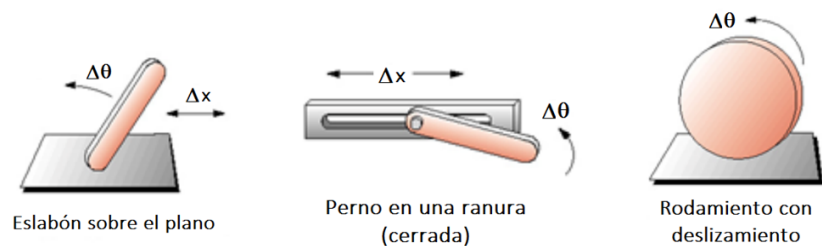


Figura 2.3: Juntas cinemáticas de par superior [10].

La síntesis de un mecanismo se refiere al diseño de un mecanismo que realice un movimiento predefinido, siga una trayectoria o tenga cierta velocidad en un eslabón o un punto del mecanismo (acoplador) [11], de tal manera que puede ser clasificada como síntesis de tipo, síntesis de número y síntesis dimensional.

La síntesis de tipo se utiliza para determinar qué elementos son ideales para conformar un mecanismo (eslabones, engranes, poleas, bandas, tipo de pares cinemáticos, etc.) para cierta tarea [12].

La síntesis de número se enfoca en hallar la cantidad de elementos que deben componer un mecanismo para lograr un determinado comportamiento [12].

La síntesis dimensional es generalmente la más utilizada ya que de ella se obtiene el tamaño adecuado del mecanismo que debe cumplir con cierta función. En este tipo de síntesis de mecanismos, se incluyen la síntesis de generación de movimiento, la síntesis de generación de trayectoria y la síntesis de generación de función [12].

La síntesis de generación de movimiento se refiere a determinar varias posiciones de un elemento completo del mecanismo (por ejemplo, un eslabón) y hallar la posición inicial de los demás elementos que, una vez que se mueven, logran alcanzar todas las posiciones deseadas del eslabón o elemento de interés [12].

La síntesis de generación de trayectoria se refiere a que solo un punto del mecanismo pase por cierta cantidad de puntos que están a lo largo de un recorrido (generalmente cerrado) [12].

Finalmente, la síntesis de generación de función relaciona el giro del eslabón de entrada con el giro del eslabón de salida, es decir, si el eslabón de entrada gira cierto ángulo la síntesis de función determina el mecanismo cuyo eslabón de salida gire un ángulo determinado en función del ángulo de entrada.

La diferencia entre cada una de ellas puede apreciarse en la Figura 2.4, donde en 2.4.a se observa la síntesis de generación de movimiento, donde el eslabón de interés es el  $AB$ ; se busca que en el mecanismo sintetizado, el eslabón completo pase de la posición  $A_1B_1$  a la  $A_2B_2$ . En 2.4.b se aprecia la síntesis de generación de trayectoria, donde el punto  $E$  se mueve a lo largo de la trayectoria  $l$  y en 2.4.c se muestra la síntesis de generación de función, donde el eslabón de entrada 2 afecta el movimiento del eslabón de salida 4 de manera controlada, es decir, conocido el giro de  $180^\circ$  del eslabón 2, se sabe que el eslabón 4 se mueve un ángulo predefinido.

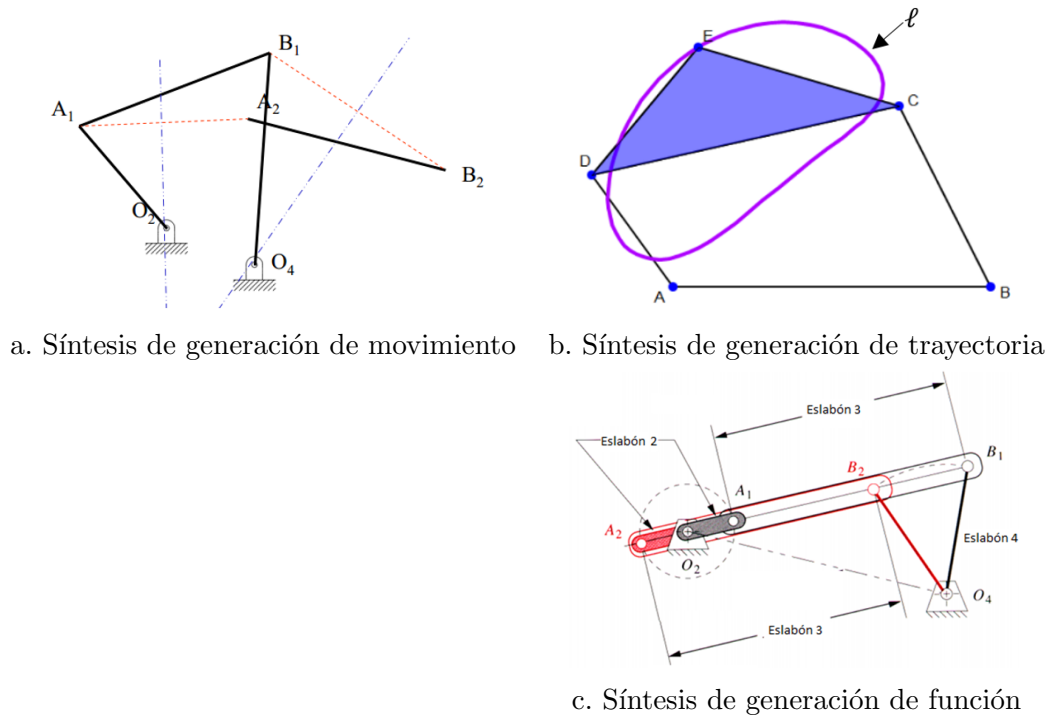


Figura 2.4: Diferentes métodos de síntesis de mecanismos. Modificada de [12].

## 2.2. Estado del Arte

La síntesis de mecanismos planos es un área de interés para muchos campos, ya que con ella se garantiza que el mecanismo alcance las posiciones deseadas.

La síntesis de generación de trayectoria originalmente se trabajaba como síntesis exacta, es decir, se construye un sistema de  $n$  ecuaciones con igual número de incógnitas que se resuelve para determinar la posición inicial del mecanismo, el cual pasa exactamente por cada uno de los puntos deseados.

Sin embargo, el número de incógnitas y ecuaciones se incrementa si aumentan los puntos por los que el mecanismo debe pasar, como se observa en la Tabla 2.1 donde para un mecanismo de Watt, para dos puntos de precisión se genera un sistema de 4 incógnitas y 16 ecuaciones.

Puntos	Ecuaciones	Incógnitas
2	4	16
3	8	18
4	12	20
5	16	22
6	20	24
7	24	26
8	28	28
9	32	30

Tabla 2.1: Relación entre número de puntos de precisión, ecuaciones e incógnitas para síntesis exacta de mecanismos [10].

En este caso, debe conocerse o bien proponerse un valor inicial para doce de las variables, de tal forma que se logre una solución única para el sistema de ecuaciones planteado y no un número infinito de soluciones, es decir, para sistemas de ecuaciones de más incógnitas que ecuaciones, el número infinito de soluciones se obtiene porque algunas variables dependen de otras, así que para que el sistema tenga solución única, es necesario asignar valores a tantas variables excedentes del número de ecuaciones del sistema.

Es posible observar que conforme se aumenta el número de puntos de precisión, el sistema se vuelve más complejo, requiriendo el uso de herramientas computacionales para encontrar la solución.

De la misma forma, se requieren cada vez menos datos hasta el caso límite de ocho puntos de precisión, donde se tienen 28 incógnitas y 28 ecuaciones, generando un sistema de solución única. Sin embargo, para nueve puntos de precisión (o más) se tienen más ecuaciones que incógnitas, por lo que se presenta un caso de sistemas de ecuaciones inconsistente, el cual no tiene solución.

En este caso es posible hacer uso de la síntesis óptima, que es aquella que, si bien no pasa exactamente por los puntos deseados, pasa lo más cerca posible de éstos y tiene la ventaja de que pueden agregarse puntos sin tener la restricción de volver

incompatible el sistema de ecuaciones de síntesis.

En las siguientes secciones se describirá el funcionamiento de algunos algoritmos de optimización representativos, así como sus resultados de aplicación para resolver problemas de síntesis óptima de mecanismos de cuatro barras (SOM4B).

### 2.2.1. Algoritmo de Recocido Simulado (SA)

Uno de los primeros algoritmos metaheurísticos en desarrollarse que continúan siendo populares es el de Recocido Simulado (SA del inglés *Simulated Annealing*) que es una técnica de búsqueda aleatoria para optimización global.

El algoritmo está basado en el proceso de recocido del acero, donde para reducir la dureza o para eliminar las tensiones internas del material, éste se somete durante un tiempo determinado a temperaturas elevadas, manteniéndose así para cambiar su estructura cristalina y luego se enfría lentamente para reducir esfuerzos internos.

Durante el proceso, la velocidad baja de enfriamiento da lugar a una estructura cristalina con la mínima energía, lo que produce tamaños grandes de los cristales de la estructura de forma que se reducen los defectos en estructuras metálicas.

El proceso de recocido involucra un control cuidadoso de temperatura y tasa de enfriamiento, mejor conocido como programa de recocido (annealing schedule). El cambio de temperatura debe llevarse a cabo de forma muy lenta, contrario a lo que sucede durante el proceso de temple de metales, donde el enfriamiento se realiza de forma muy rápida para aumentar la dureza del material al producir una estructura desordenada.

El recocido simulado fue desarrollado por Kirkpatrick et al. en 1983 y desde entonces se ha aplicado en casi cualquier área de optimización. Esto se debe principalmente a que los métodos basados en el gradiente y otras búsquedas determinísticas (métodos exactos) presentan la desventaja de no funcionar cuando las funciones a minimizar son discontinuas o bien, es posible que se estanquen en valores mínimos locales, mien-

tras que el SA puede manejar ambos casos sin problema [13].

Existen estudios que han demostrado que el recocido simulado converge al mínimo global óptimo si se tiene suficiente aleatoriedad en combinación con un enfriamiento muy lento [14].

Esencialmente el SA es un algoritmo de búsqueda conocido como caminata aleatoria (Random Walk), que converge si las condiciones son apropiadas, en este caso, dichas condiciones consisten en suficiente aleatoriedad y un gran número de iteraciones, representadas por un enfriamiento lento.

La idea básica del algoritmo SA es el uso de búsqueda aleatoria en términos de una cadena de Markov. La cadena de Markov es aquella en la que la probabilidad de cambio de un estado  $i$  a otro  $j$  depende sólo del estado actual, es decir, los estados previos al  $i$  no se reflejan en el estado siguiente.

Además, no sólo se aceptan cambios que mejoran la función objetivo, sino que también mantiene algunos que no sean ideales, con una probabilidad  $P$  (probabilidad de transición) dada por:

$$P = \exp\left(\frac{-\Delta E}{k_B T}\right) \quad (2.2)$$

Donde  $k_B$  es la constante de Boltzmann. Por simplicidad se usa  $k_B = 1$ .  $T$  es la temperatura para controlar el proceso de recocido y  $\Delta E$  es el cambio en el nivel de energía, la cual se relaciona directamente con la Función Objetivo ( $f$ ) a minimizar.

Para determinar si se acepta o no un movimiento, se usa como límite un número aleatorio  $r$ , tal que si la probabilidad de transición es mayor, se acepta. El número  $r$  se puede obtener directamente de una distribución uniforme de probabilidad.

Un parámetro muy importante en el algoritmo SA es la temperatura inicial  $T_0$ . La elección de la misma es crítica ya que para un cambio determinado  $\Delta f$ , si  $T$  es demasiado grande ( $T \rightarrow \infty$ ) entonces  $P \rightarrow 1$ , lo que significa que casi todos los cambios se aceptarían, sin importar que se alejen del mínimo de la función objetivo (FO).



Si  $T$  es demasiado baja ( $P \rightarrow 0$ ), entonces cualquier  $\Delta f > 0$  (peor solución) será raramente aceptada y como consecuencia, la diversidad de las soluciones será limitada. Sin embargo, cualquier mejora de  $\Delta f$  será casi siempre aceptada. En otras palabras, si la elección de  $T$  es muy alta no se alcanzarán mínimos, pero si es demasiado baja el sistema se estancará en un mínimo local.

Otro parámetro importante es la rapidez de cambio de la temperatura en el proceso de enfriamiento, de tal forma que es indispensable que el sistema se enfríe gradualmente de una temperatura alta hasta finalmente establecerse en un estado, en el mejor de los casos, correspondiente a un mínimo global de la FO.

Existen distintas formas de controlar la velocidad de enfriamiento, aunque dos esquemas de SA comúnmente usados son el que utiliza una tasa de enfriamiento lineal y el que utiliza una tasa geométrica de enfriamiento.

La variación lineal de la temperatura se muestra en la ecuación (2.3) donde el término  $\beta$  se multiplica por el tiempo  $t$  para ir reduciendo gradualmente la temperatura del sistema, en decrementos idénticos, mientras que la variación geométrica de la temperatura, observada en la ecuación (2.4) utiliza un valor predefinido  $\alpha < 1$  que multiplica a la temperatura inicial hasta que el sistema se estabiliza.

$$T = T_0 - \beta t \quad (2.3)$$

$$T(t) = T_0 \alpha^t, t = 1, 2, \dots, t_f \quad (2.4)$$

La ventaja del esquema geométrico es que la temperatura tiende a cero cuando el tiempo tiende a infinito, ya que  $\alpha < 1$ , por lo que no se requiere especificar el número máximo de iteraciones, Sin embargo, el proceso de enfriamiento debe ser lo suficientemente lento como para permitir que el sistema se estabilice fácilmente (en la práctica se usan valores de  $\alpha$  de entre 0.7 y 0.99)[13].

Para una temperatura dada, se requiere realizar múltiples evaluaciones de la función

objetivo, ya que se va realizando una variación aleatoria de la solución y determinando si la nueva solución mejora o no el valor actual de la FO. Si son muy pocas evaluaciones, se corre el riesgo de que el sistema no se estabilice y por consecuencia, no converja al óptimo global.

Por el contrario, si se tienen demasiadas evaluaciones, se consumen recursos (procesamiento de la computadora, tiempo) y el sistema convergerá lentamente, ya que el número de iteraciones para lograr la estabilidad puede ser exponencial al tamaño del problema.

Esto es una desventaja importante en el caso de la síntesis óptima de mecanismos, ya que las funciones objetivo son complejas e involucran al menos 10 variables, lo cual puede hacer que las evaluaciones consuman una gran cantidad de recursos.

En conclusión, existe un balance entre el número de evaluaciones y la calidad de la solución. Pueden realizarse muchas evaluaciones en pocos niveles de temperatura o pocas evaluaciones a muchos niveles de temperatura.

Existen dos formas de establecer el número de iteraciones: fijas o variables. En el primer caso se usa un número fijo de iteraciones para cada temperatura, mientras que el segundo intenta aumentar el número de iteraciones a temperaturas bajas de tal forma que el mínimo local pueda ser bien explorado.

El algoritmo SA descrito se muestra en la Figura 2.5. Este algoritmo es muy sencillo de implementar, siendo esa la razón por la que sigue utilizándose. Sin embargo, para el propósito específico de problemas de síntesis óptima de mecanismos planos, aunque es posible que con las suficientes iteraciones y lenta velocidad de enfriamiento, encuentre un valor mínimo suficiente para considerar que el problema se resolvió satisfactoriamente, requiere más recursos computacionales que otros algoritmos como los algoritmos evolutivos.

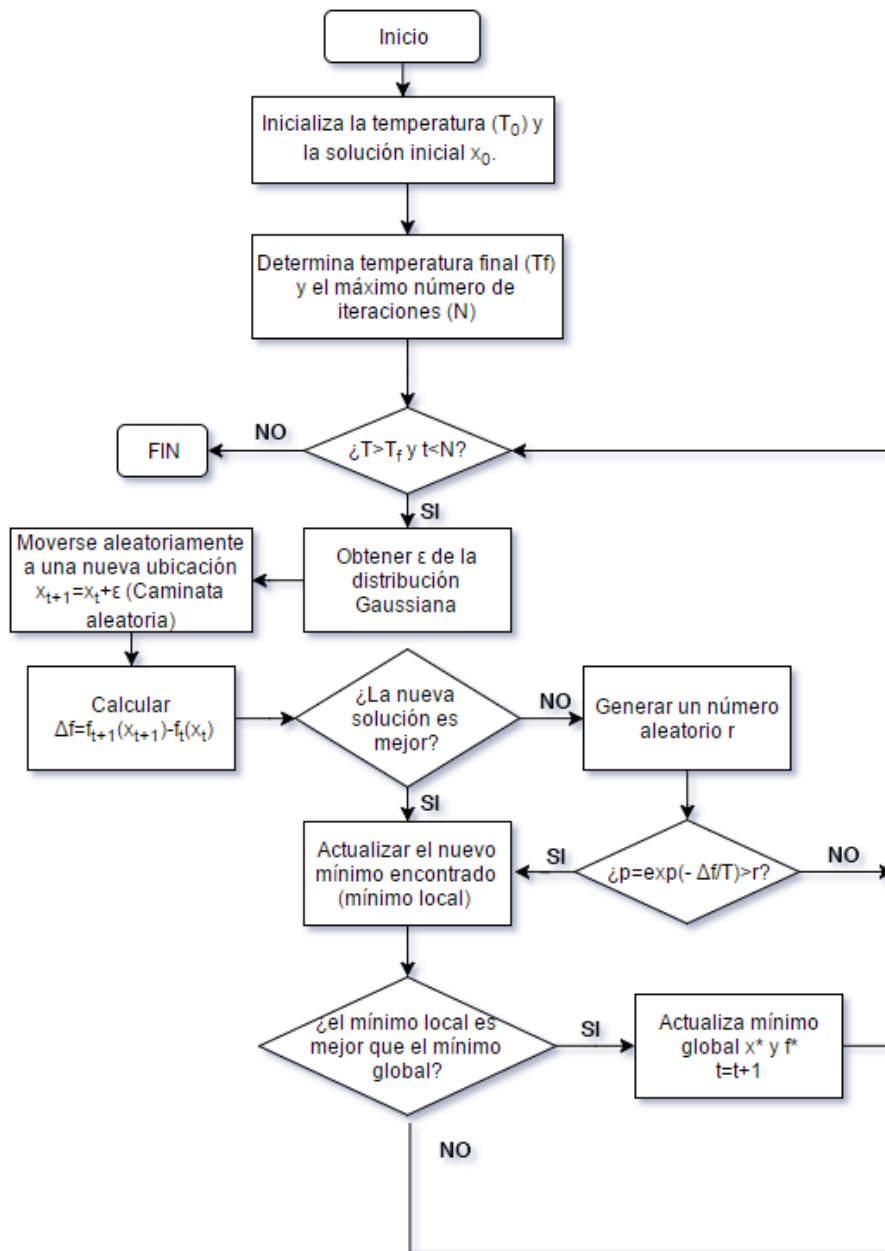


Figura 2.5: Algoritmo SA

### 2.2.2. Algoritmos Evolutivos (EA)

Los algoritmos evolutivos (EA, del inglés *Evolutionary Algorithms*) son programas computacionales que intentan resolver problemas complejos simulando el proceso evolutivo de la teoría de la evolución de Charles Darwin [16].

