





Article

A New Study on Optimization of Four-Bar Mechanisms Based on a Hybrid-Combined Differential Evolution and Jaya Algorithm

Sy Nguyen-Van ¹, Qui X. Lieu ^{2,3}, Nguyen Xuan-Mung ^{4,*} and Thi Thanh Nga Nguyen ^{1,*}

¹ Faculty of Mechanical Engineering, Thai Nguyen University of Technology, 3-2 Street, Thai Nguyen City 250000, Vietnam; vansy@tnut.edu.vn

² Faculty of Civil Engineering, Ho Chi Minh City University of Technology (HCMUT), 268 Ly Thuong Kiet Street, Ward 14, District 10, Ho Chi Minh City 700000, Vietnam; lieuxuanqui@hcmut.edu.vn

³ Vietnam National University Ho Chi Minh City (VNU-HCM), Linh Trung Ward, Thu Duc District, Ho Chi Minh City 700000, Vietnam

⁴ Faculty of Mechanical and Aerospace Engineering, Sejong University, Seoul 05006, Korea

* Correspondence: xuanmung@sejong.ac.kr (N.X.-M.); nguyennga@tnut.edu.vn (T.T.N.N.)

Abstract: In mechanism design with symmetrical or asymmetrical motions, obtaining high precision of the input path given by working requirements of mechanisms can be a challenge for dimensional optimization. This study proposed a novel hybrid-combined differential evolution (DE) and Jaya algorithm for the dimensional synthesis of four-bar mechanisms with symmetrical motions, called HCDJ. The suggested algorithm uses modified initialization, a hybrid-combined mutation between the classical DE and Jaya algorithm, and the elitist selection. The modified initialization allows generating initial individuals, which are satisfied with Grashof's condition and consequential constraints. In the hybrid-combined mutation, three differential groups of mutations are combined. DE/best/1 and DE/best/2, DE/current to best/1 and Jaya operator, and DE/rand/1, and DE/rand/2 belong to the first, second, and third groups, respectively. In the second group, DE/current to best/1 is hybrid with the Jaya operator. Additionally, the elitist selection is also applied in HCDJ to find the best solutions for the next generation. To validate the feasibility of HCDJ, the numerical examples of the symmetrical motion of four-bar mechanisms are investigated. From the results, the proposed algorithm can provide accurate optimal solutions that are better than the original DE and Jaya methods, and its solutions are even better than those of many other algorithms that are available in the literature.

Keywords: differential evolution (DE); Jaya algorithm; hybrid-combined mutation; hybrid-combined differential evolution and Jaya algorithm (HCDJ); dimensional synthesis of four-bar mechanisms.



Citation: Nguyen-Van, S.; Lieu, Q.X.; Xuan-Mung, N.; Nguyen, T.T.N. A New Study on Optimization of Four-Bar Mechanisms Based on a Hybrid-Combined Differential Evolution and Jaya Algorithm. *Symmetry* **2022**, *14*, 381. <https://doi.org/10.3390/sym14020381>

Academic Editor: Alexander Zaslavski

Received: 26 December 2021

Accepted: 2 February 2022

Published: 14 February 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Symmetrical motion mechanisms in which the forward and return strokes are the same movement have been applied in many mechanical systems. Four-bar mechanisms with symmetrical and asymmetrical motions are commonly used in mechanical devices such as sewing machines, round balers, and suspension systems of automobiles [1,2]. Achieving higher precision design of four-bar mechanisms, whose symmetrical motions satisfy the input path of a point in the coupling link, is still a challenge for the kinematic dimensional calculation. It results from the highly nonlinear objective function with many constraints.

Regarding methods for designing four-bar mechanisms, Zhang [3] proposed the graphical method, and Freudenstein [1] has been used in analysis to compute the kinematic dimensions of linkages. However, this method resulted in low precision and waste time [4]. In recent decades, random search algorithms such as differential evolution [2,5–8],

genetic algorithm [4,9], simulated annealing algorithm [10], and particle swarm optimization [2,11,12] have been used to solve kinematic dimensions of the mechanisms to increase accuracy in mechanism design.

The synthesis of four-bar mechanisms has been mentioned in recent years due to their wide applications in mechanical systems. Fernandez et al. [13] proposed the determination of kinematic dimension in order to minimize the objective function based on the dimensional constraint equation of mechanisms. The design of a four-bar mechanism used for the shadow robot is presented in [14]. Ramon et al. [15] applied the combination of difference evolution and local search algorithms for synthesis planer mechanisms, in which the four-bar mechanism is one of the examples in this work. These works, however, have limited the application cases of four-bar mechanisms. The teaching-learning based optimization algorithm has been used to determine the kinematic dimension of four-bar mechanisms [16,17]. In these researches, the signed timing has not been considered. Varedi-Koulai [18] synthesized four-bar mechanisms using graphical and analytical methods.

With random search algorithms, the Jaya algorithm is also a newly proposed approach [19]. This method has been applied for solving numerous optimization problems [20,21]. In order to obtain an optimal solution by using the Jaya algorithm, a high computational cost is essentially required. Thus, to improve its performance, the Jaya algorithm has been combined with other algorithms [5,22,23]. Up until now, the application of the Jaya algorithm in designing four-bar mechanisms is still limited [24].

