

Síntesis dimensional óptima de un mecanismo para seguimiento de trayectoria por medio de búsqueda armónica y evolución diferencial

Alvaro Sánchez-Márquez, Silvia Sánchez-Márquez,
Josefina Hernández-Tapia, Alberto Hernández-Lazcano,
Lyonel Sergio Carrasco-Pérez, Claudia Alicia Romero-León

Universidad Autónoma de Tlaxcala
Unidad Académica Multidisciplinaria,
Campus Calpulalpan,
México

{asanchez, silvia.sanchez, jhernandezt, ahlazcano
20012178, 20154466}@uatx.mx

Resumen. En este trabajo se presenta la síntesis dimensional de un mecanismo de cuatro barras para el seguimiento de una trayectoria definida geoméricamente como semicírculo, dicho caso de estudio se plantea como un problema de optimización numérico con restricciones. Es importante mencionar que este tipo de problemas se vuelven más complejos cuando las restricciones aumentan, sin embargo es resuelto por algoritmos heurísticos, específicamente búsqueda armónica y evolución diferencial. Aun considerando la complejidad dada por el número de restricciones de este caso de estudio (25 en nuestro caso); los resultados obtenidos determinan las proporciones de los eslabones para el mecanismo, se consideran satisfactorios en base al error obtenido que es de precisión 10^{-6} .

Palabras clave: Síntesis dimensional, optimización, búsqueda armónica, evolución diferencial.

Optimal Dimensional Synthesis of a Mechanism for Trajectory Tracking Through Harmonic Search and Differential Evolution

Abstract. This work presents the dimensional synthesis of a four-bar mechanism for tracking a trajectory geometrically defined as a semicircle. This case study is formulated as a numerical optimization problem with restrictions. It is important to mention that kind of problems become more complex as the restrictions increase; however, it is solved by heuristic algorithms, specifically harmonic search and differential evolution. Even considering the complexity given by the number of restrictions of this case study (25 in our case); the results obtained determine the proportions of the links for the mechanism, they are considered satisfactory based on the error obtained, which is precision 10^{-6} .

Keywords: Dimensional synthesis, optimization, harmonic search, differential evolution.

1. Introducción

El hombre ha mostrado su ingenio para idear máquinas y mecanismos con el objetivo de tener una vida más fácil y cómoda, las máquinas son dispositivos que se utilizan al modificar, transmitir y dirigir fuerzas para llevar a cabo un objetivo específico [12]. El diseño de éstos mecanismos se enfoca en principalmente tres técnicas: síntesis de tipo (clase de mecanismo seleccionado), síntesis cuantitativa o analítica (número de elementos del mecanismo) y la síntesis dimensional (longitud y ángulo de los elementos del mecanismo), el mayor interés en este trabajo radica en ocuparse en la técnica de síntesis dimensional debido a que se busca las dimensiones óptimas de un mecanismo de cuatro barras para el seguimiento de una trayectoria específica.

Los problemas de síntesis dimensional se pueden resolver mediante dos métodos: los métodos de programación matemática (también llamados clásicos) y los métodos heurísticos (específicamente los algoritmos metaheurísticos), los cuales son procedimientos de búsqueda y han demostrado ser capaces de obtener una buena solución (aunque no garantizan encontrar el valor óptimo) a problemas difíciles.

Estos métodos imitan fenómenos simples observados en la naturaleza, destacan algoritmos genéticos (Genetic Algorithm, GA) [2], optimización basada en colonia de hormigas (Ant Colony optimization) [17], colonia artificial de abejas (Artificial Bee Colony, ABC) [20], algoritmo de luciérnagas (Firefly Algorithm, FA) [21], Evolución diferencial (Differential Evolution, ED) [16], Búsqueda Armónica (Harmony Search, HS) [22], por mencionar algunos.

El presente trabajo aborda el seguimiento de trayectoria por medio de búsqueda armónica, un algoritmo metaheurístico que basa su funcionamiento en el proceso de la improvisación de músicos de jazz, fue desarrollado por Geem et al. en 2001 [10]. Los algoritmos de búsqueda armónica han sido aplicados para una gran amplia variedad de trabajos tales como: identificación de parámetros para modelos de celdas solares [1], diseño óptimo de costes de las redes de distribución de agua [11], diseño geométrico óptimo de Cúpulas Geodésicas [15], composiciones musicales [9], entre otros.

Por otro lado, el algoritmo de evolución diferencial propuesto por Storn y Price [16] es un modelo evolutivo basado en la población, utilizado para resolver numerosos problemas complejos, algunas investigaciones con aplicación de este algoritmo metaheurístico están basadas en: la síntesis óptima de un mecanismo plano para seguimiento de trayectoria [18], la sintonización de clasificadores difusos para el reconocimiento del lenguaje de señas [19], control de un motor de corriente directa bajo incertidumbre paramétrica [8], diagnóstico de fallos en sistemas industriales [5], diseño automático de redes neuronales artificiales [7], por citar algunos.

Este proyecto de investigación aborda el análisis del mecanismo de cuatro barras, así como su cinemática; se determinan las particularidades del problema de optimización numérica y la trayectoria propuesta para el caso de estudio. Los algoritmos aplicados para resolver el problema de optimización son búsqueda armónica y evolución diferencial, cuyos resultados obtenidos muestran la mejora significativa con Evolución diferencial.

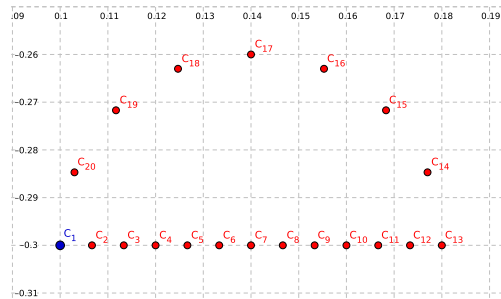


Fig. 1. Trayectoria deseada (\bar{x}_E, \bar{y}_E) .