En los EA, un número de individuos se localizan en el espacio de búsqueda del problema y compiten continuamente por descubrir áreas óptimas en el espacio de búsqueda. Se espera que con el tiempo, los más exitosos evolucionen hasta descubrir la solución óptima [16].

Los individuos en los algoritmos evolutivos se representan comúnmente por vectores. Cada individuo representa una posible solución al problema. El algoritmo comienza con una población inicial de cierto tamaño con individuos generados de forma aleatoria dentro del espacio de búsqueda del problema de optimización.

Cuando se sustituyen los valores que componen el vector que representa al individuo en la función objetivo a optimizar, se obtiene un valor para dicha función. Este valor es conocido como la aptitud del individuo y entre menor sea su valor, se dice que está más cerca de un mínimo de la función objetivo, por lo que el individuo es una mejor solución respecto de otros.

El uso de operadores de recombinación y mutación modifica la lista inicial de individuos en el EA, que siguiendo la teoría evolutiva de Darwin, selecciona a aquellos con mejores valores de aptitud para que después de un cierto número de iteraciones en donde se producen nuevos individuos, éstos sean mejores que sus antecesores de tal forma que se alcancen valores menores de la función objetivo [16].

Entre los algoritmos evolutivos más conocidos se encuentran los algoritmos genéticos, el Lamarckismo y las modificaciones a los algoritmos genéticos, como el caso de los algoritmos genéticos con coeficientes de Fourier, que han demostrado ser capaces de alcanzar soluciones muy cercanas a valores óptimos para funciones objetivo en espacios de búsqueda complejos o discontinuos [16].

## Algoritmos genéticos

Los Algoritmos Genéticos (AG), desarrollados en las décadas de 1960 y 1970, son aquellos métodos adaptativos que pueden usarse para resolver problemas de búsqueda y optimización. Se basan en el proceso genético de los organismos vivos, considerando que a lo largo de las generaciones, las poblaciones evolucionan en la naturaleza de manera más o menos similar a la evolución debida a los principios de selección natural y supervivencia de los más fuertes [17].

Por imitación de este proceso, los AG son capaces de ir creando soluciones para problemas del mundo real, como para este caso, la síntesis de un mecanismo plano. La evolución de dichas soluciones hacia valores óptimos del problema depende en gran medida de una adecuada inicialización de la población.

Los principios básicos de los algoritmos genéticos fueron establecidos por Holland en 1975 y utiliza la terminología y principios de la genética para obtener soluciones a distintos problemas. En un algoritmo genético se debe tener un conjunto de soluciones iniciales, denominado **población**, del cual cada elemento es un **individuo**. A su vez, cada individuo está compuesto de una serie de características, o **cromosomas**, los cuales constituyen variables que forman una posible solución al problema de optimización [18].

Cada individuo debe satisfacer una **función de aptitud (FA)** que es una expresión matemática que contiene las variables que forman a cada individuo y que es distinta para cada problema o aplicación. Cuando se sustituyen los valores de los cromosomas del individuo en la función de aptitud, es decir, al evaluar el valor de las variables de un individuo en dicha función, el valor de esta debería minimizarse para la mejor solución.

A partir de la población inicial, se evalúa la aptitud de cada elemento y se reordenan de acuerdo con su valor de FA de menor a mayor, donde los primeros elementos son mejores debido a que obtienen los menores valores en la FA.

Una vez reordenados se comienza a realizar cruces entre los individuos para crear un nuevo conjunto de soluciones a partir de la población original, dando paso a una nueva **generación**.

El paso de una generación a otra debe realizarse a través de una serie de cruces entre pares de elementos, considerados **padres** para dar lugar a un par de nuevos elementos o **hijos**. El método seleccionado para las pruebas que se llevarán a cabo es una selección aleatoria del primer padre y un torneo determinístico para el segundo.

La selección aleatoria, como su nombre lo indica, consiste en elegir un elemento al azar del conjunto de elementos disponibles, siendo este el primer padre. Para seleccionar el segundo padre se eligen al azar dos elementos, pero en este caso de ambos se selecciona al elemento con un mejor valor de FA (menor) para ser el segundo padre.

El cruce entre los elementos seleccionados es **puntual**, es decir, se determina aleatoriamente un punto de cruce o número de cromosoma hasta el cual el primer hijo será idéntico al primer padre, es decir, compartirán los primeros cromosomas hasta el punto de cruce, y después el primer hijo será idéntico al segundo padre.

De manera similar, el segundo hijo es idéntico al segundo padre en los primeros cromosomas, hasta el punto de cruce y contiene los cromosomas del primer padre después del punto de cruce. En la Figura 2.6 se observa a dos padres compuestos por ocho cromosomas, cuyo punto de cruce es la mitad del elemento, produciendo dos hijos a partir de los valores de los cromosomas de los padres.

El algoritmo genético contiene también **mutaciones** para realizar un proceso de renovación de la población. La mutación depende del valor conocido como **tasa de mutación**  $T_m$ , de la misma manera que el proceso de cruce depende del valor de la **tasa de cruce**  $T_c$ .

La mayoría de los algoritmos genéticos cuenta con valores preestablecidos de tasa de cruce y de mutación, cuyos valores van de 0 a 1. El 0 representa un 0% de

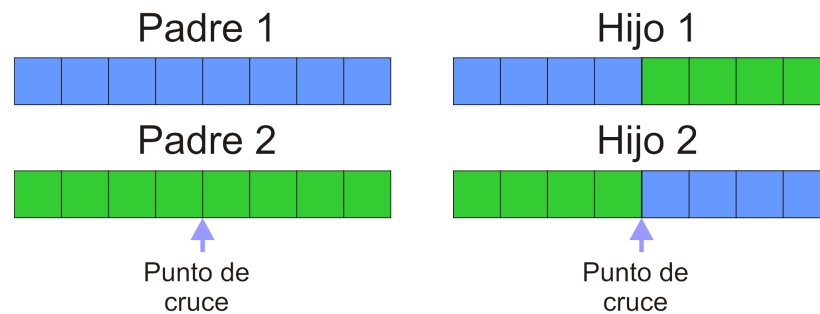


Figura 2.6: Tipo de cruce: puntual

posibilidad (tanto de cruce como de mutación) y 1 representa el 100% de probabilidad de que se lleven a cabo.

Convencionalmente los valores de la  $T_m$  son bajos y los de la  $T_c$  altos, para tener una mayor probabilidad de cruce que de mutación. Esto sin embargo depende de cada aplicación ya que una mayor tasa de mutación genera una población con elementos más variados que podrían ser una mejor alternativa que los que originalmente constituyen a la población.

Como se ha mencionado, los AG se aplican a la resolución de problemas en diferentes áreas y la síntesis de mecanismos no es la excepción. En este caso se considera síntesis óptima porque el acoplador pasa cerca de los puntos deseados, si bien no exactamente, a una distancia lo suficientemente pequeña para considerar que la solución satisface el problema.

La síntesis de mecanismos de cuatro barras (SM4B) es un problema cuyo estudio comenzó a finales del siglo XIX y continúa siendo estudiado en la actualidad, sin embargo, se requieren técnicas numéricas de síntesis cuando el número de **puntos de precisión**, es decir, los puntos que debe alcanzar el acoplador son numerosos o están dados en desorden [19].

Existen numerosos estudios de SOM4B. Algunos de los primeros trabajos en esta área se llevaron a cabo utilizando algoritmos evolutivos [15]. El uso de AG permite

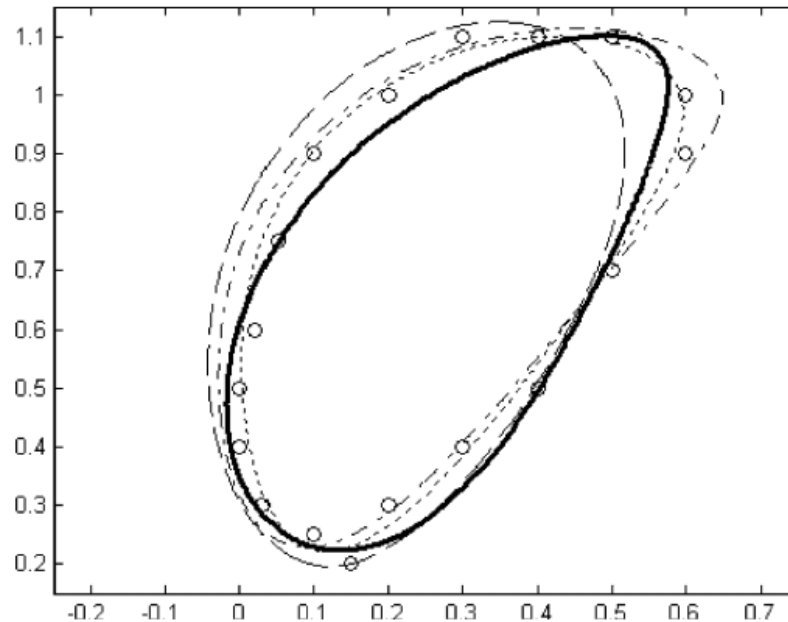


Figura 2.7: Trayectorias del acoplador de diferentes mecanismos sintetizados [20]

resolver problemas de síntesis de mecanismos de cuatro barras con múltiples puntos de precisión sacrificando ligeramente la exactitud con la que la trayectoria del acoplador pasa por los puntos deseados. En la Figura 2.7 se aprecian diferentes trayectorias que recorre el punto del acoplador de un mecanismo generado con AG con respecto a los puntos de precisión por los que se desea que pase. En línea continua se encuentra la mejor solución encontrada [20], la cual está dada por la trayectoria cuyas distancias a los puntos de precisión es menor. Esto se conoce como **error estructural** y es uno de los principales criterios de decisión para determinar que una solución es mejor que las demás [21].

Además de [23] y [19], que fueron de los primeros autores que aplicaron algoritmos genéticos a la síntesis de mecanismos, en [24] se utilizaron debido a que son fáciles de implementar y requieren calcular únicamente el valor de la función objetivo, usando operadores genéticos para explorar y evaluar las posibles soluciones. También es posible obtener un conjunto de soluciones que pueden evaluarse posteriormente para determinar si satisfacen criterios de diseño adicionales.



Los AG son ampliamente utilizados para sintetizar mecanismos de cuatro barras, como se observa en [20] y [22] que realizan estudios comparativos de mecanismos sintetizados con AG comparando los resultados del algoritmo con los obtenidos por [24], concluyendo que aún cuando ambos métodos utilizan AG, [20] es más sencillo de implementar y tiene una convergencia más rápida.

Estudios adicionales sobre AG aplicados a la síntesis de mecanismos reconfigurables, es decir, aquellos mecanismos cuya estructura topológica y parámetros cinemáticos o dinámicos puedan ser ajustados durante el proceso de movimiento han sido reportados por [25] y [26].

Una estructura topológica contempla los tipos de juntas cinemáticas, restricciones dimensionales del mecanismo y los parámetros cinemáticos, que serían en este caso la posición, velocidad y aceleración del mecanismo en cualquier punto de su trayectoria o bien las fuerzas estáticas a las que se encuentre sometido. Los parámetros dinámicos se refieren a las fuerzas de inercia que actúan sobre el mecanismo a medida que cambia su aceleración [12].

Estos elementos también fueron considerados en la síntesis de un mecanismo de Watt para aplicación en una prótesis de rodilla, sintetizado utilizando AG. En este caso se comprobó la facilidad de implementación de estos algoritmos evolutivos, así como su fuerte dependencia de la población inicial para encontrar una solución óptima al problema [27][28].

Un estudio realizado en 2014 consideró el uso de algoritmos genéticos para obtener la síntesis óptima de trayectoria, comparando los resultados obtenidos con los generados mediante las herramientas disponibles en el software Matlab<sup>©</sup>, que realizan optimización numérica restringida no lineal. Una vez obtenidos los mecanismos, se utilizó MS Excel<sup>©</sup> para dibujar y simular el movimiento de los mismos, para verificar que el punto de interés en el eslabón acoplador alcance los puntos deseados [29].

## Lamarckismo

Además de los AG existen otros algoritmos evolutivos que se han utilizado para resolver problemas de optimización no lineal con restricciones, como la síntesis de mecanismos de cuatro barras. Estos algoritmos tienen el fin de aprovechar al máximo la capacidad de cálculo y velocidad de procesamiento de las computadoras actuales y se basan en conceptos biológicos.

En el caso del Lamarckismo, se utilizan los mismos principios de generación de una población inicial, selección de padres y cruces puntuales para generar hijos y dar lugar a nuevas generaciones que deseablemente sean mejores que las anteriores.

La diferencia con los AG radica en el operador de mutación. En los AG tradicionalmente se realiza la mutación después del cruce, es decir, primero se producen los hijos a partir del cruce de los padres y después estos mutan si la tasa de mutación es mayor que el número aleatorio generado para determinar si se lleva a cabo la mutación.

Esto tiende a renovar la población si esta se ha estancado. Sin embargo, en la teoría de Lamarck estas mutaciones se llevan a cabo antes de realizar el cruce, es decir, los elementos considerados como padres mutan o mejoran pasando a los hijos directamente estas mejoras [7].

La mutación antes del cruce se realiza, porque de acuerdo con la teoría de Lamarck, los cambios en los individuos no se realizan de manera gradual a través de modificaciones que tardan varias generaciones en llevarse a cabo, sino que el propio individuo al conocer sus limitantes se esfuerza tratando de eliminarlas haciendo posible alcanzar pequeños cambios que se transmiten entonces a la nueva generación.

Esta teoría es biológicamente incorrecta, sin embargo, es posible aplicarla a problemas de optimización obteniendo buenos resultados.

Además de la mutación previa al cruce, el algoritmo evolutivo planteado por [7]

considera métodos de penalización agregados a la función objetivo para mejorar la calidad de las soluciones encontradas. En este caso se considera sumar un término a la función objetivo que sea 0 en caso de que el mecanismo cumpla con la condición de Grashof y un número grande (por ejemplo 1000) si el mecanismo no cumple con dicha condición.

La condición de Grashof es una característica deseable en los mecanismos y consiste en garantizar que al menos un eslabón del mismo realiza giros completos. Esto debido a que en muchas ocasiones los mecanismos van acoplados a motores que tienen un único sentido de giro, por lo que no sería posible regresar el mecanismo si este llega a una posición a partir de la cual el giro se detiene y tiene que volver por la misma trayectoria. En este caso se dice que el mecanismo no cumple la Condición de Grashof.

Al igual que en los AG, el algoritmo basado en la Teoría de Lamarck, si bien es sencilla de comprender e implementar, requiere inicializar la población con elementos a partir de los cuales pueda obtenerse una solución al problema de optimización a través de los operadores genéticos de cruce y mutación.

El algoritmo basado en la teoría de Lamarck se denominó algoritmo de mejora selectiva limitada (LSI del inglés *Limited Selective Improvement*) [7].

### **Algoritmos genéticos (AG) con coeficientes de Fourier**

Un caso especial del uso de AG es la combinación de estos y el uso de Coeficientes de Fourier (FC) para sintetizar mecanismos. En [30] se realizó la síntesis de un mecanismo de cuatro barras a partir de un generador de curvas. Para ello se utilizaron coeficientes de Fourier para representar una curva cerrada. A partir de esta curva, compuesta por una serie de funciones seno y coseno, se genera la síntesis del mecanismo [31].

### 2.2.3. Lógica difusa (FL)

Debido al creciente interés que se ha tenido en los últimos años en la síntesis de mecanismos, se han aplicado diferentes metodologías además de los algoritmos evolutivos o bien en combinación con ellos. Una de las técnicas que ha cobrado más auge en el siglo XXI es la Lógica Difusa (FL del inglés *Fuzzy Logic*).