One of the most well-known random search algorithms is DE algorithm which can be found in [25]. DE algorithm has been used to find the optimal solution for a lot of problems [26–28]. It is similar to the other random search, as an optimal solution is found by using the DE algorithm, which also requires a considerable computation cost. For this reason, several modifications of DE were proposed [29–35]. The combination of DE and other algorithms such as GA [36,37], PSO [38,39], and fireworks algorithm (FA) [40,41] was proposed. Furthermore, some additional modifications of DE were also proposed for path synthesis of four-bar mechanisms such as Cabrera [6], Ortiz [7], Lin [8]. Concretely, Cabrera [6] proposed a modified crossover to change the values of genes in the mutant and target vectors to achieve better values of objective functions for the next generation. Based on the work of Cabrera, Ortiz [7] suggested a tuning technique for the control parameters of F , CP , MP , and range to avoid the multiple executions of the algorithm until their proper values are found. Furthermore, Lin [7] proposed a new combined mutant operator of DE. In the combined mutation, $DE/best/1$, $DE/current\ to\ best/1$, and $DE/rand/1$ are used in the first $k\%$ ranking parents, middle-ranking parents (ranked between $(k\%, m\%)$), and other inferior parents, respectively.

From the discussions mentioned above, the investigation of a hybrid algorithm between DE and Jaya for improving the optimal dimensional synthesis of the four-bar mechanism has not yet been reported. Thus, this study proposes a novel hybrid-combined differential evolution and Jaya algorithm (called HCDJ) to fill the above research gap. The current hybrid algorithm combines DE and Jaya to improve the accuracy of optimal solutions. By combining the mutation operator DE with a Jaya operator, the global exploration ability of HCDJ is significantly enhanced, and thus the solution accuracy is also improved. Concretely, in the modified mutation stage, three groups of mutations are used in the first $k\%$, middle, and remaining populations, respectively. $DE/best/1$ and $DE/best/2$, $DE/current\ to\ best/1$ and Jaya operator, and $DE/rand/1$, and $DE/rand/2$ belong to the first, second, and third groups, respectively. Additionally, the refined initialization stage is applied to choose individuals that are satisfied with Grashof's condition and consequential constraints. Finally, the elitist selection technique is used in the selection phase to determine the best solutions for the next generation. To prove the effectiveness and robustness of the HCDJ algorithm in terms of accuracy, five numerical examples for the dimensional synthesis of four-bar mechanisms with symmetrical motions are performed, and outcomes obtained by the proposed methodology are compared with those of some available algorithms in the literature. The rest of this article is arranged as follows. Section 2

presents the optimization problem of four-bar mechanisms. In Section 3, a brief review of the classical DE and Jaya is presented firstly, and then a perspective scheme of the proposed HCDJ is discussed. In Section 4, five commonly examined numerical examples for the dimensional synthesis of the four-bar mechanisms are performed to validate the effectiveness and robustness of HCDJ. Then, results obtained in five cases are discussed in Section 5. Finally, some conclusions are provided in Section 6.

2. Dimensional Synthesis of the Four-Bar Mechanisms

In this study, we focus on the dimensional synthesis of the four-bar mechanisms and the optimization problem, which is used to determine the kinematic dimensions of linkages and positions of pin-joints denoted by O_2 and O_4 (see Figure 1) by giving the input path of the coupler point C in link 3. C^i and C_d^i are set as target points indicated by the input point and the designed point, respectively, i.e., $C^i = [C_X^i, C_Y^i]$ and $C_d^i = [C_{Xd}^i, C_{Yd}^i]$. Therefore, the objective function can be written as follows [4]:

$$Error(\mathbf{X}) = \sum_{i=1}^N (C_{Xd}^i - C_X^i)^2 + (C_{Yd}^i - C_Y^i)^2, \tag{1}$$

where N is the number of points in the path of the coupler, and \mathbf{X} is preventative for the design variables' vector characterized by kinematic dimensions and positions of the linkages and can be expressed as

$$\mathbf{X} = [r_1 \ r_2 \ r_3 \ r_4 \ a \ b \ x_{02} \ y_{02} \ \theta_1 \ \theta_2^1 \ \theta_2^2 \ \dots \ \theta_2^n], \tag{2}$$

where $r_1, r_2, r_3,$ and r_4 denote the kinematic dimensions of links 1, 2, 3, and 4, respectively; $a, b, x_{02}, y_{02},$ and θ_1 are shown in Figure 1; $\theta_2^1, \theta_2^2, \dots, \theta_2^n$ are preventative for the angle positions of link 2 at input positions of coupler C, respectively.