2. Análisis del mecanismo

Un mecanismo de cuatro barras está formado por tres eslabones móviles y una barra fija, gráficamente se observa en la Figura 1. La longitud de cada una de las barras se representa por l_i , $i = 1, 2, 3, 4$, l_5 y l_6 son distancias estratégicas entre algunos vértices, el desplazamiento angular con respecto a la horizontal lo indica θ_i , r_{cx} y r_{cy} señalan la posición del acoplador C .

2.1. Cinemática del mecanismo

Para determinar la posición del punto C a lo largo de una trayectoria se realiza la síntesis del mecanismo a través del modelado geométrico, se calcula l_5 dado por (1) y por ley de cosenos el ángulo α (2):

$$l_5^2 = l_1^2 + l_2^2 - 2 l_1 l_2 \cos(\theta_2), \quad (1)$$

$$\alpha = \cos^{-1} \left(\frac{l_3^2 + l_4^2 - l_5^2}{2 l_3 l_4} \right). \quad (2)$$

Los ángulos β y δ son calculados por ley de senos y se muestran en las ecuaciones (3) y (4):

$$\beta = \sin^{-1} \left(\frac{l_2 \sin(\theta_2)}{l_5} \right), \quad (3)$$

$$\delta = \sin^{-1} \left(\frac{l_4 \sin(\alpha)}{l_5} \right). \quad (4)$$

Sin embargo el ángulo δ está conformado por los ángulos θ_3 y β como se aprecia en la Figura ??, por lo que la ecuación (1) consigue calcular el ángulo θ_3 :

$$\theta_3 = \begin{cases} \delta + \beta & \theta_2 \leq \pi, \\ \delta - \beta & \theta_2 > \pi. \end{cases} \quad (5)$$

Tabla 1. Límites de variables de diseño.

	l_1	l_2	l_3	l_4	r_{cx}	r_{cy}	θ_0	x_0	y_0	θ_2^1	...	θ_2^{20}	\bar{x}_{ini}
P_{max}	0.5	0.5	0.5	0.5	0.5	0.5	2π	60	60	2π	...	2π	0.5
P_{min}	0	0	0	0	0	0	0	-60	-60	0	...	0	-0.5

La longitud l_6 y el ángulo φ indicados en la Figura 1 corresponden a las ecuaciones (6) y (7):

$$l_6^2 = r_{cx}^2 + r_{cy}^2, \quad (6)$$

$$\varphi = \tan^{-1} \left(\frac{r_{cy}}{r_{cx}} \right). \quad (7)$$

Una vez que se han encontrado los parámetros que se incluyen en el diagrama geométrico que se muestra en la la Figura 1, entonces, es posible determinar la posición geométrica del punto acoplador C y esta dada por:

$$\begin{bmatrix} C_x \\ C_y \end{bmatrix} = \begin{bmatrix} l_2 \cos(\theta_0 + \theta_2) + l_6 \cos(\varphi + \theta_3 + \theta_0) \\ l_2 \sin(\theta_0 + \theta_2) + l_6 \sin(\varphi + \theta_3 + \theta_0) \end{bmatrix} + \begin{bmatrix} x_0 \\ y_0 \end{bmatrix}. \quad (8)$$

3. Trayectoria semicircular para el seguimiento

La trayectoria propuesta tiene una recta horizontal que parte desde el punto $C_1 = (\bar{x}_{ini}, \bar{y}_{ini})$ hasta el punto C_{13} , la mitad de la circunferencia esta dada por 7 puntos $C_{14} \dots C_{20}$, es decir, la trayectoria está formada por 20 coordenadas cartesianas, de las cuales 13 puntos forman la alineación horizontal y los 7 puntos restantes forman la mitad de la circunferencia (ver Figura 1). Dicha trayectoria fue propuesta en el trabajo [13] donde se utilizó un mecanismo planar de ocho eslabones con un grado de libertad como extremidad bípida para el seguimiento similar a la marcha humana que está definida por el desplazamiento en forma de semicírculo. El vector solución p está dado por 29 variables de diseño presentado en (9):

$$p = [l_1, l_2, l_3, l_4, r_{cx}, r_{cy}, \theta_0, x_0, y_0, \theta_2^1, \theta_2^2, \dots, \theta_2^{20}]^T \in \mathbb{R}^{29}, \quad (9)$$

donde:

- l_1, l_2, l_3, l_4 corresponden a las longitudes de las barras,
- r_{cx}, r_{cy} indican la posición del acoplador,
- θ_0 es el ángulo de orientación del sistema con respecto a la horizontal,
- x_0, y_0 señalan el origen del sistema de referencia, y
- $\theta_2^1, \dots, \theta_2^{20}$ son los valores de los ángulos de la barra l_2 con respecto a los 20 puntos dados en la Figura 1.

Algoritmo 1.1: Algoritmo de búsqueda armónica.

```

1 definir función objetivo  $f(x)$ ,  $x = (x_1, x_2, \dots, x_N)$ ;
2 definir tasa de selección de la memoria armónica  $r_{\text{accept}}$ ;
3 definir tasa de ajuste de tono  $r_{pa}$ ;
4 definir rango de ajuste de tono  $bw$ ;
5 generar memoria armónica inicial  $MA$  (arreglo  $(k \times N)$ );
6 while  $g <$  máximo número de iteraciones do
7   while  $i \leq N$  do
8     if  $\text{rand} < r_{\text{accept}}$  then
9       índice  $\leftarrow \text{rand}(1, k)$ ;
10      if  $\text{rand} < r_{pa}$  then
11        newH( $i$ )  $\leftarrow MA(\text{índice}, i) + bw * \text{rand}(-1, 1)$ ;
12      else
13        newH( $i$ )  $\leftarrow MA(\text{índice}, i)$ 
14      else
15        newH( $i$ )  $\leftarrow \text{rand}(L_i, U_i)$ 
16  reemplazar la nueva armonía por la peor de MA, si esta es mejor ;

```

3.1. Ley de Grashof

La Ley de Grashof afirma que la barra más corta de un mecanismo de cuatro barras da vueltas enteras respecto a todas las otras, si se cumple que la suma de la longitud de la barra más larga l y la de la más corta s es más pequeña o igual que la suma de las longitudes de las otras dos p y q [6], es decir:

$$s + l \leq p + q. \quad (10)$$

Cabe mencionar que el sistema mecánico planteado en este trabajo la Ley de Grashof esta dada por la ecuación 11:

$$l_1 + l_2 \leq l_3 + l_4. \quad (11)$$

3.2. Solución factible

La solución de un problema que satisface el total de las restricciones planteadas se conoce como **solución factible**. El planteamiento de restricciones obliga a considerar ciertos métodos o reglas para el manejo de las mismas, para este caso se utilizan las reglas de viabilidad de Deb [4].