La lógica difusa es una lógica alternativa a la lógica clásica, en esta última, las condiciones planteadas se cumplen o no se cumplen. Sin embargo, en la lógica difusa es posible considerar cierto grado de vaguedad en las cosas evaluadas, es decir, verificar un cumplimiento parcial de la condición planteada.

En el mundo real las situaciones son ambiguas o imprecisas por naturaleza, por lo que el razonamiento humano con frecuencia decide con base en este tipo de información. La lógica difusa fue diseñada para imitar el comportamiento de toma de decisiones del ser humano.

En [33] se utiliza una combinación de lógica difusa con algoritmos genéticos para realizar la síntesis de trayectoria de un mecanismo. En este caso el AG realiza el proceso de búsqueda de posibles soluciones y se agrega un controlador que hace uso de la lógica difusa para modificar los intervalos de búsqueda o reiniciarlos.

De acuerdo con los resultados reportados por [33], la combinación de AG y lógica difusa genera mejores resultados que los algoritmos genéticos por sí solos. En la Figura 2.8 se muestra la comparación del error obtenido a través de las diferentes etapas de búsqueda (generaciones), observando que el error inicial cuando se utiliza lógica difusa es menor que cuando se utilizan los AG de forma independiente.

Asimismo, el error estructural final demostró ser menor en el caso de la combinación AG-FL [33].

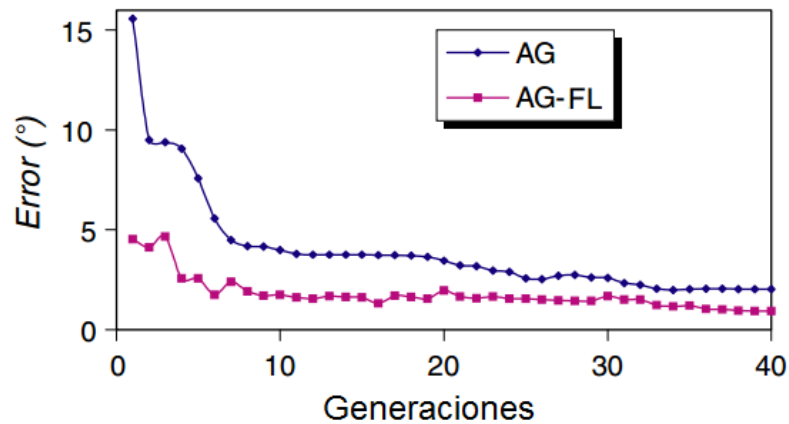


Figura 2.8: Comparación de efectividad entre AG y AG-FL [33].

#### 2.2.4. Búsqueda armónica (HS)

Existen también otros métodos de optimización menos conocidos que los AG o la lógica difusa, como es el caso de la búsqueda armónica. Este es un algoritmo solamente explorado en los últimos años que consiste en una técnica de optimización basada en un proceso de improvisación musical. Existen diferentes versiones de Búsqueda Armónica (HS del inglés *Harmonic Search*) que se han usado para resolver problemas del mundo real, aunque la gran mayoría de sus aplicaciones hasta ahora son orientadas a la optimización combinatoria [34].

Sin embargo, también es posible aplicar la búsqueda armónica a la síntesis de mecanismos. En [35] se utilizó para generar la síntesis de trayectoria de un mecanismo de Grashof de cuatro barras. El objetivo fue minimizar el error estructural. Se realizaron pruebas de comparación con síntesis de trayectoria realizada con AG obteniéndose que la búsqueda armónica converge a una solución aceptable con más rapidez. Una de las ventajas del uso de HS es la posibilidad de extender el proceso para realizar la síntesis de trayectoria de mecanismos de más eslabones [36].

De la misma forma, en [37] se hace uso del algoritmo HS para sintetizar un mecanismo de cuatro barras que sigue una trayectoria específica. Los resultados reportados indican un control muy preciso en el seguimiento de la trayectoria deseada para el

mecanismo.

Entre las ventajas del uso de la HS en la síntesis de mecanismos se encuentra la simplicidad del algoritmo que lo hace apto para diseñar otros tipos de mecanismos o dispositivos mecánicos, sin embargo, presenta la desventaja de que requiere una adecuada interpretación y su formulación es diferente para cada problema en particular, así como sus restricciones [37].

### 2.2.5. Mapeo cinemático (KM)

El mapeo cinemático (KM del inglés *Kinematic Mapping*) es una técnica que considera los desplazamientos relativos de dos cuerpos rígidos en el plano, considerando que uno de ellos permanece fijo y el otro puede moverse libremente. A través de esta consideración pueden generarse soluciones a problemas de cinemática directa de tres grados de libertad, principalmente en manipuladores paralelos [38].

En [39] se utiliza el mapeo cinemático para desarrollar de forma preliminar un algoritmo para síntesis aproximada de mecanismos de cuatro barras. La curva deseada del acoplador consiste en la curva generada por la intersección de dos superficies, haciendo que las ecuaciones de síntesis del mecanismo se encuentren definidas en exceso y aún cuando el algoritmo se reportó incompleto por no encontrar soluciones no triviales, los resultados obtenidos muestran que se llevaría a cabo un desarrollo posterior.

Es por ello que en [40] se toma como base el algoritmo [39] para realizar la síntesis de un mecanismo de cuatro barras a partir de  $n$  posiciones deseadas del acoplador, encontrando el mejor mecanismo que puede alcanzar, casi exactamente, dichas posiciones.

Esto se logró resolviendo un problema de minimización sin restricciones no lineal equivalente, usando mapeo cinemático para determinar la distancia mínima entre las  $n$  posiciones y la trayectoria real. El procedimiento comprobó su efectividad en

la SM4B [39].

### 2.2.6. Optimización por enjambre de partículas (PSO)

La optimización por enjambre de partículas (PSO del inglés *Particle Swarm Optimization*) es un algoritmo que al igual que los algoritmos evolutivos fue inspirado en la biología. Fue desarrollado por Kennedy y Eberhart en 1995 [32]. El algoritmo PSO se basa en el comportamiento que presentan los individuos que pertenecen a enjambres o parvadas.

La búsqueda se realiza dentro de una población, en la que los individuos deben desplazarse con base en la experiencia y conocimiento de los elementos vecinos. Inicialmente se define una colección o enjambre de partículas, en el cual la posición de cada partícula en  $N$  dimensiones representa una solución particular al problema de optimización [41].

A diferencia de los algoritmos evolutivos, la optimización por enjambre de partículas no utiliza mutaciones o cruces. El movimiento de las partículas se determina por aleatoriedad de números reales y comunicación global entre las partículas del enjambre [41].

Algunos nuevos algoritmos, como el algoritmo de luciérnagas, se desarrollaron a partir de los componentes usados en PSO, por lo que se considera a este último como un algoritmo pionero para la resolución de problemas de optimización [41].

Cada posible solución (o posición de cada partícula) se mueve para obtener un peso, el cual se calcula de acuerdo con qué tan bien dicha posición resuelve el problema de optimización, de manera similar al cálculo de la función de aptitud en los AG [41].

Una vez determinados los pesos de cada individuo, estos tratan de moverse hacia la mejor solución de acuerdo a su posición actual, su velocidad individual, el valor de inteligencia (o peso) individual y el valor de inteligencia del grupo (o mejor peso) de tal manera que a través de una serie de iteraciones, el conjunto de soluciones tiende

a moverse hacia la solución del problema [41].

Es decir, el algoritmo PSO busca en el espacio de una FO ajustando las trayectorias de las soluciones individuales, o partículas. El movimiento de una partícula en el enjambre consiste en dos principales componentes: uno estocástico, es decir, dependiente de procesos o cálculos que involucran valores aleatorios y uno determinístico, donde los valores obtenidos son siempre iguales para los mismos valores de entrada [41].

Cada partícula es atraída hacia la posición del mejor global actual  $g^*$  y su mejor ubicación  $x_{i^*}$ , histórica (componente determinístico), mientras que al mismo tiempo, se presenta una tendencia a moverse de forma aleatoria (componente estocástico) [41].

Cuando una partícula encuentra una posición que es mejor que cualquier posición encontrada antes, se actualiza la mejor posición para la partícula  $i$ . Entonces existe un mejor actual para todas las  $n$  partículas en cualquier tiempo  $t$  durante las iteraciones.

El objetivo es encontrar el mejor global entre todas las soluciones hasta que el objetivo ya no mejore o después de un cierto número de iteraciones [41].

El movimiento de las partículas sistemáticamente se representa como un movimiento de la partícula que puede ir en cualquier dirección, pero siempre es atraída por el mínimo local y el mínimo global.

Esto puede observarse en la Figura 2.9 donde la partícula  $i$  tiende a moverse siempre hacia el mínimo local  $x_{i^*}$  (partícula que alcanza un mínimo en esa iteración) y el mínimo global  $g^*$  (partícula con el valor mínimo histórico durante la evaluación).

El algoritmo PSO se muestra en la figura 2.10. La optimización por enjambre de partículas ha sido aplicada al diseño cinemático de mecanismos cuyos elementos se mueven en planos paralelos. Este tipo de mecanismos presentan una transmisión de movimiento más fluida.



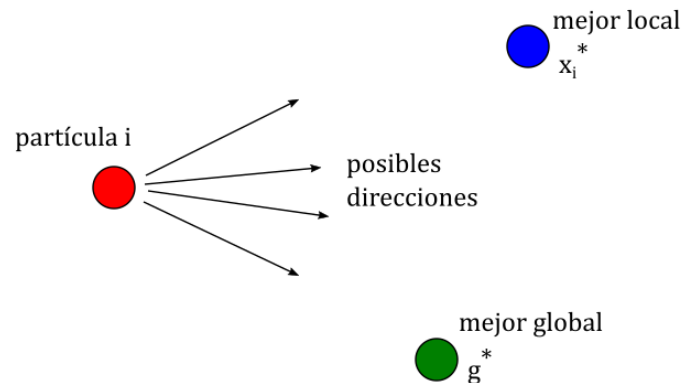


Figura 2.9: Tendencia de movimiento de una partícula en el algoritmo PSO [41].

En [42] se realizó un análisis cinemático de un mecanismo controlable de 5 barras, realizando el proceso de optimización usando el algoritmo PSO, dando como resultado una mejora en la eficiencia del proceso de búsqueda por medio de la gradual reducción del espacio de las posibles soluciones.

Otros estudios que utilizan el algoritmo PSO para síntesis óptima de mecanismos planos, concretamente en mecanismos de cuatro barras, no consideran el ángulo de giro del eslabón de entrada para alcanzar los puntos de precisión (síntesis sin temporización) debido a que cuando se define la trayectoria con un mayor número de puntos, se incrementa el tiempo de cálculo, sin que esto garantice alcanzar la solución exacta, por lo que también se considera el mínimo error estructural como criterio de optimización [43].

En 2010, se realizó una variante del algoritmo PSO, conocida como optimización por enjambre de partículas asíncrona multiobjetivo (MOPAPSO del inglés *Multi-Objective Parallel Asynchronous Particle Swarm Optimization*) con el fin de realizar síntesis óptima garantizando que los mecanismos obtenidos cumplen con la condición de Grashof, probando el algoritmo en mecanismos de hasta cinco eslabones [44].

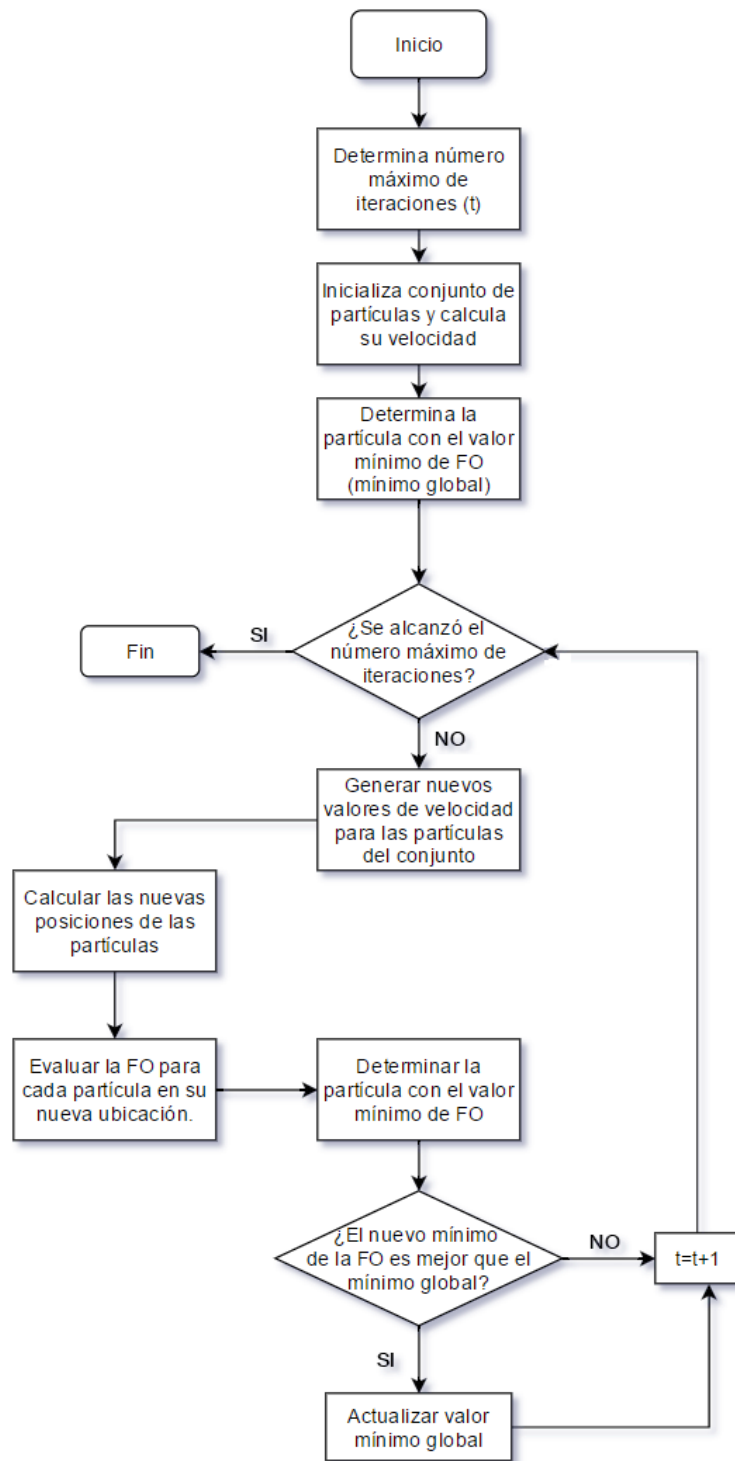


Figura 2.10: Algoritmo PSO.

### 2.2.7. Algoritmo del banco de krill modificado (MKH)

Otro algoritmo inspirado en sucesos biológicos es el algoritmo del banco de krill (KH del inglés *Krill Herd*) que se desarrolló para resolver problemas de optimización. Este algoritmo se basa en la simulación del comportamiento de los individuos en un banco de krill [46].

La distancia mínima entre cada individuo del banco y la fuente de comida, así como la distancia a la que se encuentra la mayor densidad de individuos en el banco se consideran la función objetivo que determinan el movimiento de un individuo o krill.

La posición del individuo depende del tiempo y está relacionada con tres factores básicos: el movimiento provocado por otros individuos, la actividad externa y la difusión aleatoria del banco. Este algoritmo mostró ser capaz de resolver eficientemente problemas típicos de optimización [46].

El algoritmo KH se modificó en [47] para obtener soluciones al problema de optimización planteado para generar la síntesis dimensional de un mecanismo de cuatro barras, por medio de generación de trayectoria. Se realizaron algunos cambios para mejorar la eficiencia del algoritmo, dando lugar al algoritmo del banco de krill modificado (MKH del inglés *Modified Krill Herd*).

Las modificaciones realizadas consistieron en la inicialización de la función de aptitud (Fitness Function) así como de los vectores que representan la localización de la fuente de alimentación.

Dichas modificaciones son un reflejo del comportamiento real de los individuos de krill en el banco al que pertenecen, es decir, se busca emular la tendencia del individuo a buscar una posición con cierta densidad de individuos a su alrededor (no en un espacio saturado de individuos o aislado totalmente) así como tener la fuente de comida relativamente cerca.

La segunda modificación realizada en [47] es el reemplazo de la función de cruce,

similar en el algoritmo KH a la función que se lleva a cabo en los AG, por una combinación entre los valores de función de aptitud obtenidos en una iteración. Debido a estos cambios, el algoritmo MKH mostró resultados eficientes en la síntesis de trayectoria de un mecanismo de cuatro barras.

### 2.2.8. Optimización por algoritmo de luciérnagas (FA)

Los algoritmos metaheurísticos inspirados en la naturaleza, especialmente los que se basan en inteligencia de partículas, se han desarrollado durante los últimos 10 años [45][48] a partir de la aparición del algoritmo PSO. Entre ellos, destaca el conocido como algoritmo de luciérnagas (FA, del inglés *Firefly Algorithm*) que apareció en 2008 y cuya literatura se ha expandido dramáticamente en diversas aplicaciones desde entonces [49].

El algoritmo FA fue desarrollado por Xin-She Yang a finales de 2007 y publicado en 2008. Se basa en los patrones de comportamiento de las luciérnagas tropicales [50]. Estos patrones se refieren a los generados por la luz parpadeante de las luciérnagas que se observa en verano en regiones tropicales y templadas.

Existen alrededor de 2000 especies de luciérnagas y la mayoría generan parpadeos rápidos y rítmicos. Estos patrones de luces parpadeantes con frecuencia son únicos para distintas especies.