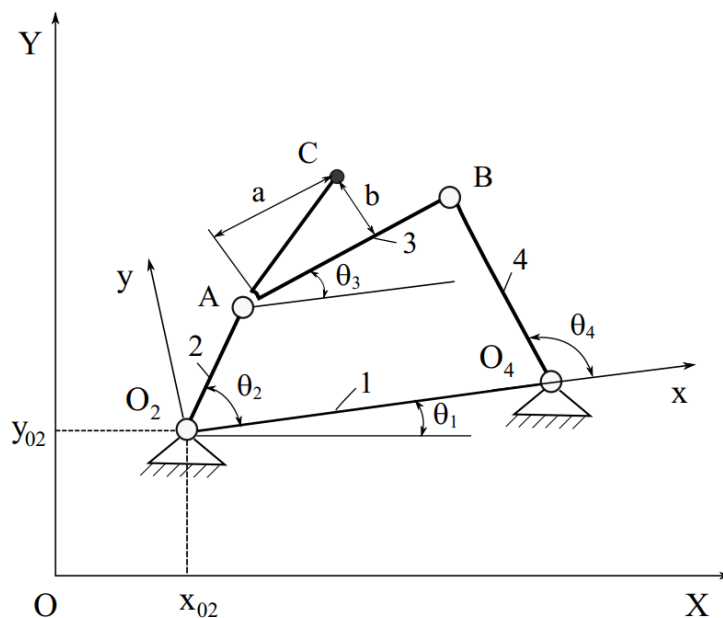


Figure 1. Kinematic diagram of four-bar mechanisms.

In Equation (1), the position of the coupler C in the reference frame O_2xy can be computed based on the closure loop as found in [5]. It can be calculated as

$$\begin{cases} C_x = r_2 \cos \theta_2 + a \cos \theta_3 - b \sin \theta_3, \\ C_y = r_2 \sin \theta_2 + a \sin \theta_3 + b \cos \theta_3. \end{cases} \tag{3}$$

The angle positions of links 3, i.e., θ_3 can be computed as

$$\theta_3 = 2 \tan^{-1}\left(\frac{-H \pm \sqrt{H^2 - 4GJ}}{2G}\right), \tag{4}$$

where G, H and J are calculated as the following equations

$$\begin{cases} G = \cos \theta_2 - K_1 + K_2 \cos \theta_2 + K_3, \\ H = -2 \sin \theta_2, \\ J = K_1 + (K_2 - 1) \cos \theta_2 + K_3, \end{cases} \tag{5}$$

in which,

$$K_1 = \frac{r_1}{r_2}; \quad K_2 = \frac{r_1}{r_3}; \quad K_3 = \frac{r_4^2 - r_3^2 - r_2^2 - r_1^2}{2r_2r_3}. \tag{6}$$

The position of the point C in the global coordinate OXY , as shown in Figure 1, can be simply computed as

$$\begin{bmatrix} C_X \\ C_Y \end{bmatrix} = \begin{bmatrix} \cos \theta_1 & -\sin \theta_1 \\ \sin \theta_1 & \cos \theta_1 \end{bmatrix} \begin{bmatrix} C_x \\ C_y \end{bmatrix} + \begin{bmatrix} x_{02} \\ y_{02} \end{bmatrix}. \tag{7}$$

It should be noted that the i^{th} design variable (\mathbf{X}_i) is in the range of $[\mathbf{X}_i^{min}, \mathbf{X}_i^{max}]$ with its upper bound of \mathbf{X}_i^{min} and lower bound of \mathbf{X}_i^{max} .

In addition to the constraint on the design variable, there are also two more constraints, namely the Grashof’s condition and the sequence of input angles. These constraints can be displayed as follows.

The Grashof’s condition allows the mechanism to have an entire ration link that is connected with the frame (link 1); this condition is denoted by $h_1(\mathbf{X})$. If Grashof’s condition is true, $h_1(\mathbf{X}) = 0$, in contrast $h_1(\mathbf{X}) = 1$. Mathematically, this constraint could be given as

$$s + l \leq p + q \tag{8}$$

in which, $[s, l, p, q] \in [r_1, r_2, r_3, r_4]$; s and l denote the shortest and longest lengths, respectively; p and q are other lengths.

The constraint for the sequence of input angles is that the angle-position values of link 2 are in sequence; this condition is denoted by $h_2(\mathbf{X})$. If this condition of input angles is true, $h_2(\mathbf{X}) = 0$, in contrast $h_2(\mathbf{X}) = 1$. Mathematically, this constraint is expressed as

$$\theta_2^m > \theta_2^{mod(m+1,Z)} > \dots > \theta_2^{mod(m+Z,Z)} \tag{9}$$

in which, θ_2^m is equal to $\min\{\theta_2^n\}$; θ_2^n is the value of θ_2 in its n^{th} position; Z is the number of input angles and $mod(n, m)$ is the remainder of the quotient of n/m .

These conditions need to be put into the objective function; thus, the objective function in Equation (1) can be rewritten as

$$f(\mathbf{X}) = Error(\mathbf{X}) + \epsilon_1 [h_1(\mathbf{X}) + h_2(\mathbf{X})], \tag{10}$$

where ϵ_1 is the penalty constant.

3. Optimization Algorithm

3.1. Differential Evolution Algorithm

In each generation of the DE algorithm, there are four main stages, which are initialization, mutation, crossover, and selection [25].