1. Entre dos soluciones factibles, se selecciona la que tiene el mejor desempeño en la función objetivo.
2. Entre una solución factible y una no factible, se elige la solución factible.
3. Entre dos soluciones no factibles, se prefiere el que tiene la suma más baja de violaciones de restricciones.

Algoritmo 1.2: Algoritmo de evolución diferencial.

- 1 Generar aleatoriamente una población inicial de soluciones
 $x_{j,i} = b_{j,L} + (b_{j,U} - b_{j,L}) * \text{rand}(0, 1)$;
- 2 Repetir;
- 3 Seleccionar un padre y tres individuos aleatoriamente ($X_{r_0,g}, X_{r_1,g}, X_{r_2,g}$);
- 4 Crear un hijo $U_{i,g}$ a partir de la cruce o recombinación entre vector padre $X_{i,g}$ y el vector de mutación $V_{i,g}$;
- 5 Evaluar la aptitud y factibilidad del hijo generado;
- 6 **if** $f(U_{i,g}) \leq f(X_{i,g})$ **then**
- 7 | $U_{i,g}$;
- 8 **else**
- 9 | $X_{i,g}$
- 10 Hasta que se satisfaga una condición de paro;

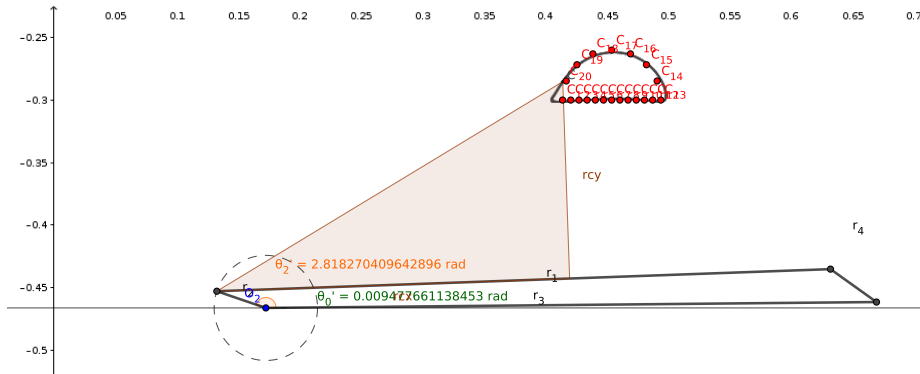


Fig. 2. Mecanismo óptimo: Búsqueda armónica.

4. Esquema de optimización

El problema de optimización numérico es descrito por las ecuaciones (12) a (13), el propósito es obtener la solución de la síntesis dimensional del mecanismo de cuatro barras que siga la trayectoria definida por el semicírculo que se muestra en la Figura 1.

4.1. Función objetivo

La función objetivo está dada por la suma de las diferencias cuadradas de la trayectoria deseada $(\bar{x}_E^i, \bar{y}_E^i)$ y la trayectoria obtenida (x_E^i, y_E^i) , de las 20 coordenadas cartesianas (ecuación (12)):

$$\text{mín } f(p) = \sum_{i=1}^{\bar{n}=20} [(\bar{x}_E^i - x_E^i)^2 + (\bar{y}_E^i - y_E^i)^2] \quad p \in \mathbb{R}^{29}. \quad (12)$$

4.2. Restricciones

Las restricciones de diseño se expresan con las ecuaciones (13) correspondientes:

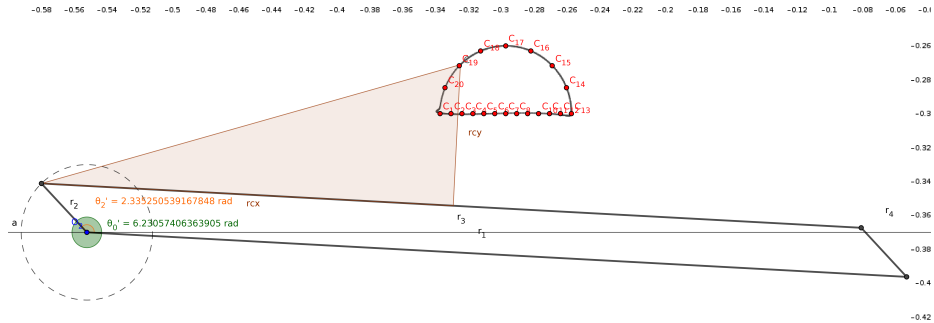


Fig. 3. Mecanismo óptimo: Evolución diferencial.

$$\begin{aligned}
 g_1(p) &= l_2 + l_3 - l_1 - l_4 \leq 0 \\
 g_2(p) &= l_2 - l_3 \leq 0 \\
 g_3(p) &= l_2 - l_4 \leq 0 \\
 g_4(p) &= l_2 - l_1 \leq 0 \\
 g_5(p) &= l_1 - l_3 \leq 0 \\
 g_6(p) &= l_4 - l_3 \leq 0 \\
 g_7(p) &= \theta_2^{10} - \theta_2^{11} \leq 0 \\
 g_8(p) &= \theta_2^{11} - \theta_2^{12} \leq 0 \\
 g_9(p) &= \theta_2^{12} - \theta_2^{13} \leq 0 \\
 &\vdots \\
 g_{25}(p) &= \theta_2^{28} - \theta_2^{29} \leq 0,
 \end{aligned} \tag{13}$$

donde: los valores de los límites superior e inferior de las variables de diseño propuestos en [13] se observan en la Tabla 1.