La luz parpadeante se produce por un proceso de bioluminiscencia, sin embargo, las verdaderas funciones de este tipo de señales aún son objeto de debate, aunque dos de las funciones fundamentales comprobadas son atraer pareja (comunicación) y atraer presas potenciales.

Además, es posible que la luz parpadeante sirva como un mecanismo de advertencia para recordar a los potenciales depredadores que las luciérnagas poseen un sabor desagradable (amargo) [53].

Se sabe que la intensidad de la luz a una distancia particular  $r$  de la fuente, obedece

la ley de inversos cuadrados, es decir, la intensidad de la luz  $I$  disminuye mientras la distancia  $r$  se incrementa como se establece en (2.5):

$$I \propto \frac{1}{r^2} \quad (2.5)$$

Además, el aire absorbe la luz, lo que la hace más y más débil mientras mayor es la distancia. Estos factores combinados hacen que más luciérnagas sean visibles a un límite de distancia, usualmente, algunos cientos de metros en la noche, que es lo suficiente para que las luciérnagas puedan comunicarse.

La luz parpadeante puede formularse de forma que se asocie con la función objetivo a optimizar, lo que hace posible adaptar el algoritmo a múltiples áreas de investigación [51].

El algoritmo FA estándar sigue algunas reglas básicas que lo hacen ligeramente diferente del proceso real de comunicación entre luciérnagas. Estas reglas son las siguientes:

1. Todas las luciérnagas son unisex, por lo que una luciérnaga puede ser atraída por otra, con independencia de su sexo.
2. El atractivo es proporcional al brillo de la luciérnaga, es decir, para un par de luciérnagas, la menos brillante se moverá hacia la más brillante. El atractivo es proporcional al brillo y ambos disminuyen al aumentar la distancia. Si no hay una luciérnaga más brillante, el movimiento será aleatorio.
3. El brillo de una luciérnaga se determina por su valor de función objetivo.

El diagrama de flujo del algoritmo FA se presenta en la Figura 2.11. En el algoritmo FA hay dos factores importantes: la variación de la intensidad de la luz y la formulación del atractivo. Por simplicidad se puede asumir que el atractivo de una luciérnaga está determinado por su brillo, el cual se asocia con la FO.

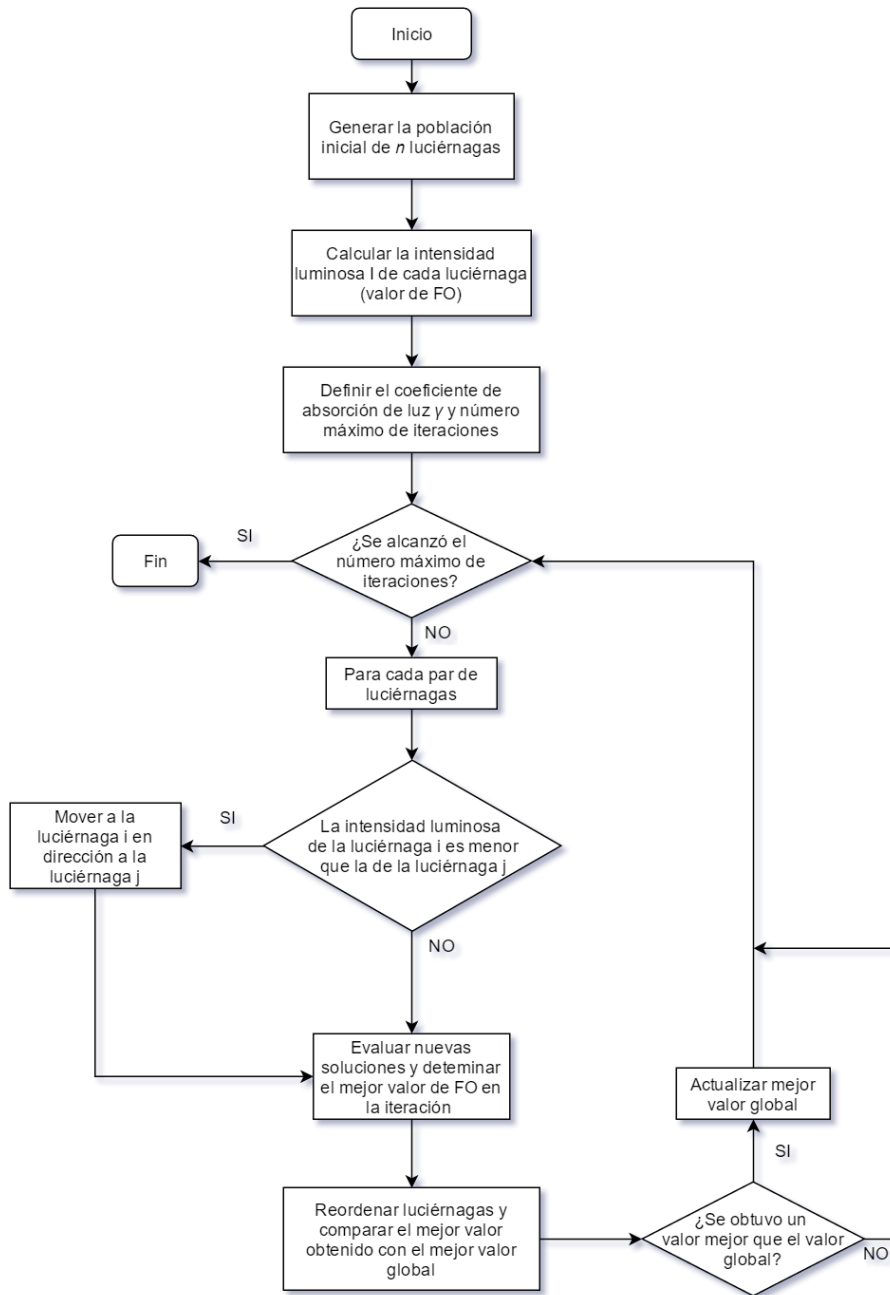


Figura 2.11: Algoritmo FA

En la mayoría de los problemas de optimización el brillo  $I$  de una luciérnaga es proporcional a su valor de FO, pero el atractivo es relativo y será variable dependiendo de la distancia  $r_{ij}$  entre las luciérnagas  $i$  y  $j$ . Además, la intensidad luminosa disminuye conforme aumenta la distancia de la fuente y la luz es absorbida por el medio.

En la forma más simple, la intensidad de la luz está dada por la ley de inversos cuadrados mencionada en la ecuación 2.5 que se convierte en:

$$I(r) = \frac{I_s}{r^2} \quad (2.6)$$

Donde  $I_s$  es la intensidad luminosa en la fuente.

Para evitar la singularidad que puede presentarse cuando la distancia  $r$  entre dos luciérnagas es cero, se utiliza la aproximación Gaussiana mostrada en la ecuación 2.7

$$I(r) = I_0 e^{-\gamma r^m} \quad (2.7)$$

Donde  $I_0$  es la intensidad de luz original a una distancia cero ( $r = 0$ ),  $\gamma$ ,  $r$  y  $m$  constantes determinadas empíricamente, cuyos valores se encuentran entre cero y uno.

En el caso del algoritmo FA el atractivo que cada luciérnaga ejerce sobre cualquier otra es una analogía de la intensidad luminosa de la ley de inversos cuadrados y su aproximación Gaussiana, es decir, se puede definir el atractivo de una luciérnaga como:

$$\beta = \beta_0 e^{-\gamma r^m} \quad (2.8)$$

Donde  $\beta_0$  es el atractivo a una distancia  $r = 0$ . Sin embargo, al ser  $\frac{1}{1+r^2}$  más rápido de calcular que una función exponencial, puede aproximarse la función de atractivo como:

$$\beta = \frac{\beta_0}{1 + \gamma r^2} \quad (2.9)$$

Mientras que la distancia euclidiana  $r$  entre dos luciérnagas está dada por:

$$r_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + \dots} \quad (2.10)$$

Los puntos suspensivos indican la diferencia entre las variables que componen las luciérnagas que se están evaluando. Finalmente, el movimiento de una luciérnaga  $i$  que es atraída hacia otra, más brillante  $j$  está dada por:

$$x_i^{t+1} = x_i^t + \beta_0 e^{-\gamma r_{ij}^2} (x_j^t - x_i^t) + \alpha \epsilon_i^t \quad (2.11)$$

Para síntesis de mecanismos planos aún no hay artículos reportados del uso del algoritmo de luciérnagas. Esto podría ser debido a que el algoritmo, si bien es relativamente sencillo de implementar, funciona muy bien para valores reales de la FO, sin embargo, en el caso de síntesis de mecanismos, algunos de los valores calculados de FO para las luciérnagas pueden resultar complejos, lo que hace difícil hacer la comparación entre valores para llevar a cabo el movimiento de las luciérnagas y lograr una convergencia hacia un valor mínimo de la Función Objetivo.

### 2.2.9. Optimización por algoritmo de colonia artificial de abejas (ABC)

El algoritmo de colonia artificial de abejas (ABC, del inglés Artificial Bee Colony) es un algoritmo de optimización heurística inspirado en el proceso de búsqueda de alimento que llevan a cabo las abejas y modificado para generar tres fases dentro de la misma iteración del ciclo de optimización. Fue propuesto en 2005 por Karaboga [52].

En el algoritmo ABC, se tienen tres tipos de abejas: obreras, espectadoras y exploradoras, además se considera que sólo hay una abeja obrera por cada fuente de alimento.

Si la fuente de alimento es buena, la abeja obrera lo indica a la abeja espectadora, en caso de que la fuente no sea considerada buena, se abandona y la abeja obrera que estaba en ella se convierte en una abeja exploradora.

En esta etapa se busca una nueva solución si se observa que no mejora a lo largo de las iteraciones. El trabajo de la abeja espectadora es monitorear las fuentes de



alimento y conservar la mejor.

En el uso del algoritmo ABC en optimización de funciones, usualmente se busca la solución con el valor más bajo de función objetivo. Si la abeja obrera tiene una buena fuente de alimento (valor bajo de FO) realizará movimientos alrededor de ésta, encontrando posiblemente una mejora en el valor de la FO.

Si por el contrario, no tiene asignada una fuente de alimento apropiada, se convertirá en abeja exploradora, moviéndose aleatoriamente en el espacio de búsqueda y con posibilidad de encontrar fuentes atractivas de alimento.

El diagrama de flujo del algoritmo ABC se muestra en la Figura 2.12

### **2.2.10. Optimización por algoritmo de búsqueda del cuco (CS)**

La naturaleza y los fenómenos que en ella se presentan han sido la inspiración de muchos algoritmos de optimización, como los algoritmos ya mencionados de optimización por enjambre de partículas, algoritmos genéticos, algoritmos de luciérnagas y algoritmo de colonia artificial de abejas, entre otros.

El algoritmo de búsqueda del cuco (CS, del inglés Cuckoo Search), desarrollado en 2009 por Xin-She Yang y Suash Deb [51], es uno de los últimos algoritmos inspirados en la naturaleza que ha sido propuesto y está basado en el parasitismo de postura en algunas especies de cucos, aves que depositan sus huevos en otros nidos para que otra especie los incube y así mejorar sus posibilidades de sobrevivencia.

En el algoritmo CS un cuco representa un vector de datos que al ser sustituidos en una función objetivo proporcionan un valor. Se espera que este valor sea lo más cercano a cero, lo cual representa un valor mínimo global.

Estudios recientes indican que el algoritmo CS presenta una mejor convergencia que los algoritmos PSO y AG [51]. El algoritmo CS utiliza paseos aleatorios, de manera

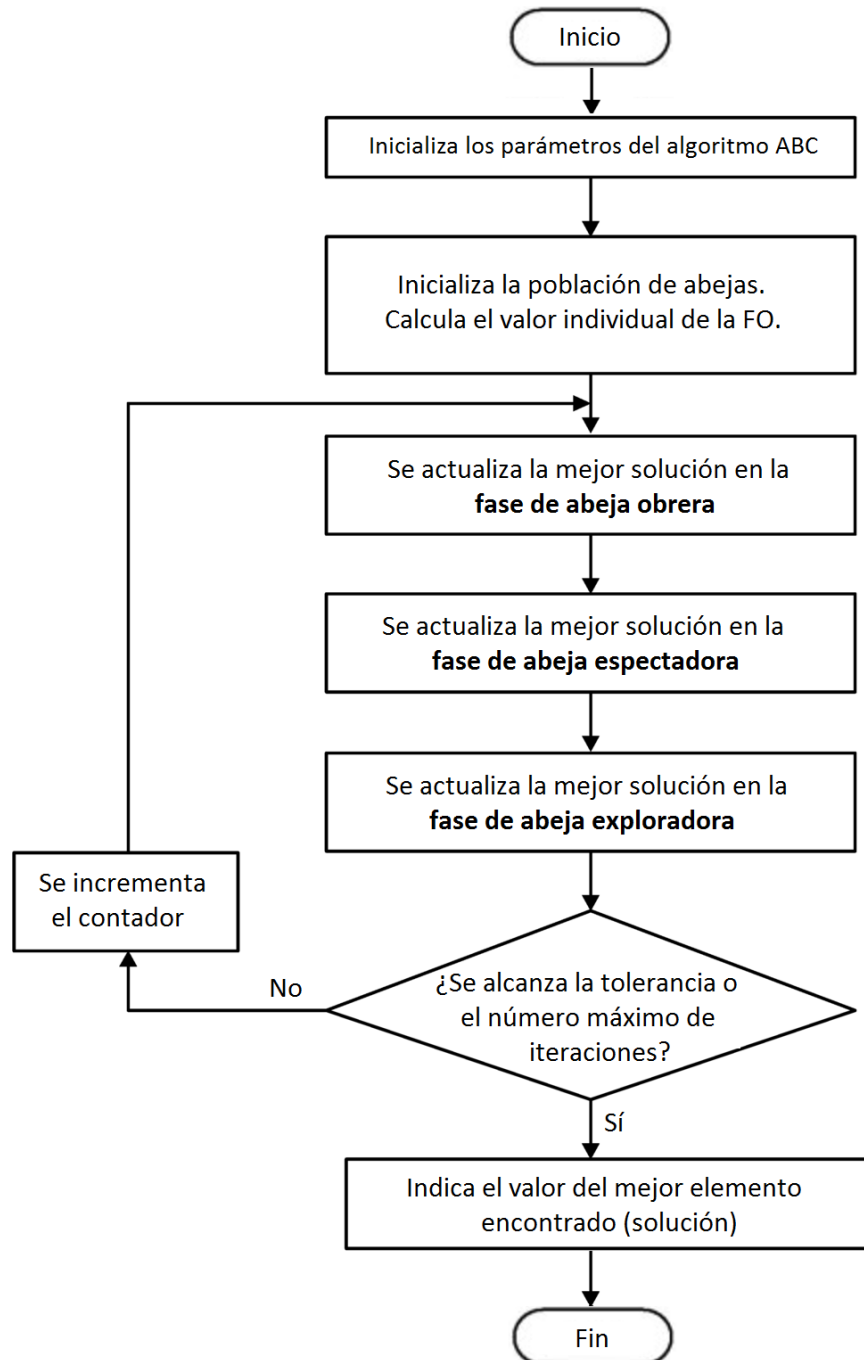


Figura 2.12: Algoritmo ABC.

similar al algoritmo de SA, sin embargo utiliza mutaciones a escala local y global para garantizar que no se estanca en un valor mínimo encontrado y lleva a cabo la selección de los mejores elementos usando elitismo, es decir, si bien es inspirado en un proceso natural independiente, utiliza estrategias de mutación y selección de los algoritmos genéticos y realiza la exploración del espacio de búsqueda de forma similar al algoritmo SA.