- *Initialization*

For each optimization problem, NP is presentative of the number of individuals in the population and its initial values are randomly selected in a predefined continuous

search space. In which, each i^{th} individual ($i = 1, 2, \dots, NP$) is a vector of design variables denoted by D and is given by

$$\mathbf{X}_{i,j}^{g=0} = \mathbf{L}_j + rand_{i,j}[0,1](\mathbf{H}_j - \mathbf{L}_j), \quad j = 1, 2, \dots, D, \tag{11}$$

where \mathbf{L}_j and \mathbf{H} present the lower and upper boundary vectors; NP denotes the number of individuals in the population; D characterizes the number of design variables; $rand[0, 1]$ creates a random value within 0 and 1. The superscripts, i.e., ($g = 0$) and g show the initial and current iterations, respectively. The i^{th} individual vector with ($g = 0, 1, \dots, g_{max}$) can be written as

$$\mathbf{X}_i^g = \{ \mathbf{X}_{i,1}^g, \mathbf{X}_{i,2}^g, \dots, \mathbf{X}_{i,j}^g, \dots, \mathbf{X}_{i,D}^g \}. \tag{12}$$

- *Mutation*

After the first stage, to increase the variety of the entire population, a mutant vector \mathbf{V}_i^g is generated from the target vectors by using a mutation. The four most frequently used mutation schemes are “DE/rand/1”, “DE/rand/2”, “DE/best/1” and “DE/best/2”. Two former DE operators have a good ability in terms of the global search but their convergence speed is slow. In contrast, two latter DE operators have a good local searching ability, yet obtained solutions may be trapped into local optima [42]. These four DE/operators are shown as

$$rand/1: \quad \mathbf{V}_i^g = \mathbf{X}_{R_1}^g + F(\mathbf{X}_{R_2}^g - \mathbf{X}_{R_3}^g), \tag{13a}$$

$$best/1: \quad \mathbf{V}_i^g = \mathbf{X}_{best}^g + F(\mathbf{X}_{R_1}^g - \mathbf{X}_{R_2}^g), \tag{13b}$$

$$rand/2: \quad \mathbf{V}_i^g = \mathbf{X}_{R_1}^g + F(\mathbf{X}_{R_2}^g - \mathbf{X}_{R_3}^g) + F(\mathbf{X}_{R_4}^g - \mathbf{X}_{R_5}^g), \tag{13c}$$

$$best/2: \quad \mathbf{V}_i^g = \mathbf{X}_{best}^g + F(\mathbf{X}_{R_1}^g - \mathbf{X}_{R_2}^g) + F(\mathbf{X}_{R_3}^g - \mathbf{X}_{R_4}^g), \tag{13d}$$

in which R_1, R_2, R_3, R_4 and R_5 are differential numbers selected in $[1, 2, 3, \dots, NP]$; F is a random number between 0 and 1, and \mathbf{X}_{best}^g is the best individual vector in the current iteration.

Owing to using such mutation schemes, the vector \mathbf{V}_i^g might be violated at its lower and higher bounds. Thus, for satisfying the boundary constraints, this vector is returned to its search space by using the following formulas

$$\mathbf{V}_{i,j}^g = \begin{cases} 2\mathbf{L}_j - \mathbf{V}_{i,j}^g & \text{if } \mathbf{V}_{i,j}^g < \mathbf{L}_j, \\ 2\mathbf{H}_j - \mathbf{V}_{i,j}^g & \text{if } \mathbf{V}_{i,j}^g > \mathbf{H}_j, \\ \mathbf{V}_{i,j}^g & \text{otherwise.} \end{cases} \tag{14}$$

- *Crossover*

To obtain good optimal solutions, the crossover stage is performed. The i^{th} trial vector $\mathbf{U}_i^g = \{ \mathbf{U}_{i,1}^g, \mathbf{U}_{i,2}^g, \dots, \mathbf{U}_{i,j}^g, \dots, \mathbf{U}_{i,d}^g \}$ is generated by mixing the target vector \mathbf{X}_i^g and the mutant vector \mathbf{V}_i^g as follows

$$\mathbf{U}_{i,j}^g = \begin{cases} \mathbf{V}_{i,j}^g & \text{if } j = Rand \quad \text{or} \quad rand_{i,j}[0,1] \leq Cr, \\ \mathbf{X}_{i,j}^g & \text{otherwise,} \end{cases} \tag{15}$$

in which $Rand$ is a random number chosen in $[1, NP]$, and the crossover value Cr is randomly selected in $[0.7, 1]$.

- *Selection*

This stage compares the trial individual, \mathbf{U}_i^g , with the target individual, \mathbf{X}_i^g , and then chooses the better ones for the next iteration based on their objective function values. This strategy can be expressed as

$$\mathbf{U}_i^{g+1} = \begin{cases} \mathbf{U}_i^g, & \text{if } f(\mathbf{U}_i^g) \leq f(\mathbf{X}_i^g), \\ \mathbf{X}_i^g, & \text{otherwise.} \end{cases} \quad (16)$$

3.2. Jaya Algorithm

The Jaya algorithm can be found in [19]. There are three steps to solve the optimization problem, which are: initial solutions, generating new solutions, and selection. The first and last steps of the Jaya algorithm are similar to the DE algorithm. In the generating new solutions, the best and worst solutions at g , denoted by \mathbf{X}_{best}^g and \mathbf{X}_{worst}^g , are created based on the tendency of one moving closer to success or reaching the best solution and trying to avoid failure or moving away from the worst solution [19]. Mathematically, this tendency can be expressed as follows:

$$\mathbf{V}_i^g = \mathbf{X}_{R_1}^g + rand[0,1](\mathbf{X}_{best}^g - |\mathbf{X}_{R_3}^g|) - rand[0,1](\mathbf{X}_{worst}^g - |\mathbf{X}_{R_3}^g|), \quad (17)$$

where R_1, R_2 , and R_3 are differential numbers selected in $[1, 2, 3, \dots, NP]$; $rand[0,1]$ is a random number between 0 and 1.

