

Mecanisynt: Software Libre para la Generación de Trayectorias en Mecanismos de Cuatro Barras Usando Algoritmos Genéticos

Mecanisynt: Free Software for the trajectory's generation in four-bar mechanism using Genetic Algorithms

Alexander D. López¹, Alfonso Rodríguez Peña², John A. Rodríguez F¹, Lisandro Vargas Henríquez³, Rafael Ramírez Restrepo⁴

¹Ingeniero Mecánico. Egresado Facultad de Ingeniería. Universidad del Atlántico. Grupo de Investigación CONFORMAT, Puerto Colombia-Colombia.

²MSc. Docente, Facultad de Ingeniería. Universidad del Atlántico. Grupo de Investigación CONFORMAT, Puerto Colombia-Colombia.

³MSc. Docente, Facultad de Ingeniería. Universidad del Atlántico. Grupo de Investigación IMTEF, Puerto Colombia-Colombia.

⁴MSc. Docente, Facultad de Ingeniería. Universidad del Atlántico. Grupo de Investigación DIMER, Puerto Colombia-Colombia.

e-mail: alfonsorodriguez1@mail.uniatlantico.edu.co

Cite this article as: A.D. López-de los Reyes, A. Rodríguez-Peña, J.A. Rodríguez-Field, L. Vargas-Henriquez, R. Ramírez-Restrepo "Mecanisynt: Software Libre para la Generación de Trayectorias en Mecanismos de Cuatro Barras Usando Algoritmos Genéticos", *Prospectiva*, Vol. 21 N° 2 2023

Recibido: 05/05/2022 / Aceptado: 06/06/2023

<http://doi.org/10.15665/rp.v21i2.2962>

RESUMEN

Existen métodos clásicos para determinar la síntesis de mecanismos basados en procedimientos gráficos y analíticos en los que su principal limitante es que el número de puntos de precisión que pueden tomarse en cuenta para definir la trayectoria deseada es restringido. En este artículo se describe el desarrollo de un software que realiza la síntesis de generación de trayectorias y la simulación cinemática de mecanismos planos de cuatro barras para múltiples puntos de precisión, utilizando como método de optimización los Algoritmos Genéticos. Además, se generan gráficas de posición, velocidad y aceleración para hacer más sencilla la interpretación de los datos y calcular la cercanía entre los puntos por los cuales pasará el mecanismo obtenido. Inicialmente, se elaboró un diagrama de flujo en lenguaje Java, donde se define la lógica algorítmica de su funcionamiento. Luego, se desarrolló una interfaz gráfica flexible, intuitiva y de fácil manejo, que se validó presentándola a posibles usuarios finales. Los resultados obtenidos se validaron comparándolos con diferentes mecanismos propuestos en estudios anteriores, demostrando que el software permitirá a los usuarios realizar la síntesis de trayectorias de forma más fácil y rápida que los métodos analíticos existentes.

Palabras clave: Algoritmo genético, síntesis de mecanismos, simulación cinemática, trayectoria, mecanismos planos.

ABSTRACT

There are classical methods to determine the synthesis of mechanisms based on graphical and analytical procedures in which the main limitation is that the number of precision points that can be taken into account to define the desired trajectory is restricted. This paper describes the development of a software that performs trajectory generation synthesis and kinematic simulation of four-bar flat mechanisms for multiple precision points, using Genetic Algorithms as an optimization method. In addition, position, velocity and acceleration graphs are generated to simplify the interpretation of the data and to calculate the proximity between the points through which the obtained mechanism will pass. Initially, a flow diagram was developed in Java language, where the algorithmic logic of its operation is defined. Then, a flexible, intuitive, and user-friendly graphical interface was developed and validated by presenting it to potential end users. The results obtained were validated by comparing them with different mechanisms proposed in previous studies, demonstrating that the software will allow users to perform the synthesis of trajectories easier and faster than existing analytical methods.

Keywords: Genetic algorithm, mechanism synthesis, kinematic simulation, trajectory, flat mechanisms.

1. INTRODUCCIÓN

Los mecanismos de cuatro barras es uno de los sistemas de eslabonamiento más utilizado en el diseño de máquinas, dada su gran flexibilidad en la generación de trayectorias controladas de un grado de libertad [1]. Los estudios se han centrado en tres grandes ramas, la síntesis, el análisis cinemático y el análisis cinético.

La síntesis es el proceso de diseño de una máquina o sistema mecánico que tiene como objetivo seleccionar el mecanismo óptimo que satisfaga los requerimientos deseados. Existen diversos tipos de síntesis; en este trabajo se aborda la síntesis dimensional de un mecanismo para determinar las longitudes, ángulos y coordenadas que son necesarios para la generación de trayectorias [1]. Existen procedimientos gráficos y analíticos clásicos para determinar la generación de trayectorias de un mecanismo [2][3][4][5][6]. El uso de los métodos gráficos es una solución rápida, pero sacrificando exactitud, y los métodos analíticos se basan en expresiones algebraicas que pueden resultar extensas y difícil para sistemas mecánicos complejos, además, limitan el número de puntos de precisión, así como el número de posiciones del acoplador como generador de movimiento, dando un único punto óptimo de diseño, que pueden llegar a desviarse de la trayectoria deseada [7]. Estas deficiencias de los métodos clásicos son superadas por los métodos modernos de optimización de mecanismo de cuatro barras para la síntesis para múltiples puntos de precisión, como la Optimización no Lineal [8], Algoritmos Genéticos (AG) [9], Redes Neuronales [10], Métodos de Monte Carlo [11], entre otras.

Los algoritmos genéticos (AG) arrojan buenos resultados para problemas de síntesis y generación de trayectoria. Por lo tanto, se han convertido en una opción para resolver problemas de optimización en la síntesis de mecanismos ya que buscan a partir de una población ofrecer un conjunto de soluciones buenas al diseñador, minimizando los errores de posición y orientación en mecanismos articulados, además, de disminuir la diferencia en la distancia entre la trayectoria generada y la deseada en mecanismos de cadena cerrada [9].

Debido a la demanda en la industria por obtener procesos cada vez más automatizados, ha habido un gran interés de investigación en el campo de la optimización de mecanismos de cuatro barras. Kunjur et al. [12], utilizaron Algoritmos Genéticos para realizar la optimización de mecanismos planos de cuatro barras, obteniendo mejores resultados que los obtenidos por los métodos analíticos. Acharya et al. En el 2009 [13], implementaron diferentes algoritmos evolutivos, como los Algoritmos Genéticos, Evolución Diferencial y Optimización por Cúmulos de Partículas (PSO), para la optimización en la generación de síntesis del mecanismo de cuatro barras. El objetivo fue realizar comparaciones de la eficiencia de estos tres algoritmos. Concluyeron que los algoritmos genéticos generan trayectorias más eficientes. Mientras que Matekar et al. [14], utilizaron Evolución Diferencial y una función modificada de error basada en los errores longitudinales y transversales entre puntos de trayectorias prescritos y los puntos de trayectoria obtenidos.

Luego a estos, siguieron métodos de optimización cada vez más potentes como los presentados por Lee et al. [15], que implementaron el método de enjambre de partículas para realizar la síntesis de mecanismos planos de cuatro barras. Bulatović [16], utilizó el método modificado de Krill Herd, que está basado en la evolución natural de una manada de krill del antártico, obteniendo los mejores resultados de la síntesis de un mecanismo plano de cuatro barras para una trayectoria de 25 puntos de precisión. A partir de estas investigaciones se han realizado software y herramientas computacionales para la síntesis y análisis cinemático de mecanismos de cuatro barras como los diseñados por Pucheta et al. [17] y el desarrollado por Machado et al. [18].