El algoritmo CS se muestra en la Figura 2.13

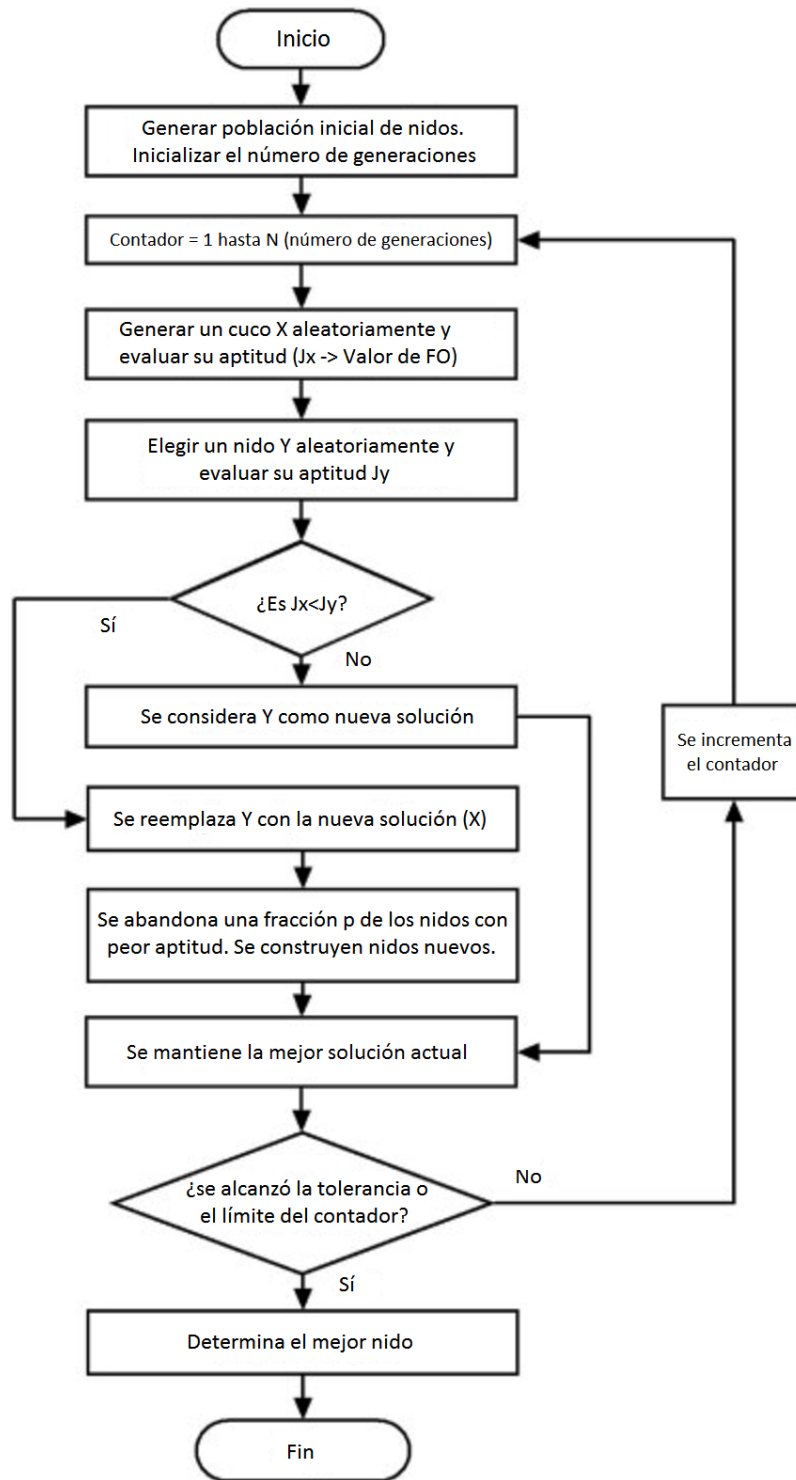


Figura 2.13: Algoritmo CS.

---

## Capítulo 3

# Síntesis óptima de mecanismos de cuatro barras (SOM4B)

En este capítulo se presenta el desarrollo de la función objetivo utilizada para la síntesis óptima de trayectoria de mecanismos planos de cuatro barras en forma general, para un número  $n$  de puntos de precisión.

También se aborda el uso de los algoritmos mencionados en el capítulo anterior para obtener el valor mínimo de la FO que representa el mecanismo de cuatro barras que pasa más cerca de los puntos dados.

### 3.1. Función objetivo de un mecanismo de cuatro barras

Para llevar a cabo la síntesis de un mecanismo de cuatro barras, es necesario definir previamente los elementos del mismo. En la Figura 3.1 se muestran los elementos necesarios para determinar la posición inicial de un mecanismo de cuatro barras, que serán determinados durante el proceso de síntesis óptima.

El punto C, es el punto del eslabón acoplador que idealmente pasaría por todos los puntos requeridos por el proceso de síntesis óptima durante la rotación del eslabón de entrada del mecanismo.

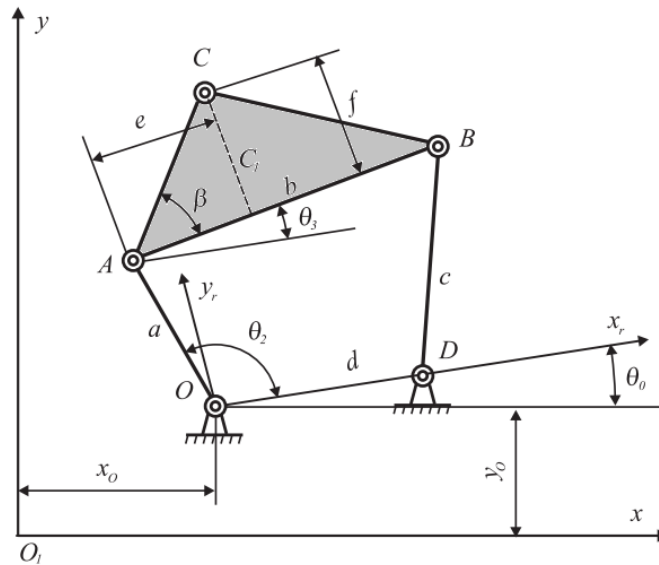


Figura 3.1: Parámetros geométricos de un M4B [47].

Los elementos presentes en la figura 3.1 son además de las longitudes de los eslabones y la distancia entre los apoyos de los eslabones de entrada y salida, los ángulos de inclinación de los eslabones de entrada y acoplador, en la primera posición, así como las coordenadas del apoyo del eslabón de entrada respecto al marco fijo de referencia. Esto puede observarse con más detalle en la siguiente lista:

- $a$  Longitud del eslabón de entrada.
- $b$  Longitud de la base del eslabón acoplador.
- $c$  Longitud del eslabón de salida.
- $d$  Distancia entre apoyos.
- $e, f$  Distancia del punto C sobre la base del acoplador y la normal a ésta.
- $x_0, y_0$  Coordenadas del apoyo del eslabón de entrada respecto al marco de referencia.
- $\theta_2$  Inclinación inicial del eslabón de entrada.
- $\theta_3$  Inclinación inicial del eslabón acoplador.
- $\theta_0$  Ángulo del vector que une el apoyo fijo del eslabón de entrada con el apoyo fijo del eslabón de salida.

La posición del punto  $C$  respecto al marco fijo de referencia está dada por:

$$\begin{bmatrix} C_x \\ C_y \end{bmatrix} = \begin{bmatrix} \cos \theta_0 & -\sin \theta_0 \\ \sin \theta_0 & \cos \theta_0 \end{bmatrix} \begin{bmatrix} C_{xr} \\ C_{yr} \end{bmatrix} + \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} \quad (3.1)$$

Donde  $C_{xr} = a \cos \theta_2 + e \cos \theta_3 - f \sin \theta_3$  y  $C_{yr} = a \sin \theta_2 + e \sin \theta_3 - f \cos \theta_3$ .

Las variables a optimizar son las longitudes de los eslabones ( $a, b, c, d$ ), los ángulos de los eslabones en la posición inicial ( $\theta_0, \theta_2, \theta_3$ ), así como los ángulos de giro de la manivela ( $\theta_2^2, \theta_2^3, \dots, \theta_2^n$ ) para alcanzar los puntos de precisión.

La función objetivo depende entonces de que la distancia del punto  $C$  del eslabón acoplador sea lo más pequeña posible de los puntos de precisión especificados en cada problema, por lo que la ecuación (3.2) representa el problema de optimización.

$$\text{mín} \left[ \sum_{i=1}^n \left[ (C_{xd}^i - P_x^i)^2 + (C_{yd}^i - P_y^i)^2 \right] \right] \quad (3.2)$$

Donde  $n$  es el número de puntos de precisión que se utilizarán para llevar a cabo el proceso de síntesis, mientras que  $C_{xd}^i$  y  $C_{yd}^i$  representan los puntos alcanzados por el punto  $C$  del acoplador durante la rotación del mecanismo y  $P_x^i$  y  $P_y^i$  son las coordenadas de los puntos de precisión con respecto al marco fijo de referencia.

Una de las ventajas de esta función objetivo es que se respeta en todo momento la hipótesis de cuerpo rígido, la cual establece que los componentes de un mecanismo no pueden estirarse o encogerse para todas las posiciones que este alcance al girar, es decir, la distancia entre dos puntos cualquiera del mecanismo debe mantenerse constante, aún cuando los elementos roten y se desplacen para alcanzar diferentes posiciones en un ciclo.

En este sentido, cuando el mecanismo no alcance una posición el valor de la función objetivo será un número complejo, indicando que para esa posición los elementos no respetan la hipótesis de cuerpo rígido, debiendo en la implementación de la síntesis incluir filtros que eliminen los resultados complejos de la síntesis.

A continuación se presentan los procedimientos de la implementación de algunos algoritmos representativos de optimización utilizados para sintetizar mecanismos planos de cuatro barras, con base en la función objetivo detallada en la ecuación (3.2).

Cabe destacar que se ajustaron los parámetros de los algoritmos utilizados de tal manera que el tiempo de evaluación fuera similar, es decir, para aquellos algoritmos que involucran un mayor número de cálculos o evaluaciones, se utilizó un número menor de iteraciones. Este ajuste en los parámetros se determinó empíricamente con base en las pruebas realizadas, a partir del registro de los tiempos obtenidos durante múltiples evaluaciones.

## **3.2. Minimización de la función objetivo del mecanismo de cuatro barras**

### **3.2.1. Uso de algoritmo SA**

El algoritmo de recocido simulado es uno de los algoritmos metaheurísticos más sencillos de implementar, siendo esta su principal ventaja, sin embargo, presenta ciertos inconvenientes, como la lentitud de convergencia, la importancia de la selección inicial de parámetros y el alto número de evaluaciones que debe realizar que lo dejan atrás como algoritmo seleccionado para problemas complejos.

Para la síntesis de mecanismos de cuatro barras, se utilizaron seis posiciones objetivo a alcanzar por el punto de interés del eslabón acoplador.

El número de evaluaciones realizadas por el algoritmo SA que se requieren para



que el error estructural sea cercano a cero es elevado, por lo que el tiempo de ejecución del programa desarrollado en el software Mathematica<sup>®</sup> de Wolfram, para alcanzar un valor bajo en la función objetivo puede llegar a ser de más de una hora en una computadora de escritorio estándar, cuando el uso de este mismo algoritmo para la optimización de funciones de una o dos variables independientes se lleva a cabo en segundos en las mismas condiciones del equipo de cómputo.

Si se compara este tiempo con el tiempo transcurrido para la evaluación de otros algoritmos de optimización, como los algoritmos genéticos, el recocido simulado es más lento si el criterio de paro es el valor alcanzado en la función objetivo, o más inexacto, para la misma cantidad de iteraciones realizadas.

Para la implementación del algoritmo de recocido simulado en la síntesis de mecanismos de cuatro barras, se eligió el problema planteado en [47] consistente en seis puntos en una trayectoria vertical, en un espacio de búsqueda de 120 unidades en el eje  $x$  y 80 en el  $y$ .

Los puntos de interés están localizados en las coordenadas (20,20), (20,25), (20,30), (20,35), (20,40) y (20,45). Los puntos de apoyo se desconocen y pueden estar localizados en cualquier lugar en el intervalo de búsqueda, así como los puntos extremos de los eslabones de entrada y salida, siendo ambos nodos del eslabón acoplador. El último punto del eslabón acoplador es el que de acuerdo con la síntesis realizada, alcanzará las posiciones determinadas por el problema.

En la Figura 3.2 puede observarse el mecanismo de cuatro barras obtenido al utilizar el algoritmo de recocido simulado para minimizar el error estructural en la trayectoria del mecanismo, la cual se desea que pase lo más cerca posible de los seis puntos de precisión colocados en línea vertical. También se observa la trayectoria que el punto del eslabón acoplador alcanza para los 360° de giro de la manivela de entrada, cumpliendo así la condición de Grashof.

El número de iteraciones está dado por la selección de la temperatura inicial y la

final, así como la tasa de enfriamiento. En este caso a la temperatura inicial se le asignó un valor de  $1000^\circ$  y la final de  $5^\circ$ , con una tasa de enfriamiento de 0.003, lo cual indica, de acuerdo al algoritmo, que en cada iteración disminuye el valor de la temperatura en 0.3 %, es decir, en la primera iteración  $T=1000^\circ$ , en la segunda  $T=997^\circ$  (disminuye 3 %) y así sucesivamente hasta que la temperatura alcanza los  $5^\circ$ . Para este proceso de búsqueda con los parámetros mencionados, la temperatura final se alcanza en la iteración 1770, que es donde el algoritmo se detiene.

Esto garantiza que la diferencia entre los valores encontrados durante las primeras iteraciones es mayor que la de los valores encontrados en las últimas iteraciones, lo que corresponde al comportamiento del algoritmo SA en el sentido de localizar valores durante las primeras iteraciones en todo el espacio de búsqueda y durante las últimas iteraciones concentrarse en los valores que se haya determinado que son los más cercanos a un mínimo, pudiendo en estos casos encontrar tanto soluciones cuyo valor de función objetivo es un mínimo global, aunque en gran parte de los casos, los mejores valores corresponden a mínimos locales de la función objetivo.

Para realizar un mayor número de iteraciones, puede incrementarse la temperatura inicial, disminuirse la temperatura final (sin llegar a los  $0^\circ$  ya que de este modo nunca se detendría el algoritmo) o bien disminuir el valor de la tasa de enfriamiento. Esto mejora los resultados obtenidos en la función objetivo, sin embargo la ejecución del programa se vuelve más lenta.

### 3.2.2. Uso de algoritmos genéticos

Los algoritmos genéticos han sido históricamente utilizados para optimización debido a que son fáciles de comprender e implementar, dada su similitud con el proceso de evolución utilizando selección natural. La síntesis óptima de mecanismos no es la excepción, ya que se ha utilizado con este fin desde 1994 [23].

Para la síntesis del mecanismo plano de cuatro barras empleando algoritmos genéticos, para este proyecto, se utilizó para fines comparativos el mismo problema presentado para el algoritmo de recocido simulado, es decir, los seis puntos de precisión

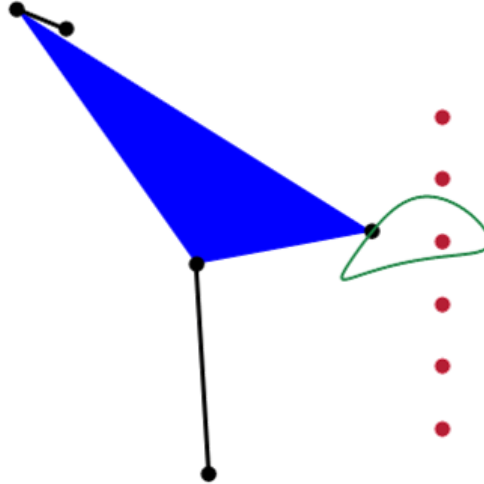


Figura 3.2: SM4B con el algoritmo de SA. Elaboración propia.

en una trayectoria vertical planteado en [47], con los mismos límites en el intervalo de búsqueda.

Asimismo, utilizando algoritmos genéticos para buscar la combinación de parámetros del mecanismo que generen la trayectoria que se aproxime más a la ideal, se utilizó como función objetivo la ecuación (3.2) para los seis puntos del recorrido. Los resultados del proceso de búsqueda son la localización de los apoyos o puntos fijos, y los puntos extremos del mecanismo en la primera posición, lo que permite construir el mecanismo y su posterior análisis cinemático para comprobar el cumplimiento de la condición de Grashof.

Respecto al algoritmo SA, los algoritmos genéticos muestran una mejor tendencia a converger hacia un valor mínimo de la función objetivo, debido a que el análisis de los elementos de cada generación no requiere grandes tiempos de cálculo y siempre conservan el conjunto de mejores elementos de cada generación (elitismo) por lo que a diferencia del algoritmo SA donde se tiene una probabilidad de aceptar una peor solución con el fin de aumentar la aleatoriedad, los algoritmos genéticos no desechan

una solución si no hay una mejor que la sustituya.

En el caso del mecanismo de cuatro barras con seis puntos de precisión, como se observa en la Figura 3.3, en las pruebas realizadas en este estudio, se logró obtener un mecanismo que pasa más cerca de los puntos de precisión deseados, describiendo un recorrido continuo y cumpliendo con la condición de Grashof, con tiempos de cálculo menores a los del algoritmo de recocido simulado.

Los parámetros usados en el algoritmo genético para la obtención del mecanismo de cuatro barras fueron los siguientes:

- Número de individuos: 50
- Elitismo: 4 elementos
- Probabilidad de cruce: 90 %
- Probabilidad de mutación: 30 %
- Número de generaciones: 1000

Estos parámetros pueden ser modificados con el fin de mejorar la solución obtenida, observándose que al aumentar el número de generaciones o de individuos se mejora el desempeño del algoritmo, con la limitante de que el tiempo de evaluación del algoritmo se incrementa.

### 3.2.3. Uso de algoritmo PSO

El algoritmo de optimización por enjambre de partículas desarrollado por Kennedy y Eberhart en 1995 [32] ha sido utilizado desde entonces, al igual que los algoritmos genéticos en múltiples aplicaciones en las que se requiere el uso de métodos heurísticos debido a que los métodos exactos no proveen de los elementos para poder resolverse, como es el caso de sistemas algebraicos inexistentes, que plantea la síntesis óptima de mecanismos.

El algoritmo PSO se ha utilizado para la síntesis óptima de mecanismos planos

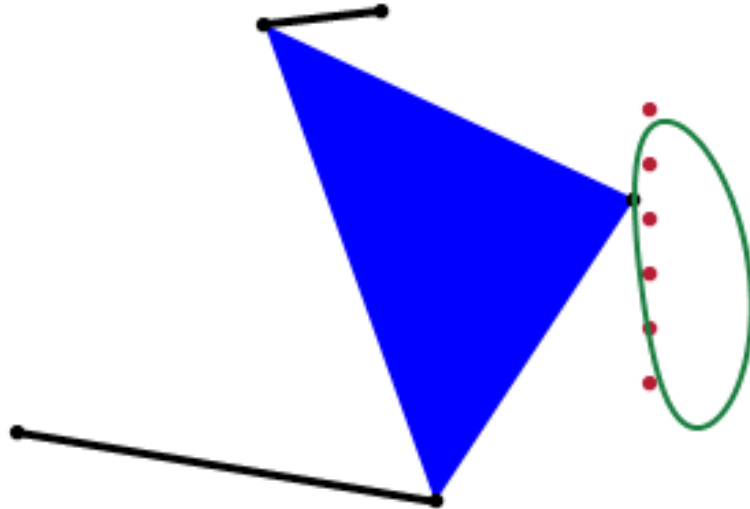


Figura 3.3: SM4B con algoritmos genéticos. Elaboración propia.

obteniendo buenos resultados [43]. Para la síntesis del mecanismo con el algoritmo PSO se utilizaron también las seis posiciones colocadas en una línea vertical, es decir, el mecanismo debe tener en su recorrido una parte idealmente recta, movimiento usual en los mecanismos de manivela-biela-corredera (MBC) y poco convencional en los mecanismos de cuatro barras, los cuales tienden a tener trayectorias curvas en el punto del acoplador.