3.3. Hybrid Differential Evolution and Jaya Algorithm

This section presents a novel hybrid-combined differential evolution (DE) and Jaya algorithm (HCDJ) for the kinematic dimension synthesis of four-bar mechanisms. Like the classical DE and Jaya algorithms, the proposed-hybrid algorithm has four main stages, which include: initialization, the hybrid-combined mutation, crossover, and selection. The crossover in DE and HCDJ is the same, and the other stages in HCDJ have some modifications, which are given in great detail as follows.

3.3.1. Modified Initialization

In the dimensional synthesis of four-bar mechanisms, the highly constrained objective function depends on the constraints related to Grashof's condition and the sequences of input crank angles. Consequently, a random initialization stage might generate a small number of individuals that satisfy these above mentioned-constraints. Hence, the poor quality of the initial population might be generated. To tackle this problem, the refinement of the initialization is proposed in [2,8]. In this refinement, the first four variables related to the link lengths of the mechanism and crank angles of the input link are randomly chosen in the given range. Regarding the lengths, if they are not satisfied with Grashof's condition, they are randomly generated until Grashof's condition is true. For crank angles of the input link, if the consequential condition is false, these angles are rearranged counter-clockwise or clockwise. For example, in one problem, six input crank angles are needed and the vector of these values is generated as follows: $[\theta_2^1, \theta_2^2, \theta_2^3, \theta_2^4, \theta_2^5, \theta_2^6] = [2\pi, \pi, \pi/4, \pi/2, \pi/5, \pi/6]$. It shows that these crank angles violate the consequential condition. Thus this vector could be rearranged as follows: $[\pi/5, \pi/4, \pi/2, \pi, 2\pi, \pi/6]$ or $[2\pi, \pi, \pi/2, \pi/4, \pi/5, \pi/6]$. As a result, the initial population can provide much more feasible solutions for finding the optimal solutions. Concretely, this modified initialization is illustrated in Figure 2.

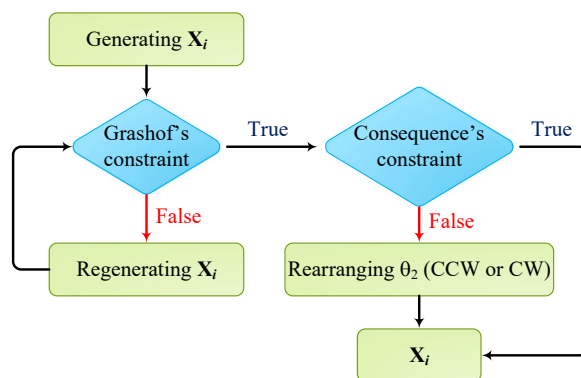


Figure 2. The modified initialization in HCDJ.

3.3.2. Hybrid-Combined Mutation

This section presents the mutation of HCDJ to increase the solution accuracy in which a hybrid-combined mutation strategy is proposed. In the proposed mutation, DE/best/1 and DE/best/2, DE/current to best/1 and Jaya operator, DE/rand/1 and DE/rand/2 belong to the first, second, and third groups of individuals, respectively. The first, second, and third groups contain the first, middle, and remaining individuals, respectively. The ratios of these groups in mutation are $x\%$, $y\%$ and $z\%$ (with $z\% = 100\% - x\% - y\%$), respectively, in which the total sum of them is 100%. In each group of mutation operators, to switch between two operators, a constant value of sigma, called σ , is used. Concretely, if a random value in the range of $[0, 1]$, called $rand()$, is bigger than σ , a new location is updated by using DE/best/1 or DE/current to best/1 or DE/rand/1; otherwise, it is updated by using DE/rand/1 or Jaya operator or DE/rand/2, respectively. Thus, a perspective view of the hybrid-combined mutation of HCDJ is provided in Figure 3, in which, a is a round number of $x\%.NP$, b is a round number of $y\%.NP$; c is the total sum of a and b .

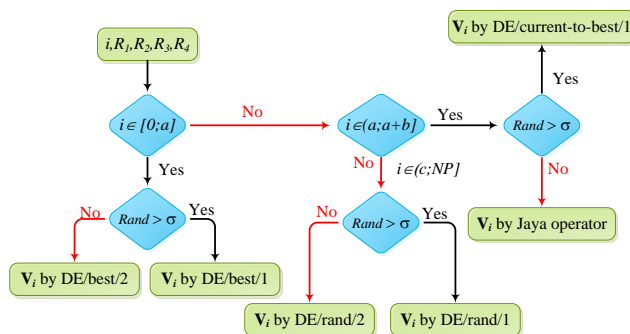


Figure 3. The hybrid-combined mutation in HCDJ.

3.3.3. Elitist Selection

In the classical DE and Jaya algorithms, finding the next population is based on the comparison of the cost function values of the old and new individuals. Even though the worse individual is better than the others in the current population, some good individuals may be neglected, and thus the convergence rate and the accuracy of the solutions are not optimized yet so far. To improve this, an elitist selection [43] is used to choose the best individuals for the next iteration. Hence, the process of the HCDJ algorithm is shown in Figure 4.