Las aplicaciones computacionales para la síntesis están en creciente desarrollo. Actualmente, existen programas comerciales capaces de aplicar la síntesis de mecanismos por diferentes métodos de optimización y otros capaces de simular la cinemática del movimiento en mecanismos de cuatro barras, pero son costosos y algunos con desarrollo incipiente. Por lo tanto, en el presente artículo se describe el desarrollo del software especializado Mecanisynth 1.0 gratuito, basados en Algoritmos Genéticos (AG), realizado en el lenguaje de programación JAVA, que podrá ser ejecutado desde cualquier sistema operativo (Windows, Linux, Mac, etc.), el cual permite la optimización de la síntesis de generación de trayectorias, la simulación cinemática de mecanismos planos de cuatro barras para múl-

tiples puntos de precisión y la generación del reporte. Los resultados obtenidos se validan comparándolos con los obtenidos en otros estudios, con el propósito de comprobar la funcionalidad del software. El software desarrollado proporciona una herramienta fácil de usar para el diseño de mecanismos y reduce el tiempo del diseño cinemático, ya que con pocos parámetros proporciona todas las posibles alternativas para alcanzar una solución válida que satisfaga la trayectoria prescrita.

2. METODOLOGÍA

2.1 Entorno de desarrollo

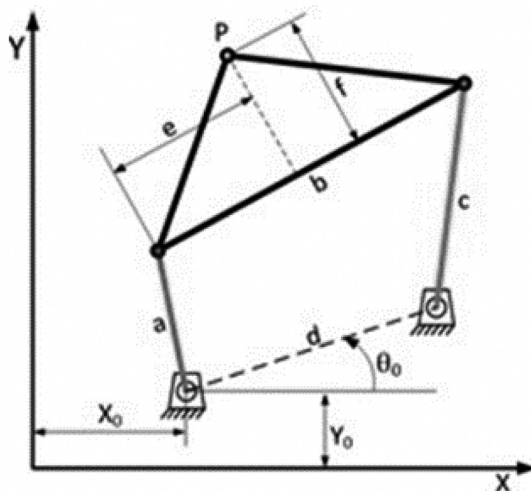
Para la codificación del software Mecanisynt 1.0 capaz de realizar la síntesis de generación de trayectoria y la simulación cinemática de mecanismos planos de cuatro barras, se realizó un algoritmo genético (GA) en el lenguaje de programación JAVA bajo un entorno integrado de desarrollo NetBeans IDE 8.0.2, el cual es un método adaptativo usado en problemas de búsqueda y optimización, que puede ser ejecutado desde cualquier sistema operativo (Windows, Linux, Mac, etc.). NetBeans IDE es un producto libre y sin restricciones de uso que permite desarrollar aplicaciones de escritorio, móviles y web. Para realizar la síntesis, el software solo requiere de los puntos que describen la trayectoria los cuales deben ser ingresados por el usuario a través de una interfaz gráfica GUI interactiva.

Inicialmente, se realizó un estudio de las técnicas y soluciones existentes, junto con las limitaciones y la planificación para superar las dificultades encontradas. Después de planificar los casos de uso y los requisitos funcionales y no funcionales del programa, se comenzó la codificación. En la búsqueda de posibles soluciones mediante métodos de optimización se deben identificar un conjunto de variables de diseño, las cuales deben describir el sistema mecánico requerido, en los mecanismos de cuatro barras pueden ser variables como dimensión de los eslabones, posición de los apoyos o ángulos entre los eslabones. Luego se define una función objetivo y las funciones que restringen al sistema. Esto es lo que se conoce como formulación matemática del problema de diseño.

2.2 Formulación de la función objetivo

La Figura 1 muestra un mecanismo planar de cuatro barras en una configuración prescrita cualquiera, donde XY es un sistema cartesiano de coordenadas fijo. La generación de trayectoria consiste en el control de un punto P (P_x , P_y) del eslabón acoplador a través de un grupo de puntos requeridos que describen una trayectoria y luego se evalúa la función error [1]. Para obtener un mecanismo de cuatro barras donde el punto P pase lo más cerca de la trayectoria deseada, se debe interactuar con las dimensiones de los eslabones y la posición de sus pivotes.

Figura 1. Variables de diseño.
Figure 1. Design variables.



2.3 Función objetivo

El objetivo del problema de la síntesis de trayectoria es tener un error de posición mínima, y para lograrlo, las variables de diseño que se deben optimizar son: manivela o barra de entrada(a), acoplador o biela (b), balancín o barra de salida (c) y barra de referencia (d). θ_0 es el ángulo de rotación del sistema de referencia y las coordenadas (e, f) determinan el punto P del eslabón acoplador. Para determinar la mejor solución se debe establecer la función de optimización o función objetivo, $f_{(x)}$ [16][19]. La función de optimización se muestra en las ecuaciones (1), (2) y (3).

$$f(X_i) = \sum_{j=1}^N \left[(P_{xa}^j - P_x^j)^2 + (P_{ya}^j - P_y^j)^2 \right] + M_1 h_1(X_i) \quad (1)$$

$$P_{xa}^j = A_x^j + r \cos \theta_s^j \quad (2)$$

$$P_{ya}^j = A_y^j + r \sin \theta_s^j \quad (3)$$

Donde X_i es el individuo número i de la población, N es el número de puntos de precisión que describen la trayectoria deseada por el usuario, (P_{xa}^j, P_{ya}^j) , es la ubicación del punto de precisión en la posición i , determinado a partir del análisis de posición, (P_x, P_y) , es la ubicación del punto de precisión, en la posición i requerida por el usuario, $h_1(X)$ es una constante que toma el valor de 0 si el mecanismo cumple con la condición de Grashof, y el valor de 1 en el caso contrario, M_1 es un peso constante adimensional, que penaliza a la función objetivo al no cumplir con la condición de Grashof, en este estudio se asignó un valor de $M_1 = 500$.

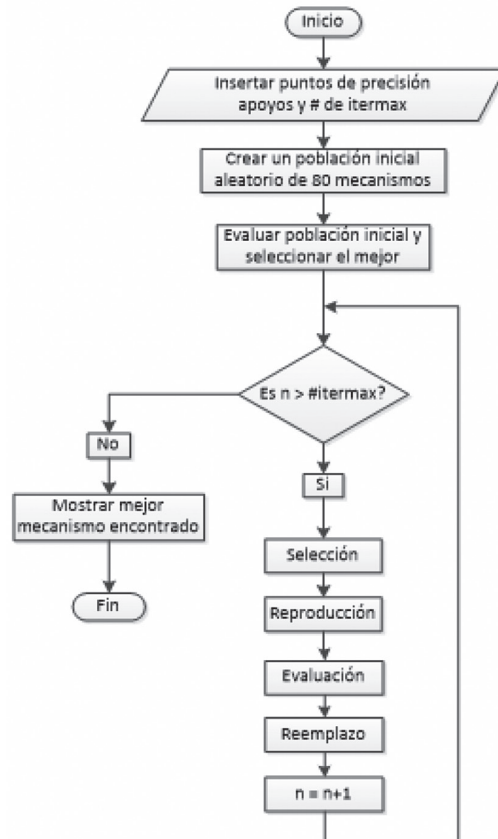
El criterio de Grashof es la condición predominante para que el mecanismo haga la revolución completa de la manivela o el enlace de entrada. s y $l \leq p + q$, donde, s y l son los enlaces más cortos y largos respectivamente y p y q son los otros dos enlaces del mecanismo.