El algoritmo PSO muestra una mejora en el desempeño, al tener una convergencia más rápida al llegar a menores valores de FO en pocas generaciones, lo que al compararlo con los algoritmos genéticos lo hace más conveniente, sin embargo, el algoritmo PSO tiende a no mejorar significativamente después de cierto número de generaciones, por lo que en ocasiones es posible que se lleven a cabo una gran cantidad de cálculos adicionales en los que no se mejora el valor de la función objetivo.

En la Figura 3.4 puede observarse que la trayectoria generada utilizando el algoritmo PSO es más cercana a los puntos de precisión deseados, para el tiempos de cálculo similares.

Los parámetros usados en el algoritmo PSO para la obtención del mecanismo de cuatro barras fueron los siguientes:

- Número de partículas: 300
- Número de iteraciones: 100

Estos parámetros también pueden ser modificados con el fin de mejorar la solución obtenida, observándose que al aumentar el número de iteraciones el desempeño del algoritmo no mejora significativamente, a diferencia del aumento en el número de partículas utilizado, ya que en este caso sí se observa una mejora en el desempeño del algoritmo, con la desventaja, al igual que en la síntesis óptima usando algoritmos genéticos, de que el tiempo de evaluación del algoritmo se incrementa.

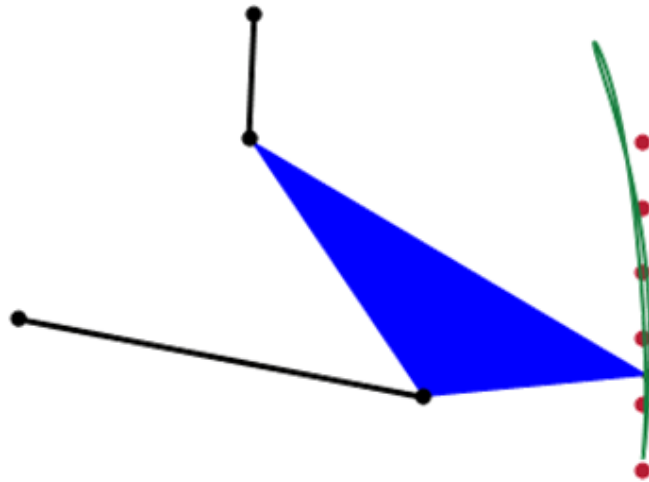


Figura 3.4: SM4B con algoritmo PSO. Elaboración propia.

### 3.2.4. Uso de algoritmo FA

El algoritmo de luciérnagas se utilizó también para sintetizar un mecanismo plano de cuatro barras. Este algoritmo tiende a recorrer el espacio de búsqueda atrayendo a las soluciones generadas hacia la más "brillante", representada por el vector solución

con valor más bajo de función objetivo.

El algoritmo FA tiene excelentes resultados de convergencia para funciones de variables reales y espacios de búsqueda reducidos. En el caso de los mecanismos de cuatro barras y la función objetivo (3.2), se tiene un gran número de variables, una probabilidad muy alta de que en algún punto del proceso se generen valores complejos de la función objetivo, además de un espacio de búsqueda muy amplio.

La combinación de estos tres factores hacen que el uso del algoritmo FA tenga dificultades en su aplicación a problemas de síntesis óptima de mecanismos planos, por lo que si bien el algoritmo tiene excelentes resultados cuando se cierra el espacio de búsqueda, es decir, los límites de las variables se reducen, esto sólo es posible cuando se conoce la solución aproximada del mecanismo que se busca, lo cual es poco usual en el proceso de síntesis de mecanismos, donde se parte del supuesto de que lo que se conoce del mecanismo es muy poco, en este caso los puntos aproximados de la trayectoria que se desea que alcance el acoplador.

Los parámetros usados en el algoritmo FA para la obtención del mecanismo de cuatro barras fueron los siguientes:

- Número de luciérnagas: 15
- Número de iteraciones: 600

En la Figura 3.5 puede observarse un mecanismo de cuatro barras sintetizado usando el algoritmo FA con el mismo espacio de búsqueda usado en la síntesis con el algoritmo de recocido simulado y el algoritmo PSO. En este caso, el resultado no es preciso, debido a que el algoritmo FA no obtiene resultados satisfactorios cuando los límites de las variables involucradas están lejos.

Por el contrario, en la Figura 3.6 se observa el resultado de la síntesis cuando los límites de las variables corresponden a valores más cercanos, en este caso obtenidos de la mejor solución en la síntesis con el algoritmo PSO. Es importante destacar que el algoritmo FA refina la solución obtenida mediante PSO, dado que como ya se

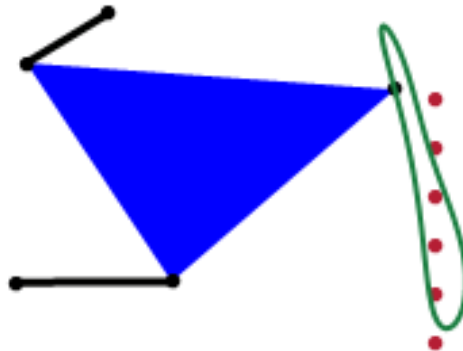


Figura 3.5: SM4B con algoritmo FA. Elaboración propia.

ha señalado, para espacios reducidos de búsqueda, el algoritmo FA tiene un mejor desempeño que en espacios grandes.

Cabe destacar que a diferencia del algoritmo PSO, el algoritmo FA encuentra me-

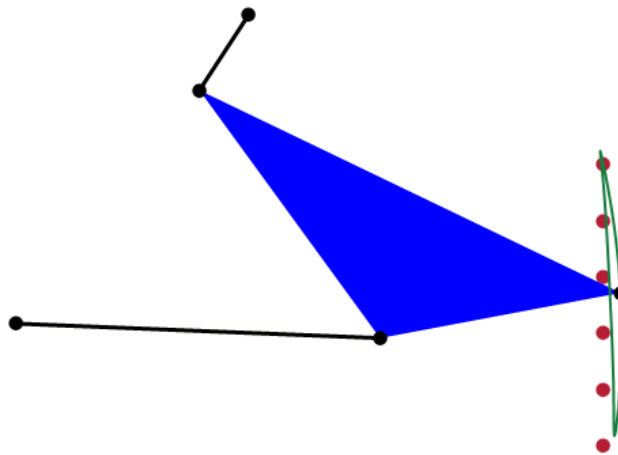


Figura 3.6: SM4B con algoritmo FA, con espacio de búsqueda reducido. Elaboración propia.

jores resultados al incrementar el número de iteraciones, con un menor número de luciérnagas, logrando obtener resultados similares a los obtenidos con el algoritmo PSO incrementando el número de iteraciones del algoritmo, preferentemente en intervalos de búsqueda reducidos.



El algoritmo FA podría utilizarse para refinar el resultado de la búsqueda del mecanismo obtenido por otros métodos de optimización, como el PSO o AG, generando un espacio de búsqueda reducido alrededor de la mejor solución encontrada para buscar una mejor solución en el área cercana.

### 3.2.5. Uso de algoritmo ABC

El algoritmo de colonia de abejas artificial se utilizó para realizar la síntesis de mecanismos de cuatro barras con el fin de detectar si era posible superar la limitación obtenida con el algoritmo FA, es decir, si el algoritmo ABC es capaz de encontrar un mecanismo partiendo de la misma función objetivo pero con un espacio de búsqueda amplio.

Debido a que consta de tres fases, la fase de abeja obrera, la fase de abeja espectadora y la fase de abeja exploradora, se busca varias veces durante la misma iteración mejorar el valor mínimo de función objetivo, añadiendo una expansión al espacio de búsqueda, lo que permite tener una mayor posibilidad de convergencia aún en estos intervalos mayores.

Para las pruebas realizadas al algoritmo ABC nuevamente se utilizó el problema expuesto en [47], de los seis puntos de precisión en una línea vertical, así como los intervalos de las variables involucradas utilizados en la SM4B con SA, AG y PSO.

Los parámetros utilizados en el algoritmo ABC fueron los siguientes:

- Número de abejas: 20
- Número de iteraciones: 1000

Puede observarse que los parámetros en relación al número de posibles soluciones (en este caso abejas) es similar a lo que se utilizó en el algoritmo FA, es decir, pocos elementos o individuos y un número de iteraciones grande.

En la Figura 3.7 puede observarse uno de los mecanismos sintetizados utilizando el algoritmo ABC. Los resultados son mejores que en el algoritmo FA dado que para intervalos grandes de las variables involucradas, se obtienen resultados similares a los que el algoritmo FA genera cuando el espacio de búsqueda es muy reducido.

Respecto al tiempo de cálculo, el algoritmo ABC tiene un desempeño similar al algoritmo PSO, por lo que también es una opción a evaluar para generar de forma rápida un conjunto de soluciones que serán analizadas posteriormente usando el algoritmo FA.

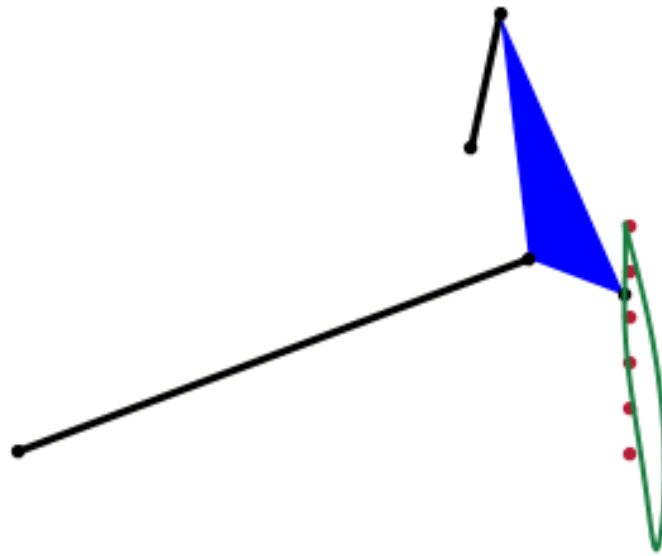


Figura 3.7: SM4B con algoritmo ABC. Elaboración propia.

### 3.2.6. Uso de algoritmo CS

El algoritmo CS también se utilizó para resolver el problema de síntesis de mecanismo de cuatro barras que debe recorrer una serie de seis puntos colocados verticalmente.

El algoritmo CS tiene la ventaja de poder realizar las iteraciones del ciclo de forma más rápida que algoritmos como los AG, PSO o FA, por lo que puede usarse un

número alto de iteraciones que son ejecutadas en el mismo tiempo que menos iteraciones en los algoritmos mencionados.

Los parámetros utilizados en el algoritmo CS fueron los siguientes:

- Número de nidos: 20
- Número de iteraciones: 5000

En la Figura 3.8 se observa el mecanismo obtenido utilizando el algoritmo CS para obtener la posición de todos los puntos de interés del mismo, así como la trayectoria del punto de interés del eslabón acoplador para los 360° de giro de la manivela.

El algoritmo CS presenta una convergencia rápida si se considera que el tiempo

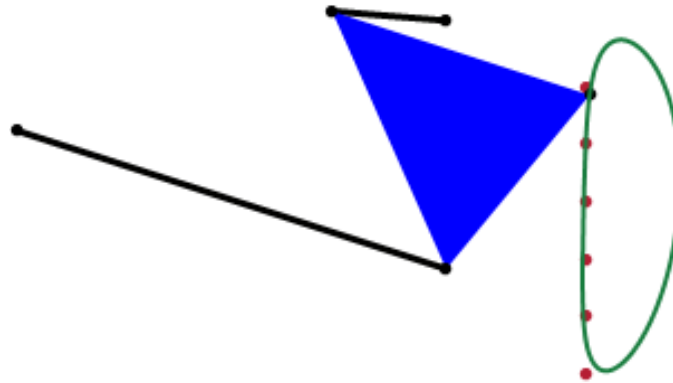


Figura 3.8: SM4B con algoritmo CS. Elaboración propia.

promedio en el que realiza 5000 iteraciones es equivalente al tiempo en el que se hacen 1000 iteraciones en los algoritmos genéticos o el algoritmo FA, por lo que en términos de tiempo de cálculo, resultan convenientes y de rápida convergencia, dado que invariablemente generan soluciones con valores de la función objetivo menores a 5 unidades.

En el capítulo siguiente se realizarán propuestas de posibles combinaciones de algoritmos, con objeto de construir la metodología de forma tal que se asegure que

siempre se obtengan valores cercanos a cero de la función objetivo en un tiempo razonable, en este caso, menor a una hora, independientemente de la cantidad de iteraciones realizadas.

---

## Capítulo 4

# Propuesta de metodología para SOM4B

En este capítulo se evaluará si es posible combinar los algoritmos presentados, con el fin de lograr eliminar las desventajas de cada uno y asegurar que independientemente de la cantidad de variables que incluye la función objetivo o el tamaño de los intervalos de búsqueda, es posible generar un mecanismo de cuatro barras que pase por los puntos dados con un error estructural mínimo y cumpliendo la condición de Grashof.

### 4.1. Conjunto de soluciones iniciales

Una de las problemáticas detectadas al utilizar los algoritmos evaluados al realizar la síntesis óptima de mecanismos planos fue que todos parten de un proceso aleatorio de inicialización de soluciones.

Dentro de esta inicialización, usualmente se sigue una distribución normal, asignando una mayor cantidad de valores al centro del espacio de búsqueda.

Esto puede ser un problema sobre todo para los algoritmos genéticos, ya que como se mencionó anteriormente, si en el conjunto inicial de soluciones no se encuentran valores cercanos al mínimo global, es difícil que el algoritmo lo alcance, incluso con

probabilidades altas de mutación.

En el caso del resto de los algoritmos el hecho de que la distribución del conjunto inicial de soluciones corresponda a una distribución normal no parece afectar el desempeño de los mismos.

No obstante, se propone dividir el espacio en partes iguales, de acuerdo a la cantidad de soluciones iniciales deseadas y repartir el intervalo de búsqueda para generar una red que cubra en intervalos iguales todo el espacio de búsqueda, asegurando que se cubre la totalidad del espacio en vez de centrarse en valores a la mitad del mismo.

Esto representa una ventaja sobre todo para aquellas funciones objetivo cuyo mínimo global está lejos del centro del espacio de búsqueda, ya que existen más posibilidades de encontrarlo en un menor número de iteraciones entre mejor repartidas en dicho espacio se encuentren las soluciones iniciales.

## 4.2. Algoritmos de rápida convergencia

Una vez que se logran repartir las soluciones iniciales en todo el espacio de búsqueda se desea utilizar un algoritmo cuya convergencia sea rápida, es decir, que incluso con soluciones iniciales que abarcan espacios amplios de búsqueda, sea capaz de encontrar en pocas iteraciones una solución cercana a cero.

Entre los algoritmos que mostraron un mejor desempeño relativo a la rapidez de convergencia se encuentran el algoritmo PSO, el algoritmo ABC y el algoritmo CS, por lo que son candidatos a iniciar la metodología, teniendo como ventaja que ya se les han colocado filtros para no considerar las soluciones complejas, de forma que la última solución encontrada será siempre un valor real.

### 4.3. Algoritmos de análisis detallado

Con elementos de la población cercanos entre sí, es posible refinar la búsqueda del mínimo global utilizando algoritmos que demostraron una alta capacidad para detallar soluciones, en espacios de búsqueda reducidos.

En este caso se propone el uso de los algoritmos genéticos y el algoritmo FA, los cuales tendrán por objetivo reducir en lo posible el valor de la función objetivo con base en el conjunto de elementos inicial producto de la ejecución previa de otro u otros algoritmos.

### 4.4. Propuesta de combinación de algoritmos

Debido a que las pruebas realizadas a lo largo de esta investigación arrojan que si bien un algoritmo de optimización es capaz de alcanzar un valor mínimo global si se permite que itere un número suficiente de veces, es posible que la mejora no sea sustancial después de realizar estas evaluaciones por días, semanas o incluso meses.

Es por ello que se propone que las evaluaciones se realicen en el menor tiempo posible, para obtener una solución al problema que, aún cuando no corresponda al valor mínimo global de la función objetivo, el valor alcanzado de error estructural sea evidentemente bajo.

Se considera un valor bajo, aquel que numéricamente corresponde a un valor de función objetivo menor al 1% de la longitud del intervalo de búsqueda en el eje coordenado donde esta longitud sea menor.

Este valor porcentual se plantea debido a que en caso de cumplirse dicha condición, también es posible observar a simple vista que la trayectoria del mecanismo pasa de manera muy cercana a todos los puntos planteados.

En este sentido se utilizarán tanto la inicialización de valores en todo el intervalo de búsqueda para todas las variables involucradas, así como un algoritmo de rápida

convergencia para filtrar el conjunto de soluciones en el menor tiempo posible.