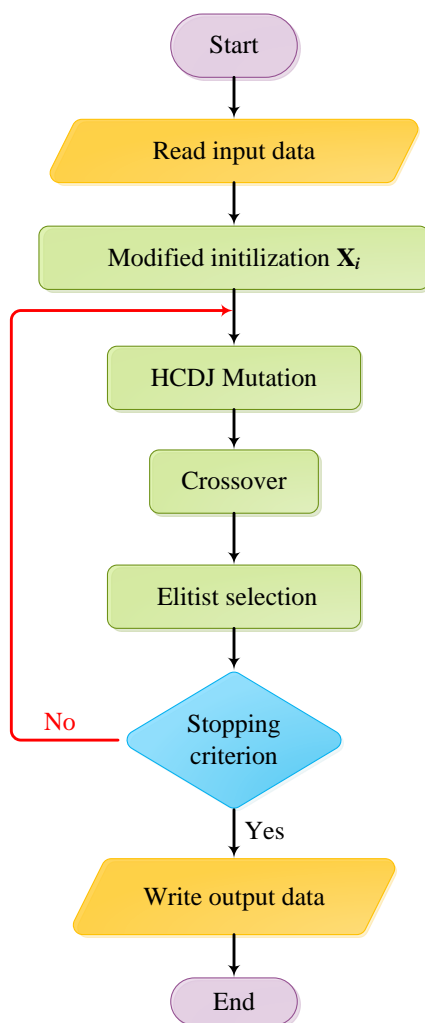


Figure 4. The HCDJ flowchart.

4. Numerical Examples

This section presents several examples in design kinematic dimensions of four-bar mechanisms with symmetrical motions, which have been investigated in [2,4,7,8,44] by using GA, DE, and PSO, respectively. Thus, results obtained by HCDJ are compared with those of GA, DE, and PSO in such algorithms. The four-bar mechanisms with symmetrical motions must satisfy the Grashof's condition, as shown in Section 2 in order to make the symmetrical motion of the mechanisms. In Cases 1 and 5, the problem is the path generation without prescribed timing. In contrast, in Cases 2, 3, and 4, the problems are known by the input of the coupler point. In this work, Jaya, DE, and HCDJ algorithms are applied for finding the optimum solutions, as shown in Figure 1, and the optimal results of these algorithms are compared with the other algorithms that have been used in the previous studies. The population sizes (NP) in Cases 1 and 5 are equal to 100, and the population sizes are equal to 50 in both Cases 2, 3, and 4. The maximum iterations in Cases 1, 4, and 5 are equal to 1000 and are equal to 100 in Cases 2 and 3. By investigations, the values of ϵ_1 in Equation (10) are chosen as 10^6 in Cases 1 and 5, 10^3 in Cases 2 and 3 and 10^4 in Case 4. Since Jaya, DE, and HCDJ are random-optimization algorithms, each different run provides different optimal solutions. To tackle this problem, DE, Jaya, and HCDJ used 50 independent times. Subsequently, the minimal errors of Jaya, DE, and HCDJ are provided and compared with other algorithms in the literature. In addition, the standard deviation and mean values of the minimal errors in 50 runs of Jaya, DE, and HCDJ are also reported.

For validations of the obtained solutions, the synthesized mechanisms are illustrated in GeoGebra classic 5.

4.1. Case 1

In this case, the input data of the coupler point are presented in [2,4,7]. The design variables' vector, input path and boundaries for design parameters are respectively provided in Equation (18) to Equation (20).

The design parameters are presented by a vector as follows:

$$\mathbf{X} = [r_1 \ r_2 \ r_3 \ r_4 \ a \ b \ x_{02} \ y_{02} \ \theta_1 \ \theta_2^1 \ \theta_2^2 \ \dots \ \theta_2^6]. \tag{18}$$

The input data of the coupler pointer is given as follows:

$$\mathbf{C}^i = \left[\begin{matrix} (20, 20); (20, 25); (20, 30); \\ (20, 35); (20, 40); (20, 45) \end{matrix} \right]. \tag{19}$$

The boundary conditions of the design parameters can be expressed as follows:

$$r_1, r_2, r_3, r_4 \in [0, 60]; a, b, x_{02}, y_{02} \in [-60, 60]; \theta_1, \theta_2^1, \theta_2^2, \theta_2^3, \theta_2^4, \theta_2^5, \theta_2^6 \in [0, 2\pi]. \tag{20}$$

4.1.1. Effects of the Parameters of **F** and σ in HCDJ on the Optimal Solutions

This section investigates the effects of the parameters of **F** and σ in HCDJ on the optimal solutions. Firstly, the mutant factor (**F**) used in HCDJ is investigated by considering the following five cases: 0.4, 0.5, 0.6, 0.7, and the range [0.7, 1]. In this examination, the values of σ , **x%** and **y%** are equal to 0.3, 30% and 30%, respectively. Obtained results are shown in Table 1. It can be seen that when **F** is equal to 0.7, the HCDJ algorithm yields the best optimal solutions compared to other cases of **F**. Next, the seven different cases of σ are investigated in which the values of **F**, **x%** and **y%** are equal to [0.7, 1], 30% and 30%, respectively. Obtained results are shown in Table 2. It can be seen that when σ is equal to 0.2, the HCDJ algorithm yields the best optimal solutions compared to other cases of σ . From the obtained results in Tables 1 and 2, the suitable values of **F** and σ are set to 0.7 and 0.2, respectively, and are recommended for the HCDJ.