5. Algoritmos de optimización

5.1. Búsqueda armónica

Algoritmo heurístico denominado HS (Harmony Search) funciona como el proceso de imitar la improvisación en una interpretación musical [10]. Cuando un músico está improvisando, éste realiza una de las siguientes acciones [3]:

1. Toca alguna melodía conocida que ha aprendido anteriormente.
2. Toca algo parecido a la melodía anteriormente mencionada, ajustándola poco a poco al tono deseado.
3. Compone una nueva melodía basándose en sus conocimientos musicales para seleccionar nuevas notas aleatoriamente.

Tabla 2. Vector de variables de diseño: Búsqueda armónica.

No.	l_1	l_2	l_3	l_4	r_x	r_y	θ_0	x_0	y_0	θ_1^j
1	0.473220235	0.032348431	0.499998021	0.16732344	0.190263529	0.277719669	2.623727398	0.513412554	-0.044473163	1.456826347
2	0.357548313	0.02766025	0.499997377	0.215078399	0.398151669	0.4845414	0.096656775	-0.504103062	-0.897136203	4.875237892
3	0.230598661	0.024717261	0.23555831	0.224274507	0.08147685	0.479662013	5.18714452	-0.132433173	-0.754448593	5.843051441
4	0.473785354	0.044503366	0.499950923	0.234468952	0.313458259	0.097885115	5.936458913	-0.427542003	-0.42356719	4.286482092
5	0.499976136	0.040223415	0.499990628	0.471766814	0.388987613	0.496087111	2.171576131	0.042798361	0.216308851	2.58386484
6	0.49998144	0.031916127	0.499998914	0.375044365	0.224940904	0.343949447	5.371381307	0.055924119	-0.592084039	5.091658307
7	0.310054364	0.024934815	0.499996165	0.280420971	0.361783114	0.488451561	6.193476265	0.043113645	-0.878503522	4.753038661
8	0.4837542	0.025588016	0.483763782	0.456918592	0.01755792	0.300690404	1.63542402	-0.01072323	-0.031700189	2.303140839
9	0.2965412	0.034737921	0.49729193	0.336792048	0.468449808	0.499950675	3.089174274	-0.255573413	0.406320105	1.484799349
10	0.499985135	0.030308522	0.499996356	0.353298103	0.178549775	0.369635285	5.32047189	-0.341865792	-0.599641824	5.145381557
11	0.332377987	0.030267334	0.499991727	0.398256894	0.36469443	0.461863265	2.719237038	0.169837093	0.298938252	1.685892377
12	0.497270992	0.042077503	0.499997974	0.045802669	0.287464472	0.1577466	0.009477661	0.173080739	-0.466452908	4.357711398
13	0.469612045	0.028317258	0.483601031	0.247263493	0.094346999	0.382458402	2.26227848	0.00951829	0.041765236	1.903460226
14	0.393626859	0.031227668	0.499993407	0.239235876	0.307411243	0.311213369	6.025917772	-0.675472455	-0.657442267	4.798831606
15	0.375849593	0.030099818	0.499998361	0.273443904	0.354690807	0.483041976	2.865752793	0.073499166	0.279417963	1.950012352
16	0.483090594	0.025826762	0.483112561	0.48310485	0.035473685	0.27989311	1.675524479	-0.245971882	-0.042414933	2.230121516
17	0.428323133	0.029332525	0.499994217	0.310189384	0.267991793	0.482605648	5.641375033	-0.216514115	-0.772925214	5.00337375
18	0.371102184	0.027736338	0.495439776	0.439147904	0.320373143	0.497853853	5.542765985	-0.553723073	-0.851294561	5.448851125
19	0.465130746	0.033363127	0.499994785	0.179088721	0.244353186	0.361411395	5.775649223	-0.426001761	-0.605508409	4.917821651
20	0.420306478	0.031498786	0.499996869	0.46608843	0.341338075	0.496129164	5.30743682	-0.591383695	-0.801402614	5.883065375
No.	θ_2^j	θ_3^j	θ_4^j	θ_5^j	θ_6^j	θ_7^j	θ_8^j	θ_9^j	θ_{10}^j	Función Objetivo
1	1.607717607	1.764846423	1.934375266	2.117689894	2.262306383	2.459292729	2.625418371	2.804695009	2.978571167	3.160961162
2	4.986551868	5.084339723	5.161516226	5.269341592	5.35435459	5.425912562	5.511801533	5.605582008	5.687173312	5.760303503
3	5.944685999	6.042169371	6.129622554	6.216072086	6.3020549953	6.38804296	6.47424476	6.56020541	6.646404945	6.732612306
4	4.464122331	4.634475545	4.803509498	4.952514241	5.085347146	5.227752314	5.351277061	5.47424548	5.62178953	5.751516023
5	2.69346462	2.776273301	2.877600243	2.971184255	3.065702956	3.154802705	3.253766342	3.3534978	3.445906324	3.548448558
6	5.227046836	5.364957845	5.518565952	5.687172744	5.786436907	5.96744162	6.090860924	6.23093001	6.37998273	6.5272556
7	4.884295421	4.987031505	5.115591588	5.195468915	5.289281561	5.387204733	5.478547842	5.556862753	5.648263794	5.738904836
8	2.42864774	2.585007506	2.743755919	2.930284764	3.120581137	3.30629435	3.47309067	3.673021926	3.885153044	4.076182355
9	1.59314161	1.694740506	1.79267986	1.882158671	1.964565214	2.047771093	2.126656724	2.204129584	2.280691235	2.352518368
10	5.297438174	5.455774426	5.611597744	5.76347223	5.879626523	6.054869053	6.179460423	6.325758811	6.480319957	6.630751179
11	1.855386231	1.991431885	2.106878151	2.241285338	2.354136077	2.46804048	2.567727743	2.667035782	2.768793422	2.869509054
12	4.576514075	4.743145545	4.895886088	5.032665145	5.165114568	5.294098039	5.416215053	5.546107989	5.671667199	5.806669348
13	2.068673264	2.231424123	2.396993586	2.556034623	2.72246835	2.907700134	3.040381667	3.232188203	3.38641017	3.561889223
14	4.966077441	5.113288874	5.232510454	5.347799643	5.46673551	5.594361322	5.716884158	5.838792731	5.938535388	6.089385015
15	2.067652804	2.185626752	2.303194403	2.41565523	2.530228449	2.643890146	2.753969416	2.881619114	2.985051273	3.09193226
16	2.389292194	2.512769504	2.695734335	2.865828736	3.047446126	3.26102275	3.433428978	3.615914036	3.826574703	4.025430253
17	5.157258757	5.304305427	5.434447373	5.567591294	5.683668606	5.79770865	5.922697146	6.023900029	6.13543603	6.26518001
18	5.569968813	5.690830788	5.805911937	5.91084514	6.017814681	6.11654951	6.215508349	6.304755821	6.39436277	6.484350331
19	5.067424773	5.202492345	5.329799988	5.454606196	5.577412211	5.709979159	5.838178798	5.96778277	6.11598348	6.254414149
20	5.986167038	6.084631884	6.191265897	6.30029103	6.4087329375	6.5183114279	6.6292198116	6.743863967	6.861545166	6.934254361