Una vez encontrado un conjunto de soluciones con valores de la función objetivo menores al 1 % del intervalo de búsqueda, se utilizarán como los valores iniciales de un algoritmo de análisis detallado, para que a través de éste se mejoren las soluciones encontradas.

No se utiliza de inicio un algoritmo de análisis detallado debido a que es posible que se requiera más tiempo sólo para llegar a los valores iniciales dados por el primer algoritmo utilizado.

En la Tabla 4.1 se observan las combinaciones que se proponen evaluar, para verificar el desempeño del algoritmo individual con el algoritmo combinado.

Cabe mencionar que inicialmente se propone la combinación de un algoritmo de rápida convergencia con un algoritmo de análisis detallado, sin embargo, se evalúa también una combinación de dos algoritmos de rápida convergencia.

Rápida convergencia	Análisis detallado	Posibles combinaciones
PSO	AG	PSO-AG
	FA	PSO-FA
	ABC	PSO-ABC
ABC	AG	ABC-AG
	FA	ABC-FA
CS	AG	CS-AG
	FA	CS-FA

Tabla 4.1: Combinaciones de algoritmos de optimización a evaluar para la metodología de síntesis de mecanismos.



## 4.5. Propuesta de metodología para síntesis óptima de mecanismos

La metodología que se desea proponer y evaluar, requiere de la inicialización no aleatoria de los valores en el espacio de búsqueda, así como de combinación de al menos dos algoritmos de optimización.

Esto permite reducir o eliminar las desventajas encontradas en la evaluación de la aplicación de los algoritmos en la SOM4B.

Las etapas de la metodología se describen a continuación y se muestran en la Figura 4.1:

1. **Determinación del problema:** Este debe corresponder a la síntesis de trayectoria de un mecanismo de cuatro barras.
2. **Selección de los puntos de la trayectoria:** Se propone usar al menos seis puntos de precisión ubicados a lo largo de la trayectoria.
3. **Determinación del espacio de búsqueda:** Corresponde a la ubicación del mecanismo en un marco de referencia XY, donde se ubicarán los dos apoyos del mecanismo, así como los puntos correspondientes a las juntas cinemáticas del mismo, en la primera posición.
4. **Inicialización de las soluciones:** El espacio de búsqueda se divide en intervalos regulares para garantizar que no se tiene una tendencia inicial a buscar soluciones al centro del mismo, como resultaría en una inicialización aleatoria, basada en la distribución normal.
5. **Búsqueda preliminar:** Uso de un algoritmo de rápida convergencia para descartar rápidamente soluciones iniciales que no tienden a reducir el valor de la función objetivo. Al final de esta etapa, el conjunto de soluciones se encontrará cerca de un valor mínimo.
6. **Refinamiento de la solución:** Utilizando como valores iniciales los resultados obtenidos con un algoritmo de rápida convergencia, esta etapa busca en

un menor intervalo de búsqueda, para mejorar el valor de la función objetivo obtenido.

- 7. **Evaluación del mecanismo obtenido:** Una vez obtenidos los resultados, se realiza un análisis cinemático del mecanismo, para determinar la trayectoria que sigue el punto del acoplador.

En caso de no cumplir con la reducción del valor establecido de error estructural o la condición de Grashof, es posible repetir la búsqueda.

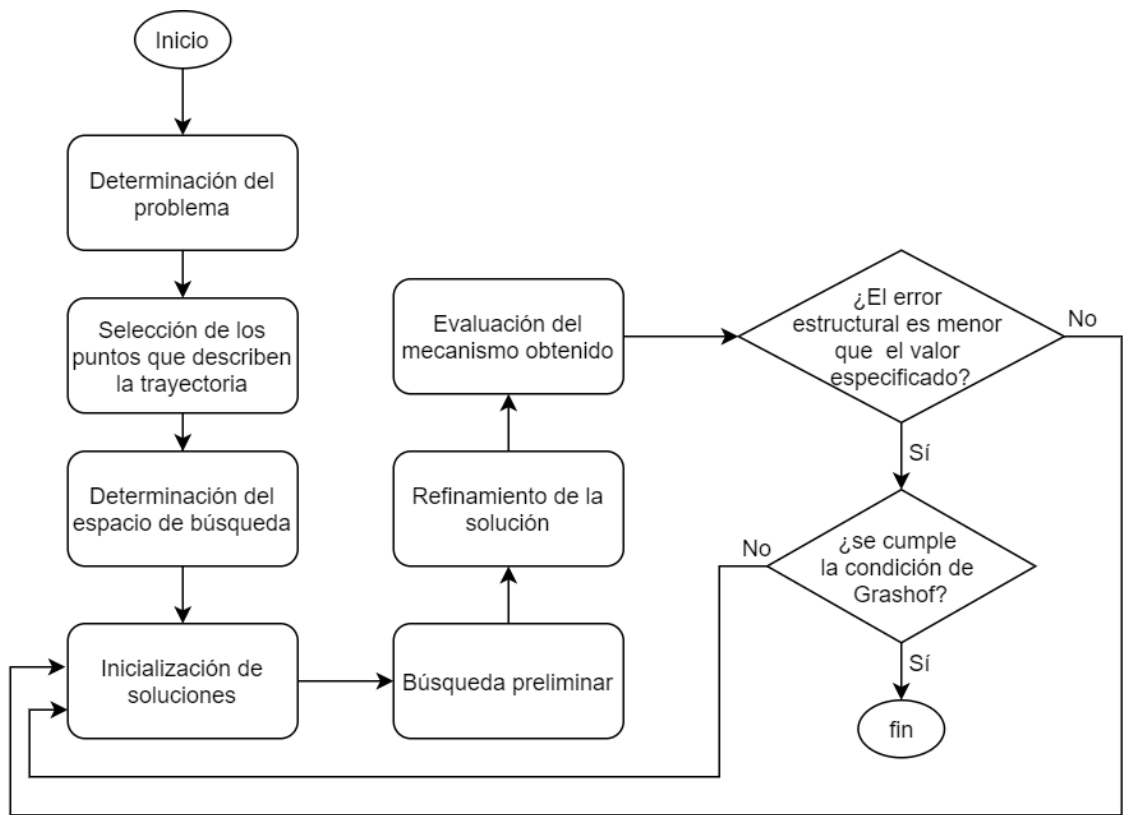


Figura 4.1: Etapas de la metodología propuesta. Elaboración propia.

## 4.6. Análisis de resultados

### 4.6.1. Combinación PSO-AG

Debido a que el algoritmo PSO funciona mejor con un número alto de partículas y pocas iteraciones, es uno de los algoritmos ideales para realizar un filtrado rápido de posibles soluciones.

Para el algoritmo de análisis detallado se contempló el algoritmo genético básico, con los valores para cada algoritmo detallados en la Tabla 4.2.

Algoritmo PSO	Algoritmo genético
Partículas: 4000 Iteraciones: 30	Individuos: 30 Tasa de cruce: 0.9 Generaciones: 300 Tasa de mutación: 0.25
Error estructural: $\leq 1$	Error estructural: $\leq 0.1$

Tabla 4.2: Parámetros de los algoritmos en la combinación PSO-AG.

Se contabilizaron 150 pruebas con duración promedio de 40 minutos para que la combinación de algoritmos obtuviera un mecanismo útil, con dos criterios de paro, ya sea el alcanzar un error estructural menor a una unidad para el caso del algoritmo PSO o menor a 0.1 unidades para el AG o bien, en ambos casos, alcanzar el número máximo de iteraciones establecido.

Los resultados de las pruebas contabilizadas muestran una mejora en el desempeño de ambos algoritmos, en relación al tiempo de ejecución y el valor obtenido de la FO.

### 4.6.2. Combinación PSO-FA

En el caso de la combinación entre algoritmos PSO-FA, se considera nuevamente al algoritmo PSO como el que tiene por objetivo filtrar el conjunto de soluciones para dar paso a la ejecución del algoritmo FA. Los valores de los parámetros para

cada algoritmo se detallan en la Tabla 4.3.

Se contabilizaron también 150 pruebas, mejorando la duración promedio de ejecución que se obtuvo de 29 minutos, en todos los casos esta combinación de algoritmos obtiene un mecanismo útil y también se consideraron los mismos criterios de paro que para la combinación PSO-AG, es decir, la obtención de un error estructural menor a una unidad para el caso del algoritmo PSO y menor a 0.1 unidades para el FA o bien, en ambos casos, alcanzar el número máximo de iteraciones establecido.

Los resultados de las pruebas contabilizadas indican también una mejora en el desempeño de ambos algoritmos, tanto para el tiempo de ejecución como para el valor obtenido de la FO.

Algoritmo PSO	Algoritmo FA
Partículas: 4000	Luciérnagas: 50
Iteraciones: 30	Iteraciones: 40
Error estructural: $\leq 1$	Error estructural: $\leq 0.1$

Tabla 4.3: Parámetros de los algoritmos en la combinación PSO-FA.

### 4.6.3. Combinación PSO-ABC

De manera adicional, se realizó una prueba para observar el desempeño de dos algoritmos considerados de rápida convergencia anidados, ejecutando primero el algoritmo PSO y posteriormente el ABC para la obtención de la síntesis de mecanismo de cuatro barras.

Se utilizó el algoritmo PSO, con 300 partículas y 50 iteraciones, para obtener un conjunto de soluciones que tienden a acercarse al mínimo global.

En la segunda etapa se utilizaron los primeros 20 elementos del conjunto final de soluciones del algoritmo PSO como conjunto inicial para el algoritmo ABC, debido a que los primeros elementos de la lista son aquellos que obtienen menores valores

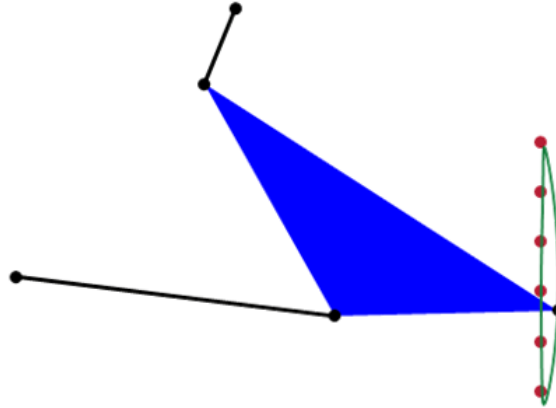


Figura 4.2: SM4B con una combinación de algoritmos PSO y ABC.

en la función objetivo y se ejecutaron 250 iteraciones del algoritmo.

Se encontró que el algoritmo ABC tiende a mejorar los valores obtenidos de la función objetivo, logrando reducirlos hasta un 65 %.

La combinación de ambos algoritmos permite la reducción de tiempo de cálculo, ya que las ejecuciones realizadas se llevaron a cabo en un tiempo menor que la ejecución individual de cada algoritmo.

En la Figura 4.2 se observa el mejor mecanismo obtenido con esta combinación de algoritmos, observándose una mejora evidente también en la reducción del valor de la función objetivo, al acercar aún más la trayectoria del punto de interés del eslabón acoplador al conjunto de puntos deseado.

Es de resaltar que el tiempo promedio de las 150 evaluaciones de esta combinación es de alrededor de 17 minutos, debido a que se reducen significativamente el número de elementos evaluados en el algoritmo PSO y el número de iteraciones en el algoritmo ABC, sin embargo también se observa la reducción del valor final de la FO y la trayectoria alcanzada por el mecanismo obtenido pasa de manera casi exacta

por los puntos planteados.

#### 4.6.4. Combinación ABC-AG

El algoritmo ABC fue uno de los más complejos de implementar para la síntesis de mecanismos de cuatro barras, debido a que los operadores están diseñados para trabajar con números reales y para la síntesis óptima deben considerarse valores complejos.

Sin embargo, una vez completado el proceso de implementación, se verificó que es uno de los algoritmos capaces de reducir rápidamente el valor de la FO para el conjunto de soluciones en proceso, por lo que se decidió usarlo como primera etapa para inicializar las soluciones para un algoritmo adicional.

El algoritmo ABC en conjunto con los AG muestra un desempeño similar a la combinación PSO-AG, en relación al tiempo de ejecución y mecanismos obtenidos.

Los valores para cada algoritmo se detallan en la Tabla 4.4. Se consideraron 150 pruebas cuya duración promedio fue de 47 minutos para que la combinación de algoritmos obtuviera un mecanismo útil, con los mismos criterios de paro al alcanzar el número máximo de iteraciones o bien valores de la FO menores a 1 y 0.1 unidades respectivamente.

Si bien por el tiempo promedio observado podría pensarse que no se obtiene una mejora en relación a las combinaciones hasta ahora evaluadas, la combinación ABC-AG obtiene un valor de error estructural promedio menor a los obtenidos en las combinaciones de algoritmos PSO-AG y PSO-FA.

#### 4.6.5. Combinación ABC-FA

La combinación entre el algoritmo ABC y FA permite observar el rápido decrecimiento en el valor de la FO durante la primera etapa de ejecución y la posterior

Algoritmo PSO	Algoritmo genético
Abejas: 20 Iteraciones: 250	Individuos: 30 Tasa de cruce: 0.9 Generaciones: 300 Tasa de mutación: 0.25
Error estructural: $\leq 1$	Error estructural: $\leq 0.1$

Tabla 4.4: Parámetros de los algoritmos en la combinación ABC-AG.

mejora en el valor final de la FO en un tiempo promedio de ejecución menor a las combinaciones hasta ahora evaluadas. Los valores para cada algoritmo se detallan en la Tabla 4.5.

Al igual que con las combinaciones anteriores, se contabilizaron 150 pruebas con duración promedio de 34 minutos para alcanzar como resultado un mecanismo útil, así mismo, se utilizaron los dos criterios de paro utilizados hasta ahora, es decir, el alcanzar un error estructural menor a una unidad para el caso del algoritmo ABC o menor a 0.1 unidades para el AG o bien, en ambos casos, alcanzar el número máximo de iteraciones establecido.

Algoritmo PSO	Algoritmo FA
Abejas: 20 Iteraciones: 250	Luciérnagas: 50 Iteraciones: 40
Error estructural: $\leq 1$	Error estructural: $\leq 0.1$

Tabla 4.5: Parámetros de los algoritmos en la combinación ABC-FA.

#### 4.6.6. Combinación CS-AG

El algoritmo CS tiene un tiempo de evaluación por iteración bajo, por lo que se realizaron pruebas de observación del comportamiento de los resultados al incrementar el número de iteraciones.

De este modo, se encontró que es posible realizar la ejecución de 50000 iteraciones en un tiempo muy similar al obtenido para el algoritmo PSO con sus 30 iteraciones y 4000 partículas.

Esto es debido a que el algoritmo PSO realiza evaluaciones a 4000 elementos por iteración, mientras que en el algoritmo CS estas evaluaciones se llevan a cabo en un número significativamente menor de elementos, por lo que es posible y recomendable realizar más iteraciones para mejorar la solución obtenida.

Para la etapa de análisis detallado se contempló el algoritmo genético, con los valores para cada algoritmo detallados en la Tabla 4.6. Se registraron 150 evaluaciones con duración promedio de 37 minutos hasta que se generaba un mecanismo útil, con similares criterios de paro.

Estos criterios de paro consisten en alcanzar un error estructural menor a una unidad para el caso del algoritmo CS o menor a 0.1 unidades para el AG o bien, en ambos casos, alcanzar el número máximo de iteraciones establecido, observándose ligeras mejoras tanto en relación a tiempos de ejecución como valores de la FO obtenidos.

Algoritmo PSO	Algoritmo genético
Nidos: 20	Individuos: 30
Iteraciones: 50000	Tasa de cruce: 0.9
	Generaciones: 300
	Tasa de mutación: 0.25
Error estructural: $\leq 1$	Error estructural: $\leq 0.1$

Tabla 4.6: Parámetros de los algoritmos en la combinación CS-AG.

#### 4.6.7. Combinación CS-FA

La última combinación evaluada de algoritmos de rápida convergencia para la primera etapa y algoritmos de análisis detallado para refinar las soluciones es la combinación de los algoritmos CS y FA.



Ambos algoritmos muestran un buen desempeño individual y el hecho de que la inicialización del algoritmo FA dependa de los resultados preliminares del algoritmo CS mejora el desempeño final comprobándose tanto por los valores de la FO objetivo obtenidos, como por la reducción en los tiempos de ejecución.

Para el número de registros analizados en todas las combinaciones anteriores, es decir, 150 ejecuciones de esta combinación, se obtuvo un tiempo promedio de 26 minutos, para los mismos criterios de paro y los parámetros de los algoritmos detallados en la Tabla 4.7.

Algoritmo PSO	Algoritmo FA
Nidos: 20	Luciérnagas: 50
Iteraciones: 50000	Iteraciones: 40
Error estructural: $\leq 1$	Error estructural: $\leq 0.1$

Tabla 4.7: Parámetros de los algoritmos en la combinación CS-FA.

#### 4.6.8. Comparación de las combinaciones evaluadas

El tiempo total de las pruebas realizadas fue de casi 32000 minutos, este tiempo considera solamente la realización de los cálculos por el equipo de cómputo. Se consideraron solamente las ejecuciones que arrojaron los mejores 150 valores de la función objetivo para cada combinación, independientemente del tiempo logrado para alcanzarlas. De estos, se obtuvo el tiempo promedio ya mencionado para cada combinación, el cual se observa en el en la Tabla 4.8:

Así mismo, se realizó el análisis del número de veces que se detuvo la ejecución de los algoritmos por alcanzar el valor buscado de la función objetivo, el cual se muestra en la Tabla 4.9, donde se observa que los algoritmos de rápida convergencia que llegaron a detenerse un mayor número de veces por ese motivo fueron el algoritmo PSO y el CS. El algoritmo ABC se detuvo una mayor cantidad de veces por alcanzar el número de iteraciones establecido.