Table 1. Effects of the parameters of mutant factor **F** on the optimal solutions.

Mutant Factor (F)	0.4	0.5	0.6	0.7	0.7–1
Best	0.07695433	0.10400792	0.000154146	1.20×10^{-19}	6.76×10^{-9}
r ₁	45.87582	54.84049	51.12059	12.2555920	18.21862
r ₂	12.11600	14.10419	12.28880	2.0128211	8.97325
r ₃	37.42606	46.71333	32.62586	39.4665020	13.48372
r ₄	39.09108	36.70026	30.78353	29.2238260	21.96705
a	31.98852	41.49218	8.86192	−20.7556260	21.58954
b	4.75479	16.66319	52.47351	−37.5648490	7.80185
x ₀₂	29.597174	38.581491	59.116081	−20.904232	2.3876745
y ₀₂	2.0836604	−4.3644949	1.3369118	29.795821	46.424455
θ ₁	0.8441283	0.96808922	0.50624868	1.7806168	3.9588942
θ ₂ ¹	4.5375149	4.2886192	6.0495074	6.2564052	1.5083156
θ ₂ ²	5.3345119	5.2128531	6.2491013	0.69846887	2.3068354
θ ₂ ³	5.735772	5.5620331	0.15263921	1.3722591	2.8305475
θ ₂ ⁴	6.0675517	5.8415786	0.34030181	2.0808477	3.3349026
θ ₂ ⁵	0.12997377	6.103831	0.54174091	2.8112326	3.8315377
θ ₂ ⁶	0.7069234	0.10575595	0.776431	3.6756295	0.23569772

Table 2. Effects of the parameters of σ on the optimal solutions.

Sigma (σ)	0.2	0.3	0.4	0.5	0.6	0.7	0.7–1.0
Best	6.44×10^{-11}	5.16×10^{-7}	6.10×10^{-5}	1.61×10^{-3}	8.04×10^{-4}	1.36×10^{-3}	9.61×10^{-2}
r_1	6.454124	8.902184	51.03007	55.98162	56.07093	18.34878	41.153418
r_2	4.247626	2.860693	6.886417	9.433685	9.696423	5.683074	15.047996
r_3	35.66853	11.74068	18.9	35.82626	14.01636	39.66747	31.093173
r_4	37.86445	6.550978	39.01872	55.88284	52.08595	44.77886	29.025764
a	9.921297	44.48886	−26.9677	36.44039	−6.46232	55.32052	21.266427
b	−15.3695	0.222302	−38.3009	37.45101	18.08502	−28.5643	14.791038
x_{02}	5.956732	−21.6466	59.95463	59.72303	29.49918	−36.5538	27.663117
y_{02}	35.39474	42.21716	29.51548	1.256564	18.1272	46.72558	10.912645
θ_1	5.107614	5.5287	4.744479	0.469354	5.815868	4.869602	0.72840409
θ_1^1	1.1885	0.577693	1.15204	5.911371	0.530409	2.44282	5.262023
θ_2^2	2.185696	1.699321	1.35547	6.232504	0.761861	2.874473	5.6355356
θ_3^3	3.164171	2.399636	1.555896	0.216276	0.994329	3.272605	5.8961624
θ_4^4	4.331262	3.02243	1.758188	0.47504	1.250005	3.660652	6.1145361
θ_5^5	5.306879	3.603063	1.966733	0.758422	1.564631	4.057665	0.058349954
θ_6^6	0.015795	4.251288	2.185992	1.135433	2.144167	4.489581	0.32377987

4.1.2. Effects of the Parameters of $x\%$ and $y\%$ in HCDJ on the Optimal Solutions

Next, the effects of the parameters of $x\%$ and $y\%$ in HCDJ on the optimal solutions are studied. In this examination, the values of F and σ are set to 0.7, and 0.2, respectively. The values of $x\%$ and $y\%$ used in HCDJ are investigated by considering 16 different cases. The obtained results are shown in Tables 3 and 4. It can be seen that when $x\%$ and $y\%$ are equal to 20% and 30%, the HCDJ algorithm yields best optimal solutions compared to other cases of $x\%$ and $y\%$. From the obtained results in Tables 3 and 4, the suitable values of $x\%$ and $y\%$ are chosen as 20% and 30%, respectively, in HCDJ operators and are recommended for the HCDJ.

Table 3. Effects of the parameters of $x\%$ and $y\%$ on the optimal solutions.