Descripción del algoritmo HS El algoritmo HS se relaciona a un vector de variables de diseño $X^j = (x_1, x_2, \dots, x_N)$, con armonías, llamado vector armónico donde $j = 1, 2, \dots, k$.

- **Paso 1. Inicializa una memoria armónica (MA).** La memoria armónica (MA) $X = (x_1, x_2, \dots, x_N)$ se genera con una distribución uniforme considerando $L_i < x_i^j < U_i$, donde L_i y U_i son los límites inferior y superior respectivamente del problema dado, con $i = 1, 2, \dots, N$ resultando (14).

$$MA = \begin{bmatrix} X^1 \\ X^2 \\ \vdots \\ X^k \end{bmatrix} = \begin{bmatrix} x_1^1 & x_2^1 & \dots & x_N^1 \\ x_1^2 & x_2^2 & \dots & x_N^2 \\ \vdots & \vdots & \vdots & \vdots \\ x_1^k & x_2^k & \dots & x_N^k \end{bmatrix}, \quad (14)$$

- **Paso 2. Improvisa una nueva armonía.** Se genera un nuevo vector armónico $X_{new} = (x_1^{new}, x_2^{new}, x_3^{new}, \dots, x_N^{new})$ estimando dos probabilidades, $r_{accept} \in [0, 1]$ y $r_{pa} \in [0, 1]$. Con probabilidad r_{accept} se elige de manera aleatoria un índice j de la memoria armónica, y con probabilidad $(1 - r_{accept})$ se elige $x_i^{new} = rand$, que es lo que se interpreta como improvisación.
- **Paso 3. Ajuste de armonía.** Con probabilidad $r_{accept}r_{pa}$ se asigna $x_i^{new} = x_i^k + (rand)(bw)$ donde $x_i^k \in MA$, $rand \in [0, 1]$ y bw es un ancho de banda que define el rango de ajuste. Si no se cumple, es decir, con probabilidad $1 - r_{accept}r_{pa}$ se asigna $x_i^{new} = MA(j, i)$ (j es el índice elegido en Paso 2).
- **Paso 4. Actualizar memoria armónica.** Si la nueva armonía X_{new} produce un mejor desempeño que la peor armonía de la MA entonces se sustituye actualizándola por la mejor.
- **Paso 5. Si el criterio de paro no se cumple, ir al paso 2.** Terminar cuando el número máximo de iteraciones o ciclos se alcanza.

Los Pasos 1 a 5 se plasman en el pseudocódigo de HS que se muestra en el Algoritmo 1.1 [22].

5.2. Evolución Diferencial

Rainer Storn y Kenneth Price [16] desarrollaron un algoritmo denominado Evolución Diferencial (ED), este algoritmo heurístico es un optimizador basado en la población que ataca el problema de punto de partida al muestrear la función objetivo en múltiples puntos iniciales elegidos al azar [14]. Las fases principales de este método son:

- **Inicialización.** Antes de iniciar la población, los límites inferior b_L y superior b_U deberán ser definidos, los valores de los parámetros de cada vector son generados aleatoriamente dentro del rango determinado por los límites, la ecuación (15) representa el vector resultante:

$$x_{j,i} = b_{j,L} + (b_{j,U} - b_{j,L}) * rand(0, 1), \quad (15)$$

Tabla 3. Vector de variables de diseño: Evolución diferencial.