Combinación utilizada	Duración máxima [ <i>min</i> ]	Duración mínima [ <i>min</i> ]	Tiempo Promedio	Mejor valor de error estructural obtenido
PSO-AG	101	19	40	0.01801
PSO-FA	42	11	29	0.00972
ABC-AG	89	31	47	0.00197
ABC-FA	64	17	34	0.00473
CS-AG	92	19	37	0.00132
CS-FA	70	18	26	0.00093
PSO-ABC	32	8	17	0.00004

Tabla 4.8: Comparación de tiempo de ejecución y valor de error estructural obtenido

En el caso de los algoritmos utilizados para refinar la solución, partiendo del último resultado del algoritmo de rápida convergencia, tienden a comportarse de manera similar, independientemente del algoritmo anterior, con alrededor del 60% de las veces alcanzando un valor menor al establecido en el criterio de paro.

## 4.7. Postprocesamiento de resultados

### 4.7.1. Análisis cinemático de mecanismos obtenidos

A todos los mecanismos obtenidos se les realizó un análisis cinemático para la verificación de la trayectoria de manera visual. El análisis de posición del mecanismo permite detectar la trayectoria del punto del eslabón acoplador que sigue la trayectoria deseada y observar si existen puntos de agarrotamiento.

### 4.7.2. Cumplimiento de la condición de Grashof

Se verificó que todas las soluciones obtenidas cumplieran con la condición de Grashof, es decir, el eslabón considerado como manivela gira los 360°. Se descartaron las soluciones obtenidas que presentaban puntos de agarrotamiento, debido a que en la práctica, es posible que no puedan seguir girando por sí solos de manera fluida.

Combinación utilizada	Número máximo de iteraciones	Valor menor al error estructural establecido	Total
PSO-AG	PSO: 14 AG: 62	PSO: 136 AG: 88	150
PSO-FA	PSO: 38 FA: 74	PSO: 112 FA: 76	150
ABC-AG	ABC: 104 AG: 56	ABC: 46 AG: 94	150
ABC-FA	ABC: 61 FA: 73	ABC: 89 FA: 77	150
CS-AG	CS: 18 AG: 68	CS: 132 AG: 82	150
CS-FA	CS: 29 FA: 57	CS: 121 FA: 93	150
PSO-ABC	PSO: 32 ABC: 49	PSO: 118 ABC: 101	150

Tabla 4.9: Causas de detención de la ejecución del algoritmo utilizado para la SOM4B.

---

# Capítulo 5

## Conclusiones

- Se ha propuesto una metodología consistente en anidar distintos algoritmos de optimización, utilizando primero uno para generar de forma rápida una disminución en el valor de la función objetivo y otro para refinar la solución de forma local.
- Se ajustaron los parámetros de los algoritmos utilizados de forma que el tiempo de cálculo para el total de iteraciones fuera similar para todos los algoritmos evaluados. Dado que algunos de ellos requieren de un mayor número de cálculos, se realizaron evaluaciones con un menor número de elementos, para homogeneizar el tiempo total de evaluación.
- La función objetivo utilizada puede generar valores complejos, que debieron filtrarse para garantizar que todas las soluciones encontradas eran valores reales, para un correcto funcionamiento de los algoritmos de optimización seleccionados.
- El algoritmo de recocido simulado no tiene un buen desempeño en la síntesis de mecanismos de cuatro barras, por lo que no se incluyó en la metodología de síntesis de mecanismos.
- Los algoritmos genéticos pueden aplicarse a la síntesis óptima de mecanismos de cuatro barras pero están limitados por los valores iniciales de la población. Si en estos valores existen datos cercanos al mínimo global, seguramente lo

encontrarán.

- Por el contrario, si los valores iniciales de la población son lejanos al mínimo global, incluso con una probabilidad de mutación alta, es difícil que se logre encontrar dicho valor usando algoritmos genéticos.
- El algoritmo PSO presenta una rápida convergencia y resultados alentadores en su aplicación a la síntesis de mecanismos de cuatro barras.
- El uso de más iteraciones en el algoritmo PSO no mejora significativamente el valor mínimo encontrado, aumentando los tiempos de cálculo de la computadora utilizada.
- El algoritmo FA genera mejores resultados en su aplicación a la síntesis de mecanismos de cuatro barras si el espacio de búsqueda es reducido.
- El algoritmo ABC tiene un desempeño similar al algoritmo PSO, dado que obtiene buenos resultados al converger a valores de la función objetivo cercanos a cero, incluso cuando el espacio de búsqueda es amplio.
- El algoritmo de búsqueda cuco genera resultados alentadores ya que su desempeño en la síntesis de mecanismos de cuatro barras también tiende a converger de forma rápida.
- Los criterios de paro para los algoritmos fueron tanto el número de iteraciones (cuando se deseaba que el tiempo no sobrepasara cierto valor) como el valor mínimo de la función objetivo esperado.
- Se evaluaron siete combinaciones de algoritmos utilizando como criterio para la combinación la rapidez de convergencia observada durante la implementación de los mismos, así como la capacidad de seguir encontrando y mejorando las soluciones cuando los elementos evaluados son valores cercanos.
- Todas las combinaciones tuvieron resultados mejores que la ejecución individual de los algoritmos, sin embargo no es fácil cuantificar esta mejora de forma exacta debido a que durante la ejecución de todos los algoritmos se utilizan operadores generados aleatoriamente, por lo que únicamente se evaluó

el desempeño a través del tiempo de ejecución y el valor de error estructural obtenido.

- La combinación de algoritmos PSO y ABC demostró ser capaz de obtener resultados superiores a todas las demás evaluadas, tanto en el tiempo de ejecución, como en el valor alcanzado de la función objetivo.
- Al realizar el análisis cinemático de todos los mecanismos obtenidos se comprueba que cumplen con la condición de Grashof y que no presentan puntos muertos.
- Es posible obtener un mecanismo de cuatro barras útil a través de la síntesis óptima de trayectoria utilizando puntos que pertenezcan a la trayectoria ideal para determinar el mecanismo en su posición inicial, utilizando una combinación de métodos heurísticos anidados, lo que garantiza el error estructural mínimo.

---

## Referencias

- [1] L. Birglen and C. Gosselin, “Optimal design of 2-phalanx underactuated fingers,” *Proceedings of International Conference on Intelligent Manipulation and Grasping*, pp. 110–116, 2004.
- [2] N. Nava Rodriguez, G. Carbone, and M. Ceccarelli, “Optimal design of driving mechanism in a 1-DOF anthropomorphic finger,” *Mechanism and Machine Theory*, vol. 41, no. 8, pp. 897–911, 2006.
- [3] E. Ngale Haulin, A. Lakis, and R. Vinet, “Optimal synthesis of a planar four-link mechanism used in a hand prosthesis,” *Mechanism and Machine Theory*, vol. 36, no. 11-12, pp. 1203–1214., 2001.
- [4] S. Yao, M. Ceccarelli, G. Carbone, Q. Zhan, and Z. Lu, “Analysis and optimal design of an underactuated finger mechanism for LARM hand,” *Frontiers of Mechanical Engineering*, vol. 6, no. 3, pp. 332–343, 2011.
- [5] M. Husty, M. Pfulner, H. Schröcker, and K. Brunntahaler, “Algebraic methods in mechanism analysis and synthesis,” *Robotica*, vol. 25, no. 6, pp. 661–675, 2007.
- [6] K. Jong-Won and S. TaeWon, “A new numerical method for the design of a four-bar linkage mechanism,” *The 14th IFToMM World Congress*, 2015. DOI Number: 10.6567/IFToMM.14TH.WC.OS3.002.
- [7] W. Zhang, S. Xudong, and W. Liwen, “Lamarckism and mechanism synthesis: approaching constrained optimization with ideas from biology.” 2011.

- 
- [8] S. Ebrahimi and P. Payvandy, “Efficient constrained synthesis of path generating four-bar mechanisms based on the heuristic optimization algorithms.,” *Mechanisms and Machine Theory*, vol. 85, pp. 189–204, 2015.
- [9] J. Nocedal and S. Wright, *Numerical Optimization*. Springer, 2006.
- [10] R. Norton, *Diseño de Maquinaria: Síntesis y análisis de máquinas y mecanismos*. Mc Graw Hill, 2005.
- [11] M. Sandeep, M. Sonpimple, and M. Giripunje, “Development of traditional optimization approach for optimal synthesis of path generator linkage,” *International Journal of Engineering and Technical Research*, vol. 1, no. 9, 2013.
- [12] A. Erdman and G. Sandor, *Diseño de Mecanismos: Análisis y Síntesis*. Prentice Hall, 1998.
- [13] S. Kirkpatrick, C. Gellat, and M. Vecchi, “Optimization by simulated annealing,” *Science*, vol. 220, no. 4589, pp. 671–680, 1983.
- [14] D. Bertsimas and J. Tsitsiklis, “Simulated annealing,” *Statistical Science*, vol. 8, no. 1, pp. 10–15, 1993.
- [15] S. Slesongsom and S. Bureerat, “Optimal synthesis of four-bar linkage path generation through evolutionary computation,” *Transactions of the TSME*, vol. 3, no. 2, pp. 46–53, 2015.
- [16] G. Jones, “Genetic and evolutionary algorithms,” *Encyclopedia of Computational Chemistry*, vol. 2, pp. 1127–1136, 1998.
- [17] J. Holland, *Adaptation in natural and artificial systems*. University of Michigan Press, 1975.
- [18] M. Mitchell, *An Introduction to Genetic Algorithms*. MIT Press, 1999.
- [19] G. P. Roston and R. H. Sturges, “Genetic algorithm synthesis of four-bar mechanisms,” *Artificial Intelligence for Engineering, Design, Analysis and Manufacturing*, vol. 10, pp. 371–390, 11 1996.



- 
- [20] J. Cabrera, A. Simon, and M. Prado, “Optimal synthesis of mechanisms with generic algorithms,” *Mechanisms and Machine Theory*, vol. 37, pp. 1165–1177, 2002.
- [21] A. Cabrera, F. Nadal, J. Muñoz, and A. Simon, “Multiobjective constrained optimal synthesis of planar mechanisms using a new evolutionary algorithm,” *Mechanism and Machine Theory*, vol. 42, pp. 791–806, 2007.
- [22] J. Cabrera, A. Ortiz, F. Nadal, and J. Castillo, “An evolutionary algorithm for path synthesis of mechanisms,” *Mechanisms and Machine Theory*, vol. 46, pp. 127–141, 2011.
- [23] E. W. Fang, “Simultaneous type and dimensional synthesis of mechanisms by genetic algorithms,” *Mechanism Synthesis and Analysis*, vol. 70, 1994.
- [24] A. Kunjur and S. Krishnamurty, “GA in mechanical synthesis,” *Applied mechanisms and robotics*, vol. 4, pp. 18–24, 1997.
- [25] M. Lu-Zhong, L. An-Xin, S. Hui-Ping, and Y. Ting-Li, “A method for structure synthesis of reconfigurable mechanisms based on genetic optimization algorithm,” in *International Conference on Reconfigurable Mechanisms and Robots*, 2009.
- [26] L. Wen-Yi, “A GA–DE hybrid evolutionary algorithm for path synthesis of four-bar linkage,” *Mechanism and Machine Theory*, vol. 45, pp. 1096–1117, 2010.
- [27] E. Lugo-González, Merchán-Cruz, E. A., and L. H. Hernández-Gómez, “Synthesis optimization of planar mechanisms,” *Applied Mechanics and Materials*, vol. 15, pp. 55–60, 2009.
- [28] E. Merchán-Cruz, E. Lugo-González, L. Hernández-Gómez, R. Rodríguez-Cañizo, J. Sandoval-Pineda, A. Soria-Moya, J. Ramírez-Gordillo, and G. Urriolagoitia-Sosa, “GA-based kinematical synthesis of a watt-type mechanism for prosthetic knee applications,” *Proceedings of the fifth IASTED International Conference Computational Intelligence*, vol. 27, 2016.

- 
- [29] L. L.C. Dülger, H. Erdoğan, and M. Kütük, “Matlab’s ga and optimization toolbox: A fourbar mechanism application,” *International Journal of Intelligent Systems and Applications in Engineering*, vol. 2, no. 1, pp. 10–15, 2014.
- [30] R. Starosta, “Application of genetic algorithm and fourier coefficients (GA-FC) in mechanism synthesis.,” *Journal of theoretical and applied mechanics*, vol. 46, no. 2, pp. 395–411, 2008.
- [31] L. Wen-Yi and H. Kuo-Mo, “More effective optimum synthesis of path generating four-bar mechanisms,” *Journal of Multidisciplinary Engineering Science and Technology*, vol. 2, no. 5, pp. 905–913, 2015.
- [32] J. Kennedy and R. Eberhart, “Particle swarm optimization,” pp. 1942–1948, 1995.
- [33] M. Laribi, A. Mlika, L. Romdhane, and S. Zeghloul, “A combined genetic algorithm–fuzzy logic method (GA–FL) in mechanisms synthesis,” *Mechanism and Machine Theory*, vol. 39, p. 717–735, 2004.
- [34] X. S. Yang, *Harmony Search as a Metaheuristic Algorithm*, vol. 191 of *Studies in Computational Intelligence*. Springer Berlin, 2009.
- [35] A. Alhajj and J. Srinivas, “An optimization approach for path synthesis of four-bar grashof mechanisms,” in *15th National Conference on Machines and Mechanisms*, 2011.
- [36] M. Monish P. Wasnik, M. K. Sonpimple, and U. S. K., “Optimal synthesis of a path generator linkage using non conventional approach,” *IOSR Journal of Mechanical and Civil Engineering (IOSR-JMCE)*, 2014.
- [37] A. Sánchez-Márquez, E. Vega-Alvarado, E. Portilla-Flores, and E. Mezura-Montes, “Synthesis of a planar four-bar mechanism for position control using harmony search algorithm,” *11th International Conference On Electrical Engineering, Computing Science and Automatic Control (CCE)*, 2014.

- 
- [38] P. Zsombor-Murray, C. Chen, and M. Hayes, "Direct kinematic mapping for general planar parallel manipulators," in *In Proceedings of the CSME Forum*, 2002.
- [39] M. J. D. Hayes, T. Luu, and X.-W. Chang, "Kinematic mapping application to approximate type and dimension synthesis of planar mechanisms," *9th Advances in Robotic Kinematics*, pp. 41–48, 2004. Dordrecht, the Netherlands, pp. 41-48, 2004
- [40] M. J. D. Hayes and S. R. Rucu, "Quadric surface fitting applications to approximate dimensional synthesis," *13th World Congress in Mechanism and Machine Science*, pp. 19–25, 2011.
- [41] S. Das, A. Abraham, and A. Kona, "Particle swarm optimization and differential evolution algorithms: Technical analysis, applications and hybridization perspectives," *Studies in Computational Intelligence*, vol. 116, pp. 1–38, 2008.
- [42] K. Zhang and S. Wang, "Kinematics design of a planar parallel controllable mechanism based on particle swarm optimization algorithm.," *Journal of computers*, vol. 6, no. 6, 2011.
- [43] S. Alinia, H. Daniali, M. Ghadimi, and H. Kaliji, "Optimal synthesis of planar mechanisms with PSO algorithms," *World Applied Sciences*, vol. 18, no. 2, pp. 268–273, 2012. 10.5829/idosi.wasj.2012.18.02.670.
- [44] C. S. for Mechanical Engineering Forum, ed., *Grashof Mechanism Synthesis Using Multi-Objective Parallel Asynchronous Particle Swarm Optimization*, 2010.
- [45] M. Guzman and C. Galeano, "Mechanism synthesis using a multi-objective optimization bio-inspired technique.," in *21st Brazilian Congress of Mechanical Engineering*, 2011.
- [46] A. H. Gandomi and A. H. Alavi, "Krill herd: A new bio-inspired optimization algorithm," *Commun Nonlinear Sci Numer Simulat*, vol. 17, p. 4831–4845, 2012.

- 
- [47] R. R. Bulatovic, G. Miodragovic, and M. S. Boskovic, “Modified krill herd (MKH) algorithm and its application in dimensional synthesis of a four-bar linkage,” *Mechanism and Machine Theory*, vol. 95, pp. 1–21, 2016.
- [48] M. Dixit, N. Upadhyay, and S. Silakari, “An exhaustive survey on nature inspired optimization algorithms,” *International Journal of Software Engineering and Its Applications*, vol. 9, no. 4, pp. 91–104, 2015.
- [49] X. S. Yang, “Nature-inspired metaheuristic algorithms: Success and new challenges,” *Journal of Computer Science and Technology*, vol. 1, 2012.
- [50] X. S. Yang, “Firefly algorithms for multimodal optimization,” *Int. J. Swarm Intelligence*, vol. 1, no. 1, pp. 36–50, 2013.
- [51] X. S. Yang, *Nature-Inspired Optimization Algorithms*. Elsevier Insights, 2014.
- [52] D. Karaboga and B. Basturk, “Artificial bee colony (ABC) optimization algorithm for solving constrained optimization problems,” *Foundations of Fuzzy Logic and Soft Computing*, pp. 789–798, 2005.
- [53] S. M. Lewis and C. C.K., “Flash signal evolution, mate choice and predation in fireflies,” *Annual Review of Entomology*, vol. 53, no. 2, pp. 293–321, 2008.