2.4 Estrategia de optimización por Algoritmo Genético

Una vez que la cinemática del mecanismo se ha establecido apropiadamente, el problema de diseño debe definirse como un problema de optimización numérica, por lo que se requiere especificar las relaciones matemáticas que permitan evaluar el desempeño del sistema.

Los Algoritmos Genéticos (AG) son métodos adaptativos, usados generalmente en problemas de búsqueda y optimización, combinan la supervivencia del más apto con un intercambio de información estructurado y aleatorio, de un conjunto inicial de individuos llamado población, cada individuo contiene un cromosoma en el cual se almacena la información que representa a posible solución [19]. Los individuos de una población evolucionan tomando como base una serie de operadores genéticos, como lo son, la selección, cruzamiento y mutación, generando nuevas generaciones de individuos, en cada generación se busca optimizar una función objetivo, para obtener cada vez una generación de individuos más adaptados. En la Figura 2, se muestra el proceso de optimización del Algoritmo Genético diseñado.

Figura 2. Diagrama de flujo del Algoritmo Genético propuesto.
Figure 2. Flowchart of the proposed Genetic Algorithm.



2.4.1 Tamaño de la población inicial

Este valor debe ser estimado con métodos de ensayo y error. Una mala selección de este parámetro lleva a un AG ineficiente. Se recomiendan una población cercana a 100 individuos para problemas de síntesis de mecanismos, ya que una población con un número mayor de individuos consume más tiempo de procesamiento sin obtener mejoras en los resultados, en el caso contrario, una población pequeña no permite recorrer todo el espacio de búsqueda, y por consiguiente no se consigue un buen resultado [8] [15]. Para esta investigación se define este parámetro en 80 individuos.

2.4.2 Selección

En esta etapa se elige el mejor individuo según el valor de su función objetivo, y además se seleccionan otros dos de manera aleatoria con distribución uniforme, para formar un nuevo individuo [9], este esquema es conocido como evolución diferencial y se representa en la ecuación (4).

$$X_i = \begin{cases} X_{best} + F_c(X_1 + X_2), \varphi \leq C_r \\ X_{best} - F_c(X_1 + X_2), \varphi > C_r \end{cases} \quad (4)$$

Donde i es un número entero que recorre a toda la población de individuos [1, 80], X_i^* es un nuevo individuo seleccionado para ser cruzado con otro individuo en la etapa de reproducción, X_{best} es el mejor individuo de la población, F_c es un valor real positivo entre [0, 1] que controla la perturbación, con un valor asignado de $F_c = 0.45$; X_1 y X_2 son dos individuos de la población elegidos al azar, φ es un número real pseudoaleatorio entre [0, 1] generado con distribución uniforme, C_r es una constante que garantiza la diversidad de la población y se define como $C_r = 0.5$

2.4.3 Reproducción

La reproducción es un proceso en el cual un individuo es copiado de acuerdo con el valor de la función objetivo, de tal forma que los individuos de mejor comportamiento o funcionamiento tienen mayor probabilidad de pertenecer a la siguiente generación. Se divide en dos etapas conformadas por el cruzamiento y la mutación [20].

2.4.4 Cruzamiento

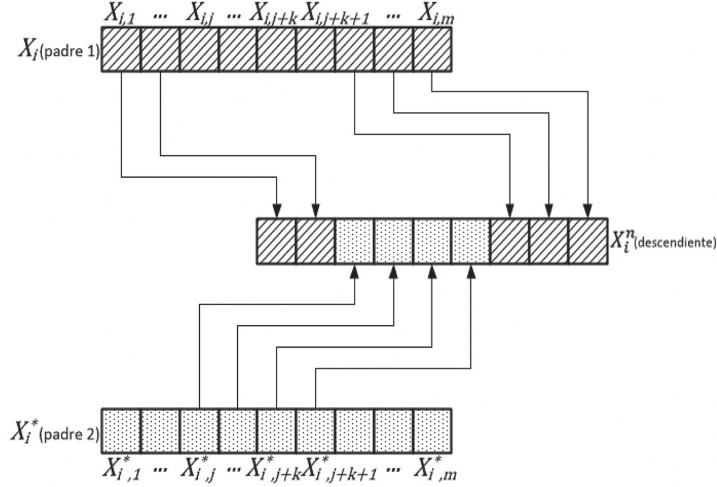
Ya que se tienen seleccionados los mejores individuos, el siguiente paso es hacer el cruce de estos individuos, que puede ser por un punto, dos puntos o uniforme. Es por esto por lo que se debe asignar una probabilidad de cruce CP que indica con qué frecuencia ocurre este. Este se representa en la ecuación (5) [20].

$$X_i^n = \begin{cases} \text{Cruzar}(X_i^*, X_i), |RandGauss| \leq CP \\ X_i^*, |RandGauss| > CP \end{cases} \quad (5)$$

Donde $RandGauss$, es un número pseudoaleatorio que sigue una distribución gaussiana o normal, con media $\mu = 0$ y desviación estándar $\sigma = 1$, se define $CP = 1.645$ para garantizar una probabilidad de cruce del 90%.

En el cruce el individuo X_i^* generado en la etapa de selección, se cruza con el individuo X_i de la población actual para generar el individuo X_i^n de la siguiente generación. La Figura 3 muestra la forma como se realiza la reproducción, una vez seleccionados dos individuos para la reproducción, se generan dos puntos de corte pseudoaleatorios con una distribución uniforme, se debe tener en cuenta que ninguno de estos puntos de corte coincida con los extremos de los cromosomas, para garantizar que se generen tres segmentos.

Figura 3. Reproducción por cruce multipuntos.
Figure 3. Reproduction by multipoint crossover.



2.4.5 Mutación

Es la que da una variación del valor de los genes en forma aleatoria, generalmente se da en valores muy pequeños, ya que no es tan significativo como el cruzamiento [9]. El modelo matemático se representa en la ecuación (6).

$$X_i^n = \begin{cases} Mutar(X_i^n), & |RandGauss2| > MP \\ X_i^n, & |RandGauss2| \leq MP \end{cases} \quad (6)$$

Donde *RandGauss2* es un número pseudoaleatorio que sigue una distribución gaussiana o normal, con media $\mu = 0$ y desviación estándar $\sigma = 1$, *MP* es la probabilidad de mutación, esta se establece en $MP = 2.576$ para una probabilidad de 1% de que ocurra la mutación en una generación. En la mutación el descendiente X_i^n sufre un cambio en uno de sus genes que está dado por la ecuación (7).

$$X_{i,randm}^n = FM * (X_{i,randm}^* - X_{randi,randm}^*) \quad (7)$$

Donde *randm* es un número entero pseudoaleatorio generado con distribución uniforme entre $\{1, 9\}$, este representa la ubicación del gen a mutar, $X_{i,randm}^n$ representa el gen seleccionado en la posición *randm* del individuo X_i^n , *FM* es un número real pseudoaleatorio con distribución uniforme entre $\{0, 1\}$ que determina la diferenciación entre los genes, $X_{i,randm}^*$ es el gen en la posición *randm* del segundo padre X_i^* , *randi* es un número pseudoaleatorio generado con distribución uniforme entre $\{1, 80\}$ y $X_{randi,randm}^*$ es el valor del gen en la posición *randm* del segundo padre del descendiente *randi*, se utilizan variables aleatorias con el fin de no condicionar el proceso de evolución y evitar que la población converja en mínimos locales.