No.	l_1	l_2	l_3	l_4	$r_{c,x}$	$r_{c,y}$	θ_0	x_0	y_0	θ_2^j
1	0.034142014	0.018956686	0.244068043	0.244066686	1.12E-005	0.041842632	4.739602097	-0.02425582	-0.321903669	5.382583148
2	0.035034132	0.019253992	0.274117666	0.272961319	0.007617314	0.04163331	4.899402154	0.091365231	-0.322295495	5.289971734
3	0.499999714	0.039985917	0.499999856	0.040072264	0.251180883	0.083434351	6.230574064	-0.552043577	-0.370109899	3.833220462
4	0.499953067	0.031717969	0.499971043	0.222800686	0.152174955	0.311277952	2.434437642	0.004610382	-0.022015981	1.601552279
5	0.499978677	0.023913189	0.499983988	0.499875478	1.70E-005	0.405176328	1.725229419	0.227884289	0.094072265	2.296687525
6	0.49999255	0.040684817	0.499995741	0.464837779	0.244670596	9.08E-006	2.502653207	-0.111789446	-0.204735548	1.384106147
7	0.368515424	0.024765042	0.439357141	0.407845983	0.060759876	0.265226434	5.052813156	-0.039845695	-0.532671818	5.13566821
8	0.499998502	0.024314165	0.499999078	0.456900937	3.49E-007	0.40873178	1.806454021	0.158854924	0.09485197	2.220446799
9	0.034105487	0.018914748	0.49995119	0.499950877	0.00085192	0.041957769	4.739493295	-0.299491132	-0.322072647	5.3635793
10	0.322063303	0.033613983	0.487821043	0.199531914	0.139716962	0.001743678	0.407298792	0.348304743	-0.372413513	3.528064717
11	0.235945839	0.035519065	0.318050349	0.143897238	0.118333897	0.013858241	6.263887649	-0.025322357	-0.343447153	3.949324787
12	0.499999833	0.024421757	0.499999903	0.445291904	2.04E-007	0.409583712	1.827027701	-0.140052662	0.094790498	2.201851158
13	0.49999246	0.040449377	0.499992467	0.040449389	0.254599188	0.007361777	6.233123544	-0.182546532	-0.295255027	3.393875969
14	0.046155456	0.02256725	0.148476876	0.14633798	0.027645374	0.037647777	5.23041022	-0.226567564	-0.325228906	5.140294024
15	0.034162511	0.019012473	0.195128149	0.195127121	3.35E-007	0.041639174	4.745959456	0.352541639	-0.321662589	5.387867272
16	0.036821205	0.019146952	0.142154894	0.131214552	3.09E-008	0.043362188	5.108502653	-0.115634122	-0.323446885	5.057664382
17	0.034108297	0.018910508	0.499991787	0.499247066	0.000503445	0.041991901	4.758910777	0.415502362	-0.322108067	5.340671486
18	0.034810702	0.019289841	0.243698431	0.241434758	0.007720081	0.041174542	4.941812098	-0.311237174	-0.321814721	5.250258414
19	0.388224787	0.039540018	0.499723838	0.251127152	0.223948615	0.000397852	3.118042566	-0.026365804	-0.177493457	0.887434623
20	0.499993052	0.023992409	0.499998248	0.492083745	3.27E-006	0.405498331	1.742728438	0.124460426	0.094308566	2.278124803

No.	θ_2^0	θ_2^1	θ_2^2	θ_2^3	θ_2^4	θ_2^5	θ_2^6	θ_2^7	θ_2^8	θ_2^9
1	5.640197718	5.851147784	5.991150465	6.103158596	6.203224734	0.014961	0.110348896	0.212100507	0.328049867	0.47399157
2	5.539483856	5.766956905	5.916527346	6.033254089	6.137818974	6.234532501	0.047181829	0.147219364	0.258111995	0.396113039
3	4.185234302	4.421799229	4.617049849	4.790975919	4.953012095	5.10873079	5.262521279	5.418735513	5.582105245	5.760065211
4	1.758116319	1.928843477	2.105021364	2.277704315	2.45280056	2.623996403	2.795340315	2.966245946	3.137428128	3.311076114
5	2.455548119	2.626666308	2.799846283	2.972115857	3.141636502	3.310857938	3.480255964	3.649599261	3.820626639	3.994651935
6	1.535529799	1.698158642	1.862220061	2.027337835	2.193073906	2.359350685	2.52648288	2.697773508	2.870369375	3.044419929
7	5.381353287	5.534626327	5.718653512	5.907749386	6.033883845	6.224882878	0.11715169	0.309926988	0.454854601	0.645566709
8	2.378901513	2.550328894	2.724604673	2.897522863	3.068069291	3.237249032	3.406268239	3.576020808	3.747669108	3.920459692
9	5.620276772	5.835200372	5.977685074	6.091206152	6.191577612	0.003760792	0.099122288	0.200072085	0.314359564	0.458461825
10	3.691254204	3.921719905	4.10261746	4.306832065	4.478011023	4.65916937	4.826120389	4.994169648	5.152406774	5.299278901
11	4.09611077	4.259166508	4.530483762	4.748559787	4.846437684	5.038752531	5.154767781	5.310792683	5.631949696	5.769781952
12	2.360107168	2.531534929	2.705762236	2.878685506	3.049401208	3.218667627	3.387685414	3.557601698	3.729245476	3.902150457
13	3.8281786	4.089447856	4.300957541	4.488419186	4.660938501	4.827252193	4.990832209	5.157644636	5.332479421	5.525669436
14	5.333868992	5.607060788	5.745506369	5.906958485	6.012576368	6.141882983	6.2399837	0.077939764	0.189984277	0.312387246
15	5.647822264	5.854540893	5.99455367	6.107772512	6.207014004	0.018219994	0.114614331	0.215824181	0.332100138	0.478557465
16	5.306911319	5.539892459	5.698203899	5.821691737	5.92807827	6.025766141	6.120056469	6.215782896	0.03650184	0.162327524
17	5.596945022	5.813434062	5.957158731	6.071342741	6.172200186	6.267636667	0.079746832	0.180223331	0.293733416	0.436596319
18	5.504570049	5.732870608	5.887361949	6.004502353	6.108103172	6.205539212	0.015660749	0.114759546	0.22480919	0.362605237
19	1.026704518	1.173252345	1.334264093	1.490645599	1.654929747	1.809212495	1.980886517	2.147998093	2.32018889	2.483438777
20	2.436219528	2.607985224	2.782113716	2.954355657	3.125088647	3.294052093	3.462639452	3.632820241	3.804135771	3.977013101

donde $\text{rand}(0, 1)$ retorna un número aleatorio distribuido uniformemente dentro del rango $[0, 1)$ y j indica un nuevo valor aleatorio para cada parámetro.

- **Mutación:** ED muta y recombina la población para producir un solo vector hijo (trial), eligiendo vectores al azar $(X_{r_0,g}, X_{r_1,g}, X_{r_2,g})$, indicando el factor escala $F \in (0, 1)$ para obtener un vector de mutación (16).

$$V_{i,g} = X_{r_0,g} + F * (X_{r_1,g} - X_{r_2,g}). \tag{16}$$

- **Cruza:** Se construye un vector hijo $U_{i,g}$ (17) entre el vector padre $X_{i,g}$ y el vector de mutación $V_{i,g}$.

$$U_{i,g} = u_{j,i,g} \begin{cases} V_{j,i,g} & \text{si } (\text{rand}_j(0, 1)) \leq Cr \\ X_{j,i,g} & \text{en otro caso.} \end{cases} \quad \text{o } j = j_{\text{rand}}, \tag{17}$$

El factor de cruza $Cr \in [0, 1]$ es una probabilidad predefinida, la cual controla la fracción de parámetros que se heredan. Para garantizar que el hijo no sea un duplicado del padre se elige un número aleatorio, si éste es menor que Cr , el parámetro se hereda del vector de mutación $V_{i,g}$, en cambio éste es heredado por el padre $X_{i,g}$.