2.4.6 Evaluación

A cada uno de los cromosomas de la población se aplicará la función objetivo para saber qué tan buena es la solución que se está codificando.

2.4.7 Reemplazo

Con la finalidad de obtener generaciones de individuos cada vez mejores, luego de la selección y reproducción, se procede a evaluar el error generado por el descendiente X_i^n , y a enfrentarlo con el primer padre X_i , donde el individuo con el menor valor en la función objetivo forma parte de la siguiente generación, de esta manera el tamaño de la población se mantiene en 80 y con individuos cada vez mejores. Como se muestra en la ecuación (8) [6].

$$X_i^{n+1} = \begin{cases} X_i^n, & f(X_i^n) \leq f(X_i) \\ X_i, & f(X_i^n) > f(X_i) \end{cases} \quad (8)$$

Donde X_i^{n+1} es el individuo número i de la siguiente generación ($n + 1$).

2.4.8 Condición de término

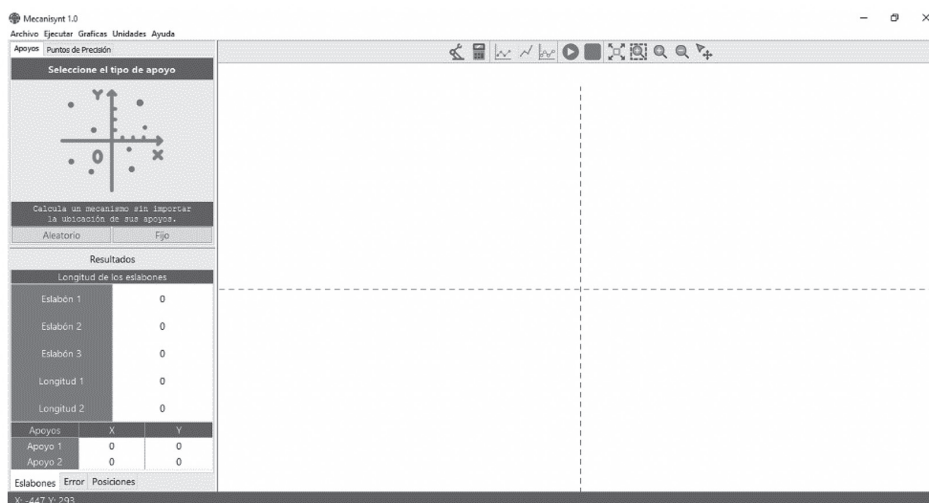
El Algoritmo Genético se deberá detener cuando se alcance la solución óptima, pero ésta generalmente se desconoce, por lo que se detendrá al alcanzar el número máximo de iteraciones definidas por el usuario a través de la interfaz gráfica.

3. RESULTADOS Y DISCUSIÓN

3.1 Presentación del software

El software fue diseñado para trabajar bajo cualquier sistema operativo (Windows, Linux, Mac, etc.), creando una interfaz de operación amigable al usuario final y de fácil uso, desarrollada en el entorno integrado NetBeans 8.0.2, presentada en la Figura. 4. Se tituló Mecanisynt 1.0 y el objetivo principal es diseñar la síntesis de trayectoria y la simulación cinemática de mecanismos planos de cuatro barras requeridos en la creación de maquinaria industrial, para un número infinito de posiciones, guiándolo mediante mensajes de recomendaciones y restricciones que le ayudarán a obtener un diseño óptimo.

Figura 4. Interfaz principal del software Mecanisynt 1.0.
Figure 4. Mecanisynt 1.0 software main interface.



3.2 Validación de la síntesis de trayectoria de mecanismos planos de cuatro barras

Para validar el software Mecanisynt 1.0, se comparan los resultados arrojados en problemas de síntesis de trayectoria de 18 y 25 puntos de precisión, con los resultados encontrados para los mismos problemas en estudios previos realizados por diferentes autores.

Para la validación de los resultados, en todos los problemas, se consideran los apoyos como aleatorios, por lo que el software deberá determinar la posición de estos. Se define el número máximo de iteraciones con respecto a la dificultad de cada problema, se debe empezar con un número medio de 300 iteraciones e ir subiendo hasta llegar a uno que entregue los mejores resultados para el problema requerido, también se puede utilizar desde el inicio un número alto de iteraciones, pero esto requerirá un alto tiempo de cálculo, dado que los resultados arrojados por el método de AG dependen en gran parte de la población inicial creada de forma aleatoria. Se debe realizar un número “n” de corridas para obtener el resultado óptimo, en los estudios seleccionados para la comparación, los autores definen en 50 el número de corridas, por lo que en este estudio se tomará la misma cantidad. Y, por último, Se define como error absoluto el error mínimo calculado dentro de las 50 corridas realizadas por el software.

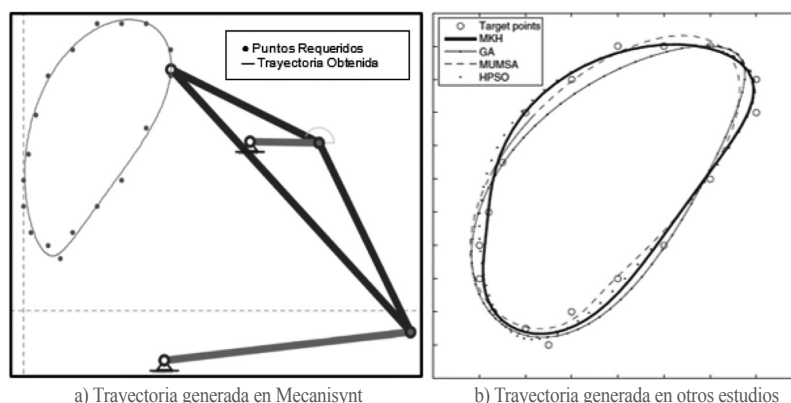
3.2.1 Validación de la síntesis de trayectoria para 18 posiciones

La Tabla 1 muestra dieciocho puntos de precisión que describen una trayectoria deseada. En la Figura 5(a) se muestra el mecanismo sintetizado con el software Mecanisynth 1.0 y en la Figura 5(b) se muestra la trayectoria generada con cuatro métodos de optimización diferentes.

Tabla 1. Puntos de precisión para la síntesis de trayectorias para 18 posiciones.
Table 1. Precision points for trajectory synthesis for 18 positions.

Punto	1	2	3	4	5	6	7	8	9
X	0.5	0.4	0.3	0.2	0.1	0.05	0.02	0	0
Y	1.1	1.1	1.1	1.0	0.9	0.75	0.6	0.5	0.4
Punto	10	11	12	13	14	15	16	17	18
X	0.03	0.1	0.15	0.2	0.3	0.4	0.5	0.6	0.6
Y	0.3	0.25	0.2	0.3	0.4	0.5	0.7	0.9	1.0

Figura 5. Gráfica comparativa de la síntesis de trayectoria para 18 posiciones.
Figure 5. Comparative plot of trajectory synthesis for 18 positions.



Fuente: La figura 5(b) tomada de Bulatović [16].

En la Tabla 2 se presentan la comparación entre los resultados obtenidos por seis métodos de optimización diferentes y los obtenidos por medio del software Mecanisynth 1.0. Se observa que los mejores resultados son los arrojados por el software Mecanisynth 1.0.