- **Selección:** El operador de selección decide que individuo es aceptado (18), si el hijo $U_{i,g}$ tiene un desempeño igual o inferior al padre $X_{i,g}$, este último es reemplazado, de lo contrario el padre conserva su lugar, ambos por al menos una generación más.

$$X_{i,g+1} = \begin{cases} U_{i,g} & \text{si } f(U_{i,g}) \leq f(X_{i,g}), \\ X_{i,g} & \text{en otro caso.} \end{cases} \quad (18)$$

Producida la nueva población, el procedimiento de mutación, recombinación y selección se vuelve iterativo hasta localizar el óptimo o cumplir el máximo de generaciones definidas por el usuario. El pseudocódigo de ED se muestra en el Algoritmo 1.2 [16].

6. Resultados

6.1. Desempeño de Búsqueda Armónica

Los resultados obtenidos se muestran en la Tabla 2, éstos resultados corresponden a los 20 mejores vectores de las variables de diseño alcanzados con el algoritmo de Búsqueda Armónica. Con el propósito de visualizar el valor correspondiente de la función objetivo, a la Tabla 2, se le ha agregado el valor de desempeño para cada vector de diseño representado en un campo como **Función Objetivo**. El vector de diseño resultante que contiene la mejor solución (mínimo valor obtenido de la función objetivo) se resalta en color amarillo. El vector solución con mejor desempeño por HS cuyo valor óptimo en la función objetivo corresponde a:

$$p = \{0,497270992, 0,042077503, 0,49999794, 0,045802669, 0,287464472, \\ 0,1577466, 0,009477661, 0,173080739, -0,466452908, 4,357711398, \\ 4,576514075, 4,743145545, 4,895886088, 5,032665145, 5,165114568, \\ 5,294098039, 5,416215053, 5,546107989, 5,671667199, 5,806669348, \\ 5,955595615, 6,128386729, 0,525950669, 0,93680096, 1,296347928, \\ 1,636128364, 1,9778168, 2,353431238, 2,743588304\}, \quad (19)$$

$$\text{mín}(F.O.HS) = 3,24685 \times 10^{-05}. \quad (20)$$

A partir del óptimo obtenido y su vector correspondiente (ecuación 20 y 19) se realiza la simulación mostrada en la Figura 2.

6.2. Desempeño evolución diferencial

Las Tablas 3 y su continuación que es la Tabla 4 indican los resultados alcanzados con el algoritmo de Evolución Diferencial, se puede visualizar el valor óptimo

Tabla 4. Dimensión óptima de las 4 barras del mecanismo

Algoritmo	l_1	l_2	l_3	l_4	r_{cx}	r_{cy}
Búsqueda Armónica	0.497270992	0.042077503	0.49999794	0.045802669	0.287464472	0.1577466
Evolución Diferencial	0.499999714	0.039985917	0.499999856	0.040072264	0.251180883	0.083434351
Algoritmo	θ_0	x_0	y_0	θ_2^1	θ_2^2	θ_2^3
Búsqueda Armónica	0.009477661	0.173080739	-0.466452908	4.357711398	4.576514075	4.743145545
Evolución Diferencial	6.230574064	-0.552043577	-0.370109899	3.833220462	4.185234302	4.421799229
Algoritmo	θ_2^4	θ_2^5	θ_2^6	θ_2^7	θ_2^8	θ_2^9
Búsqueda Armónica	4.895886088	5.032665145	5.165114568	5.294098039	5.416215053	5.546107989
Evolución Diferencial	4.617049849	4.790975919	4.953012095	5.10873079	5.262521279	5.418735513
Algoritmo	θ_2^0	θ_2^1	θ_2^2	θ_2^3	θ_2^4	θ_2^5
Búsqueda Armónica	5.671667199	5.806669348	5.955595615	6.128386729	6.259506669	6.36800096
Evolución Diferencial	5.582105245	5.760065211	5.968818105	6.03176862	6.443501996	6.838253583
Algoritmo	θ_2^6	θ_2^7	θ_2^8	θ_2^9	θ_2^0	Función Objetivo
Búsqueda Armónica	1.296347928	1.636128364	1.9778168	2.353431238	2.743588304	3.24685E-05
Evolución Diferencial	1.232165409	1.62540309	2.018496404	2.41125571	2.805075923	1.29E-06

alcanzado en la fila 3, donde la Función Objetivo es $1,29E^{-6}$. El valor óptimo que se obtuvo (22) y el vector con las 29 variables de diseño dado en (21) marcado en las Tablas 3 y 4 de color amarillo, obedece a:

$$\begin{aligned}
 p = \{ & 0,499999714, 0,039985917, 0,499999856, 0,040072264, 0,251180883, \\
 & 0,083434351, 6,230574064, -0,552043577, -0,370109899, 3,833220462, \\
 & 4,185234302, 4,421799229, 4,617049849, 4,790975919, 4,953012095, \\
 & 5,10873079, 5,262521279, 5,418735513, 5,582105245, 5,760065211, \\
 & 5,968818105, 0,03176862, 0,443501996, 0,838253583, 1,232165409, \\
 & 1,62540309, 2,018496404, 2,41125571, 2,805075923 \} ,
 \end{aligned}
 \tag{21}$$

$$\min(F.O._{ED}) = 1,29x10^{-06}.
 \tag{22}$$

La simulación del caso de estudio del óptimo obtenido con ED se muestra en la Figura 3. En resumen la Tabla 4 muestra los valores del ambos sistemas óptimos obtenidos resultado de la implementación de cada algoritmo. Como se puede apreciar el valor óptimo fue obtenido con Evolución diferencial dado por $1,29E - 06$. Cabe mencionar que las muestras presentadas en la Tabla 2, para Búsqueda Armónica solo el 5% cumple con condición de que una solución factible satisfaga la tolerancia $\delta = 1E - 04$, mientras que para Evolución Diferencial (Tablas 3 y 4) éste porcentaje es de 50%.