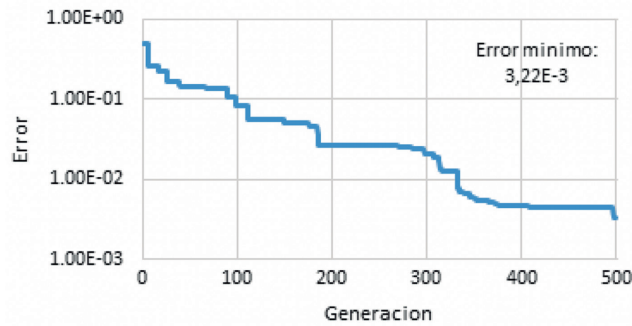
Tabla 2. Comparación de los resultados para 18 posiciones.
Table 2. Comparison of results for 18 positions.

	Kunjur and Krishnamurthy (GA)	Ortiz et al. (IOA)	Cabrera et al. (GA)	Cabrera et al. (MUMSA)	Lee and Lee (HPSO)	Bulatović et al. (MKH)	Mecanisynth 1.0 (MGA)
A	0,274853	0,245216	0,237803	0,297057	0,370000	0,421800	0,282
B	1,180253	6,38294	4,828954	3,913095	2,904800	0,878210	0,815
C	2,138209	2,620532	2,056456	0,849372	0,500000	0,580130	1,010
D	1,879660	4,040435	3,057878	4,453772	2,850000	1,004290	0,908
E	-0,833592	1,139106	0,767038	1,661063	1,020500	0,359070	-0,526
F	-0,378770	1,866109	1,850828	2,738736	1,689400	0,380810	-0,409
x_0	1,132062	1,891805	1,776808	2,806964	0,940000	0,268860	0,924
y_0	0,663433	-0,761339	-0,64199	4,853543	-1,17120	0,177150	0,648
θ_0	4,354224	1,187751	1,002168	-1,30924	0,760000	0,292940	1,175
θ_2 Pos1	2,558625	0,000000	0,226186	4,853543	0,513400	0,885950	-0,372
Error	4,30E-02	3,49E-02	3,37E-02	1,96E-02	1,10E-02	9,11E-03	3,22E-03

Fuente: Adaptado de Bulatović [16].

La Figura 6 representa la evolución del error a lo largo de 500 generaciones, se observa que, en la primera generación, el mejor mecanismo encontrado obtiene un error de 0,496 y en la última generación alcanza un error de 0,00322.

Figura 6. Evolución del error para la síntesis de trayectorias para 18 posiciones.
 Figure 6. Error evolution for trajectory synthesis for 18 positions.



3.2.2 Validación de la síntesis de trayectoria para 25 posiciones

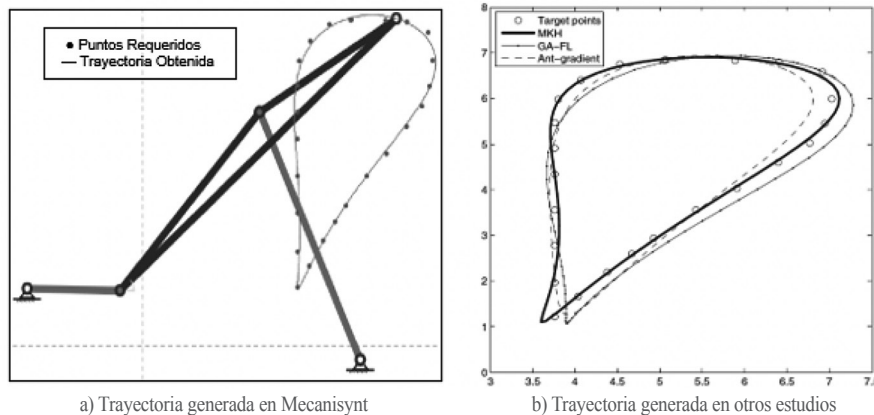
La Tabla 3 muestra veinticinco puntos de precisión que describen una trayectoria deseada.

Tabla 3. Puntos de precisión para la síntesis de trayectoria de 25 posiciones.
 Table 3. Precision points for 25-position trajectory synthesis.

Punto	1	2	3	4	5	6	7	8	9
X	7.03	6.95	6.77	6.4	5.91	5.43	4.93	4.67	4.38
Y	5.99	5.45	5.03	4.6	4.03	3.56	2.94	2.6	2.2
Punto	10	11	12	13	14	15	16	17	18
X	4.04	3.76	3.76	3.76	3.76	3.76	3.76	3.76	3.8
Y	1.67	1.22	1.97	2.78	3.56	4.34	4.91	5.47	5.98
Punto	19	20	21	22	23	24	25		
X	4.07	4.53	5.07	5.05	5.89	6.41	6.92		
Y	6.4	6.75	6.85	6.84	6.83	6.8	6.58		

El mecanismo obtenido a través de la síntesis por Algoritmos Genéticos es el mostrado en la Figura 7(a), donde se visualizan la trayectoria generada y los puntos de precisión requeridos. En la Figura 7(b) está representada la trayectoria obtenida con los métodos de optimización de los estudios comparativos.

Figura 7. Gráfica comparativa de la síntesis de trayectoria para 25 posiciones entre el software Mecanisynth 1,0 y otros estudios.
 Figure 7. Comparative plot of trajectory synthesis for 25 positions between Mecanisynth 1.0 software and other studies.



Fuente: La figura 7(b) tomada de Mohammad [21]

En la Tabla 4 se presentan los resultados obtenidos por tres métodos de optimización diferentes, en comparación a los obtenidos por medio del software Mecanisynt 1.0.

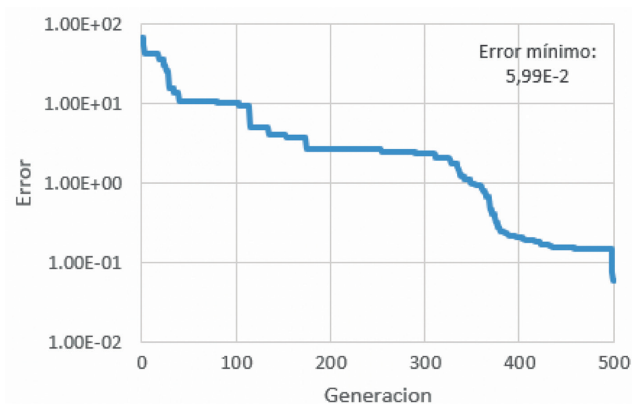
Tabla 4. Resultado síntesis de trayectoria para 25 posiciones.
Table 4. Summary result of trajectory for 25 positions

	Laribi et al. (GA-FL)	Smaili and Diab (ant-gradient)	Bulatović et al. (MKH)	Mecanisynt 1.0 (MGA)
A	3,01	1,89	1,93027	2,249
B	8,8	8,41	4,57242	5,05
C	8,8	6,75	7,36674	5,746
D	9	13,08	9,99432	8,202
E	8,626	14,177	7,90168	8,724
F	-6,985	2,794	1,49860	-1,138
x₀	-2,4	-8,77	-2,31301	-2,786
y₀	-4	1,20	2,86189	1,196
θ₀	0,489	-0,3815	-0,61763	0,183
Error	9,02E-01	5,50E-01	3,92E-02	5,99E-02

Fuente: Adaptado de Mohammad [21]

La Figura 8 representa la evolución del error a lo largo de 500 generaciones, se observa que, en la primera generación, el mejor mecanismo encontrado obtiene un error de 53,1 y en la última generación alcanza un error de 0,0599.

Figura 8. Evaluación del error para síntesis de trayectoria para 25 posiciones.
Figure 8. Error evaluation for path synthesis for 25 positions.



3.2.3 Comparación del software Mecanisynt 1.0 vs SAM 7.0

La comparación se realizó para 6, 9 y 12 puntos de precisión. Se muestran los resultados para 9 puntos de precisión.

Los parámetros iniciales utilizados en el software SAM 7.0 fueron los siguientes: $0 \leq a_1, a_2, a_3, a_4 \leq 20$; $-20 \leq e, f, O_{2x}, O_{2y} \leq 20$; $0 \leq \theta_1 \leq 2\pi$. Número máximo de iteraciones: 1000. Número de corridas: 50.

Y los parámetros iniciales para el software Mecanisynt 1.0 fueron: Número máximo de iteraciones: 300. Número de corridas: 50. Tipo de apoyo: Aleatorio.

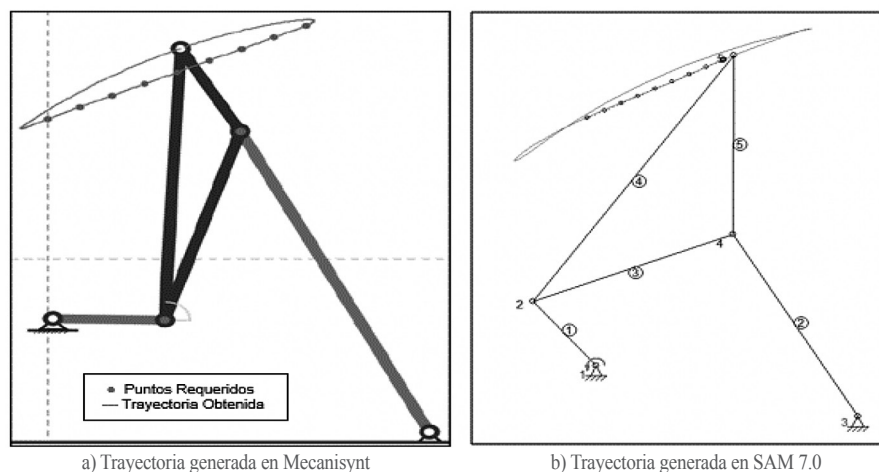
En la Tabla 5 se muestran los nueve puntos de precisión que describen una trayectoria determinada.

Tabla 5. Puntos de precisión para la síntesis de trayectoria de 9 posiciones.
Table 5. Precision points for 9-position trajectory synthesis.

Punto	1	2	3	4	5	6	7	8	9
X	8.0	7.0	6.0	5.0	4.0	3.0	2.0	1.0	0.0
Y	10.0	9.5	9.0	8.5	8.0	7.5	7.0	6.5	6.0

En la Figura 9 se muestra a la izquierda el mecanismo sintetizado con el software Mecanisynth 1.0 y a la derecha el mecanismo sintetizado por el software SAM 7.0.

Figura 9. Gráfica comparativa de la síntesis de trayectoria para 9 posiciones, software Mecanisynth 1.0 Vs SAM 7.0
Figure 9. Comparative plot of trajectory synthesis for 9 positions, Mecanisynth 1.0 Vs SAM 7.0 software.



En la Tabla 6 se presentan los resultados obtenidos por medio del software SAM 7.0, en comparación a los obtenidos por medio del software Mecanisynth 1.0.

Tabla 6. Resultados de la síntesis de trayectoria para 9 posiciones.
Table 6. Trajectory synthesis results for 9 positions.

	SAM 7.0	Mecanisynth 1.0
A	5,774	3,459
B	14,55	8,415
C	12,65	14,116
D	15,798	12,601
E	17,210	11,323
F	11,413	2,773
x_0	0,524	0,166
y_0	-11,05	-2,562
θ_0	-12,7994	-22,5292
Error	9,11E-03	2,68E-04

3.2.4 Validación del análisis cinemático

Para la validación del análisis cinemático de un mecanismo plano de cuatro barras se escogió un mecanismo de Grashof, las dimensiones de este se muestran en la Tabla 7.

Tabla 7. Dimensiones para la síntesis cinemática de mecanismos planos de 4 barras.
Table 7. Dimensions for kinematic synthesis of 4-bar planar mechanisms.

Eslabón 1	Eslabón 2	Eslabón 3	Longitud E	Longitud F	O1x	O1y	O2x	O2y
10	60	20	30	30	0	0	50	5

Las Figuras 10 y 11 muestran las trayectorias generadas en el software Mecanisynth 1.0 y SAM 7.0.

Figura 10. Simulación del recorrido de un mecanismo plano de cuatro barras en el software Mecanisynth 1.0.
 Figure 10. Simulation of the path of a flat four-bar mechanism in Mecanisynth 1.0 software.

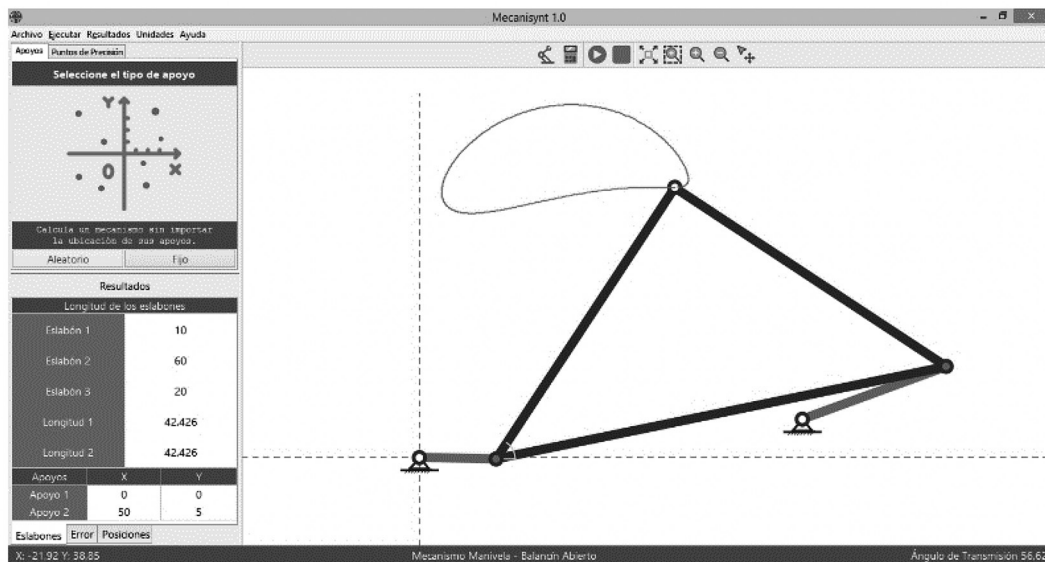
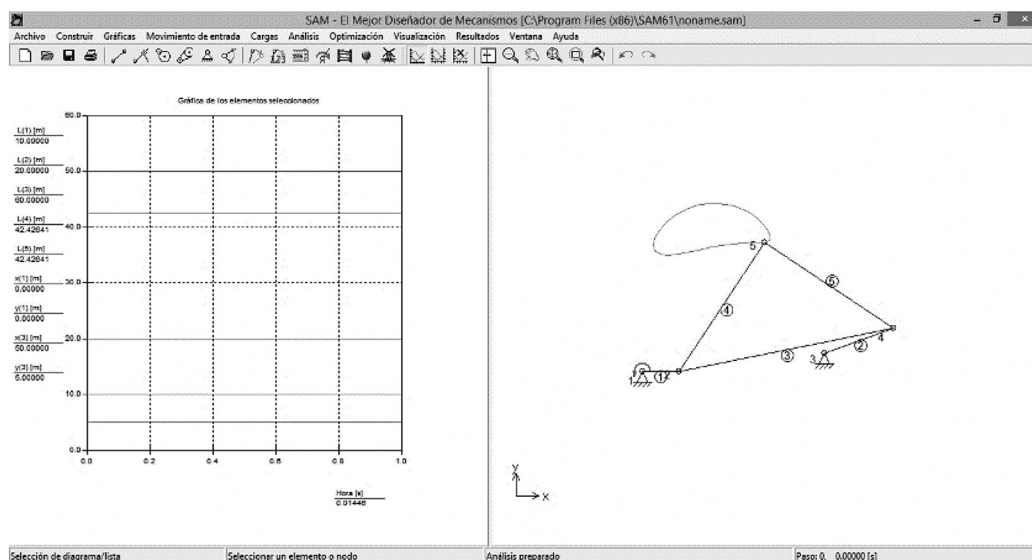