7. Conclusión

El algoritmo de Evolución Diferencial para resolver problemas de optimización con restricciones produjo un mejor desempeño comparado con el algoritmo de Búsqueda Armónica. La síntesis dimensional del mecanismo de cuatro barras óptimo, que se genera a partir de los resultados, podemos calificarlo como satisfactorio debido a que, como se observa en la Figura 3 genera una trayectoria casi análoga del semicírculo planteado como caso de estudio.

Referencias

1. Askarzadeh, A., Rezazadeh, A.: Parameter identification for solar cell models using harmony search-based algorithms. *Solar Energy*, vol. 86, no. 11, pp. 3241–3249 (2017) doi: 10.1016/j.solener.2012.08.018
2. Bäck, T.: *Evolutionary algorithms in theory and practice: Evolution strategies, evolutionary programming, genetic algorithms*. Oxford University Press (1996)
3. Cobos, C., Pérez, J., Estupiñan, D.: Una revisión de la búsqueda armónica. *Revista Avances en sistemas e Informática*, vol. 8, no. 2, pp. 67-80 (2011)
4. Deb, K.: An efficient constraint handling method for genetic algorithms. *Computer Methods in Applied Mechanics and Engineering*, vol. 186, pp. 311–338 (2000) doi: 10.1016/S0045-7825(99)00389-8
5. Echevarría, L. C. and Orestes, S., and da-Silva, A. J.: Aplicación de los algoritmos evolución diferencial y colisión de partículas al diagnóstico de fallos en sistemas industriales. *Revista Investigación Operacional*, vol. 33, no. 2, pp. 160–172 (2012)
6. Foix, I., Cardona, S., Costa, D. C.: *Teoría de máquinas*. Universidad Politècnica de Catalunya, vol. 95 (2001)
7. Garro, B., Sossa, H., Vazquez, R. A.: Diseño automático de redes neuronales artificiales mediante el uso del algoritmo de evolución diferencial (ED). *Instituto Politécnico Nacional, Centro de Innovación y Desarrollo Tecnológico en Cómputo*, vol. 46, pp 13–27 (2012)
8. Guzmán-Gaspar, J. Y.: *Evolución diferencial para el control de un motor de corriente directa bajo incertidumbre paramétrica*. Tesis de maestría, Centro de enseñanza LANIA (2015)
9. Geem, Z., Choi, J. Y.: Music composition using harmony search algorithm. *Workshops on Applications of Evolutionary Computation*, pp. 593–600 (2007) doi:10.1007/978-3-540-71805-5_65
10. Geem, Z., Kim, J. and Loganathan, G.: A New heuristic optimization algorithm: Harmony search simulation. *Applied Soft Computing*, vol. 76, pp. 60–68 (2001) doi: 10.1177/003754970107600201
11. Geem, Z.: Optimal cost design of water distribution networks using harmony search. *Engineering Optimization*, vol. 38, no. 03, pp. 259–277 (2006) doi: 10.1080/030521505000467430
12. Myszka, D. H.: *Máquinas y mecanismos*. Pearson, 4ta edición (2012)
13. Pantoja-García, J.S., Villarreal-Cervantes, M.G., González-Robles, J.C., Cervantes, G. S., *Síntesis óptima de un mecanismo para la marcha bípeda utilizando evolución diferencial*, Elsevier, *Revista Internacional de Métodos Numéricos para Cálculo y Diseño en Ingeniería*, vol. 33, no. 1, pp 138–153 (2017)
14. Price, K., Storn, R., Lampinen, M., Jouni, A.: *Differential evolution: A practical approach to global optimization*. Springer Science and Business Media (2006)

15. Saka, M. P.: Optimum geometry design of geodesic domes using harmony search algorithm. *Advances in Structural Engineering*, vol. 10, no. 6, pp. 595–606 (2007) doi: 10.1260/136943307783571445
16. Storn, R. and Price, K., Differential evolution: A simple and efficient adaptive scheme for global optimization over continuous spaces. *International Computer Science Institute*, vol. 11, pp. 341–359 (1995)
17. Vázquez, K. R.: Ant colony optimization. *Genetic Programming and Evolvable Machines*, vol. 6 (2005) doi: 10.1007/s10710-005-2991-z
18. Vega-Alvarado, E., Santiago-Valentín, E., Sánchez-Márquez, A., Solano-Palma, A., Portilla-Flores, E. A., Flores-Pulido, L., Síntesis óptima de un mecanismo plano para seguimiento de trayectoria utilizando evolución diferencial. *Research in Computing Science*, Vol. 72, pp. 85–98 (2014)
19. Villate-Gil, A., Rincón-Arandia, D. E., Melgarejo-Rey, M. A.: Evolución diferencial aplicada a la sintonización de clasificadores difusos para el reconocimiento del lenguaje de señas. *Ingeniería y Universidad*, vol. 16, no. 2, pp. 397–413 (2012)
20. Yang, X. S.: Engineering optimizations via nature-inspired virtual bee algorithms. *Artificial Intelligence and Knowledge Engineering Applications: A Bioinspired Approach. IWINAC 2005. Lecture Notes in Computer Science*, vol. 3562, pp. 317–323 (2005) doi: 10.1007/11499305_33, pp. 317-323
21. Yang, X.: Firefly algorithms for multimodal optimization. *International symposium on stochastic algorithms*, pp. 169–178 (2009) doi: 10.1007/978-3-642-04944-6_14
22. Yang, X.: Harmony search as a metaheuristic algorithm. *Music-inspired harmony search algorithm: theory and applications*, pp. 1–14 (2009) doi: 10.1007/978-3-642-00185-7_1