Figura 11. Simulación del recorrido de un mecanismo plano de cuatro barras en el software SAM 7.0.
 Figure 11. Simulation of the path of a flat four-bar mechanism in SAM 7.0 software.



Se observa que las trayectorias mostradas por los programas son semejantes, pero para que la validación fuera más precisa se analizó las gráficas de posición, velocidad y aceleración generadas por cada uno del software. Solo se muestran las posiciones para el nodo 1. (Figuras 12, 13 y 14).

Figura 12. Gráficas de comparación para la posición del nodo 1.
Figure 12. Comparison graphs for node 1 position.

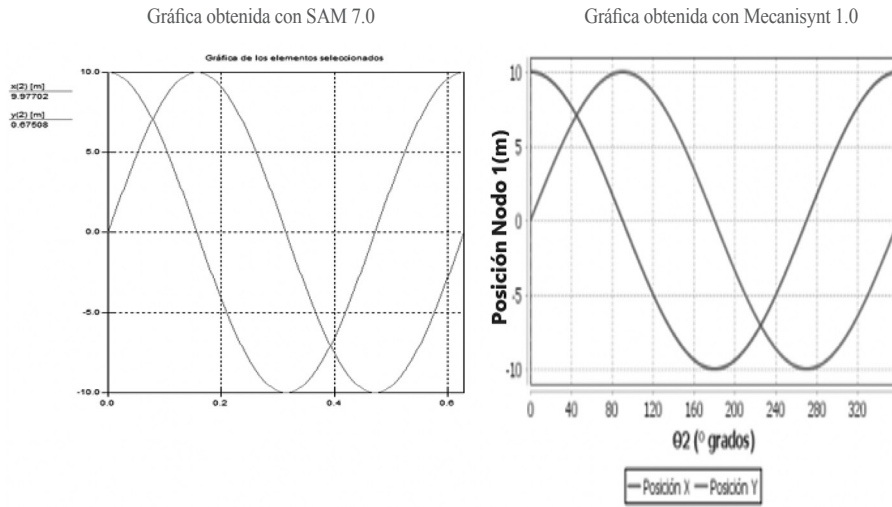


Figura 13. Gráfica de comparación para la velocidad del nodo 1.
Figure 13. Comparison graph for node 1 velocity.

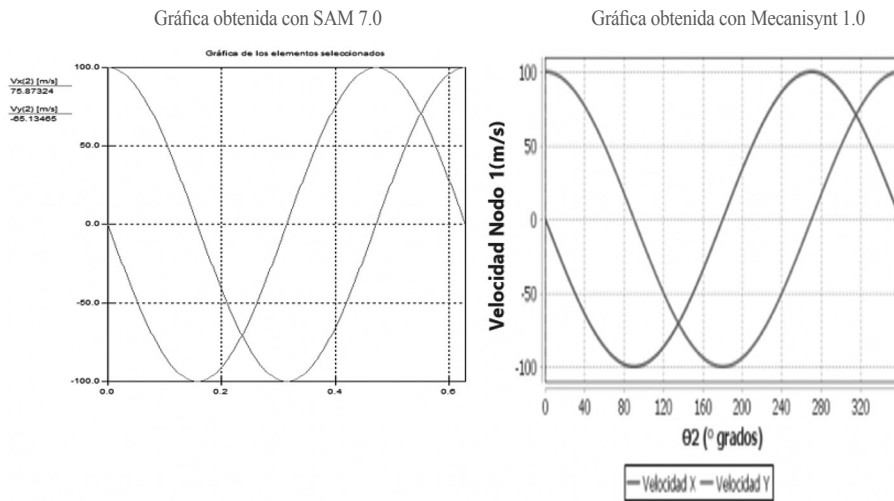
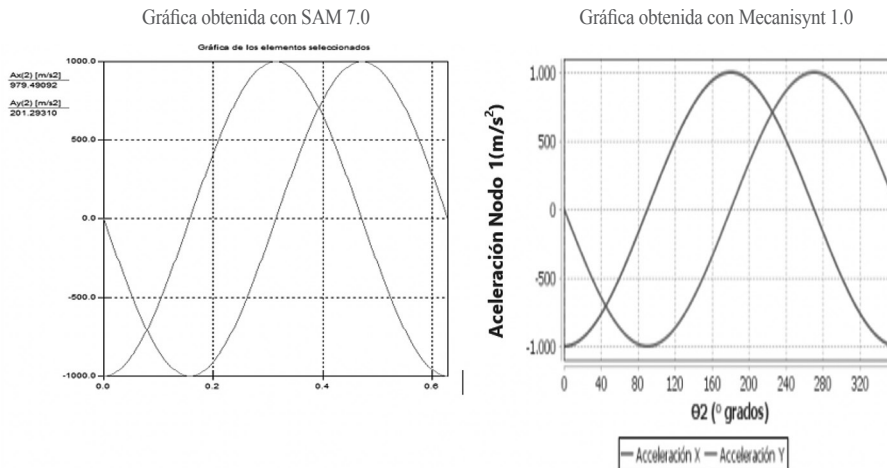


Figura 14. Gráfica de comparación para la aceleración del nodo 1.
Figure 14. Comparison plot for node 1 acceleration.



4. CONCLUSIONES

Se desarrolló un software especializado Mecanisynt 1.0 que permite la optimización por Algoritmos Genéticos de la síntesis de trayectorias para mecanismos planos de cuatro barras, respondiendo a requerimientos cinemáticos dados por el usuario, de manera fiable, práctico y con una tasa alta de precisión. La pequeña diferencia entre la síntesis de trayectorias realizadas por software comerciales y el realizado por Mecanisynt 1.0, es que el usuario debe seleccionar el mecanismo que satisfaga todas sus necesidades, ya que por estar potenciado por Algoritmos Genéticos puede entregar diferentes soluciones para un mismo problema.

Para validar la interfaz gráfica del software, se sometió a una encuesta con 80 potenciales usuarios los cuales resaltaron la facilidad de uso de esta herramienta y la estabilidad del software en sistemas operativos como Windows, MAC, y Linux. Para la validación de la síntesis se compararon los resultados obtenidos por el software Mecanisynt 1.0 con problemas reportados en la literatura para trayectorias de 18 y 25 puntos de precisión y con los resultados obtenidos con el software SAM 7.0. Se puede concluir que: Los Algoritmos Genéticos producen resultados adecuados para encontrar configuraciones óptimas desde el punto de vista del diseño en ingeniería, por lo que garantiza la confiabilidad del software. Es un algoritmo simple de fácil implementación y no requiere el uso de recursos de cómputos excesivos. Para el mecanismo de cadena cerrada se evita el cálculo del método numérico, que es laborioso y tiene restricciones como el dar datos aproximados para poder dar una solución. En el problema de generación de trayectorias, el error se reduce aproximadamente en 1,5, lo que se debe al tamaño de la población.

Los autores piensan que el programa desarrollado puede ser usado en otras situaciones prácticas e investigaciones que involucran determinar la síntesis de trayectoria y el análisis cinemático completo de mecanismos planos de cuatro barras, y que el tiempo ahorrado por la facilidad y rapidez de realizar la síntesis, a diferencia de los métodos analíticos, junto con el hecho de que el programa es gratuito y tiene una interfaz amigable, puede ayudar en las actividades de los estudiantes, docentes y diseñadores que requieran este tipo de software.

Se están realizando nuevos esfuerzos para mejorar los bloques funcionales del software, especialmente adaptarla para otro tipo de mecanismos como planares, esféricos y espaciales.

5. REFERENCIAS

- [1] R.L, Norton. "Design of machinery: an introduction to the synthesis and analysis of mechanisms and machines". McGraw-Hill. Sexta Edición, United States of America. ISBN: 9781260113310. 2020.
- [2] J.A, Hrones, N. G. L, Nelson. "Analysis of the Four-bar Linkage - It's Application to the Synthesis of Mechanisms". Mit Press and Wiley, pp 730. New York. ISBN: 9780262080033. 1951.
- [3] C.H, Suh and C.W, Radcliffe. "Kinematics and Mechanism Design". John-Wiley. New York. 1978.
- [4] A. Erdman y G. Sandor. "Diseño de Mecanismos Análisis y Síntesis". Prentice Hall Hispanoamérica S.A. 3 Ed. 119-232. México D.F. México. 1998.
- [5] Gajanan G. Waghmare, R. V. Rao & Prafulla C. Kulkarni. "Path synthesis of a four-bar linkage using a teaching-learning-based optimization algorithm". International Journal for Computational Methods in Engineering Science and Mechanics, 24:1. pp. 40-51. 2023. DOI: 10.1080/15502287.2022.2042869.
- [6] Zhang, W., Liu, Z., Liu, W., Sun, J., and Lu, H. "Dimensional synthesis of a spherical linkage crank slider mechanism for motion generation using an optimization algorithm". Mech. Sci., 14, pp.125–142. 2023. <https://doi.org/10.5194/ms-14-125-2023>.
- [7] R, Peón Escalante, J, Méndez Gamboa, J., C, Villanueva López, D, Pech Oy. "Síntesis Dimensional Óptima de un Mecanismo Planar RRRR Usando Algoritmos Genéticos". Ingeniería Revista Académica. Vol.13 N°3, pp. 5-16. 2009. ISSN: 1665-529X.
- [8] Sancibrian, R, De-Juan, A, & Viadero, F. "Non-Linear Least-Square Optimization of Mechanisms Based on Levenberg-Marquardt Method". Proceedings of the ASME 2008 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. Volume 2: 32nd Mechanisms and Robotics Conference, Parts A and B. Brooklyn, New York, USA. pp. 1477-1484. ASME. August 3–6, 2008. <https://doi.org/10.1115/DETC2008-49743>.
- [9] Mehmet Erkan Kutuk. Lale Canan Dulger. "Optimization with Genetic Algorithm (GA): Planar Mechanism Synthesis". Scientia Iranica. November 2022. doi: 10.24200/sci.2022.56432.4738.
- [10] Mo, X., Ge, W., Zhao, D., Zhang, Y. "Path Synthesis of Crank-Rocker Mechanism Using Fourier Descriptors Based Neural Network". Mechanisms and Machine Science, vol 79. Springer, Singapore. 2020. https://doi.org/10.1007/978-981-15-0142-5_4

- [11] R.S. Kalnas, S. Kota. "Incorporating Uncertainty into Mechanism Synthesis". Mechanism and Machine Theory. Vol 36 No 3, pp 843 - 851. 2001. DOI: [10.1016/S0094-114X\(01\)00019-2](https://doi.org/10.1016/S0094-114X(01)00019-2).
- [12] A. Kunjur, S. Krishnamurthy. "Genetic Algorithms in Mechanical Systems", pp. 18-24. Fecha de consulta: 15 de noviembre de 2019. URL: <http://www.ecs.umass.edu/mie/labs/mda/mechanism/papers/genetic.html>.
- [13] S.K. Acharyya, M. Mandal. "Performance of EAs for Four-bar Linkage Synthesis". Mechanism and Machine Theory. Vol. 44 N°9, pp. 1784–1794. 2009. DOI: [10.1016/j.mechmachtheory.2009.03.003](https://doi.org/10.1016/j.mechmachtheory.2009.03.003).
- [14] S. B. Matekar, G. R. Gogate. "Optimum Synthesis of Path Generating Four-bar Mechanisms Using Differential Evolution and a Modified Error Function". Mechanism and Machine Theory, vol. 52, pp. 158–179. 2012. DOI: [10.1016/j.mechmachtheory.2012.01.017](https://doi.org/10.1016/j.mechmachtheory.2012.01.017).
- [15] L. Chun-te, L. Chun-che. "On a Hybrid Particle Swarm Optimization Method and its Application in Mechanism Design". Journal of Mechanical Engineering Science. Vol. 228, pp. 2844-2857. 2014. DOI: [10.1177/0954406214522206](https://doi.org/10.1177/0954406214522206).
- [16] R. Bulatović, G. Miodragović, M. Bošković. "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. DOI: [10.1016/j.mechmachtheory.2015.08.004](https://doi.org/10.1016/j.mechmachtheory.2015.08.004).
- [17] M. A. Pucheta, A. Cardona. "Software para Síntesis de Mecanismos Planos". Mecánica Computacional. Vol. 23, pp. 3369-3389. 2014. En línea: <https://cimec.org.ar/ojs/index.php/mc/article/viewFile/451/434>.
- [18] D. Machado Mercado, G. Herrera Murgas, J. Roldán Mckinley, J. Díaz González. "Una Herramienta Computacional Didáctica para el Análisis Cinemático de Mecanismos Planos de Cuatro Barras". Revista UIS Ingenierías. Vol.14 N°1, pp. 59-69. 2015. ISSN en línea:2145-8456.
- [19] S. Ebrahimi, P. Payvandy. "Efficient Constrained Synthesis of Path Generating Four-bar Mechanisms Based on the Heuristic Optimization Algorithms". Mechanism and Machine Theory. Vol. 85, pp. 189-204. 2015. DOI: [10.1016/j.mechmachtheory.2014.11.021](https://doi.org/10.1016/j.mechmachtheory.2014.11.021).
- [20] M. Gestal, D. Rivero, J. Rabuñal, A. Pazos. "Introducción a los Algoritmos Genéticos y la Programación Genética". Servicio de publicaciones Universidad de A. Coruña. 1, Edición. pp. 11-48. España. ISBN: 978-84-9749-422-9. 2010.
- [21] M. Mohammad Etefagh, M. Saeidi Javash. "Optimal Synthesis of Four-bar Steering Mechanism Using AIS and Genetic Algorithms". Journal of Mechanical Science and Technology. Vol. 28 N°6, pp. 2351-2362, Jun. 2014. DOI: [10.1007/s12206-014-0526-3](https://doi.org/10.1007/s12206-014-0526-3).