$x\%$	10%	20%	10%	30%	20%	10%	40%	30%
$y\%$	10%	10%	20%	10%	20%	30%	10%	20%
Best	3.91×10^{-28}	1.26×10^{-28}	1.09×10^{-5}	5.15×10^{-16}	1.60×10^{-6}	6.87×10^{-16}	5.18×10^{-24}	4.67×10^{-28}
r_1	24.836075	53.819222	59.990134	29.986241	39.678744	26.798566	31.90364	27.117518
r_2	7.0254235	10.151379	9.0469478	8.8527129	11.043626	13.058193	14.177637	8.0864721
r_3	23.830941	33.044744	42.071536	24.254956	33.750908	30.955844	38.498459	30.595952
r_4	24.291336	30.925862	30.560764	37.460255	51.656635	20.100519	20.778983	15.464134
a	39.873429	21.955928	27.901291	35.107097	48.928319	43.908515	57.423517	55.288227
b	16.731031	33.985179	51.47438	4.1357288	3.6625348	21.205064	30.808235	39.50158
x_{02}	−14.892824	42.579861	59.999655	−5.6369319	−16.597876	−14.293538	−28.922891	−38.544782
y_{02}	55.070581	−0.24873838	−9.3328501	52.080728	59.858184	54.950423	59.999888	58.388173
θ_1	4.2906588	0.64922531	0.77133588	3.9968142	4.0236654	4.4203068	4.5557802	4.6659105
θ_1^1	1.65102	4.8303017	5.040377	2.1731417	2.4345948	5.929895	5.7252922	6.062974
θ_2^2	2.3678046	5.7665232	5.7358987	2.7056893	2.825861	2.8287252	2.930339	2.6785003
θ_3^3	2.897899	6.1322647	6.108357	3.1902928	3.1982302	3.18E+00	3.228627	3.0439695
θ_4^4	3.4029953	0.18029565	0.16680844	3.6891147	3.5761384	3.5640285	3.5457339	3.4098621
θ_5^5	3.9362564	0.55118656	0.55128744	4.2402232	3.9764453	3.995454	3.8992005	3.7980462
θ_6^6	4.6676931	1.3028902	1.2772089	4.9851799	4.4261382	4.7556041	4.3901059	4.3010471

4.1.3. Comparison Performances of HCDJ with Other Available Methods in the Literature

Table 5 provides the optimal results obtained by HCDJ, and other approaches. It can be seen that HCDJ gives the best optimal solutions in all algorithms, 8.84×10^{-29} for HCDJ. Additionally, Figure 5 shows the best path traced by the coupler in Case 1 by using HCDJ and there is a very good traced path of the point C. Additionally, the convergence rates of the HCDJ, DE, and Jaya are shown in Figure 6. The HCDJ reaches the optimal solutions faster than the DE and the Jaya.

Table 4. Effects of the parameters of $x\%$ and $y\%$ on the optimal solutions.

$x\%$	25%	20%	10%	50%	40%	30%	20%	10%
$y\%$	25%	30%	40%	10%	20%	30%	40%	50%
Best	7.41×10^{-24}	8.84×10^{-29}	3.10×10^{-5}	4.08×10^{-20}	2.33×10^{-6}	1.77×10^{-28}	8.08×10^{-8}	3.31×10^{-24}
r_1	36.663429	14.201369	59.87156	22.952088	49.267118	27.0259	17.946774	17.180275
r_2	9.2303679	8.0143127	11.268517	6.5882229	10.617582	10.054969	4.312293	8.2263098
r_3	22.319996	16.309922	38.873302	26.439382	30.268652	30.260301	20.637774	20.881485
r_4	42.520857	12.253871	32.295882	21.809868	59.842261	18.214837	12.222502	12.088439
a	29.339283	27.797721	24.443916	57.394819	39.252095	47.314415	55.484891	37.799731
b	5.4051762	11.305412	44.809713	18.957001	4.261466	28.561374	28.761501	17.900803
x_{02}	0.40154499	-2.309531	51.854439	-33.086389	-6.7350053	-23.684977	-37.800797	-13.404186
y_{02}	51.718045	45.695083	-8.2217234	57.009133	57.214806	56.814205	52.911313	48.657837
θ_1	3.7793071	4.5172417	0.69260257	4.6608822	3.7675405	4.5147889	4.871693	4.6989581
θ_2^1	2.0406231	6.2532447	5.6671392	1.8399122	2.4021966	6.0540795	1.253857	5.9481891
θ_2^2	2.6369825	2.4038981	6.000184	2.3617136	2.86E+00	2.7399178	2.0984192	2.5477842
θ_2^3	3.1435106	2.9362397	6.2831236	2.8021379	3.280675	3.1103903	2.6447287	2.9986464
θ_2^4	3.6699251	3.4905751	0.2831384	3.2183377	3.7213306	3.4942181	3.128454	3.4596663
θ_2^5	4.2730939	4.0650776	0.61402296	3.6367153	4.2144194	3.9154415	3.6002535	3.9454767
θ_2^6	5.5053325	5.2095019	1.2131414	4.0977968	4.859791	4.5440765	4.1378877	4.6594827

Table 5. Optimal results of Case 1.

Design	Cabrera [4]	Cabrera [6]	Ortiz [7]	WY Lin [8]	This Study		
Variables	GA	MUMSA	IOA s-at	30%–30%–40%	DE	Jaya	HCDJ
r_1	39.46629	31.78826	54.71582	45.95403	53.9345	27.33122	14.20136900
r_2	8.56291	8.20465	18.73099	10.14134	11.72806	8.675739	8.01431270
r_3	19.09486	24.93213	31.22310	36.11986	24.30051	58.2691	16.30992200
r_4	47.83886	31.38593	42.22374	49.79056	59.70239	59.94451	12.25387100
a	13.38556	34.19372	-27.29874	51.49352	29.16371	-4.0256	27.79772100
b	12.21961	14.41567	31.65051	16.49573	5.564619	47.38173	11.30541200
x_{02}	29.72250	-6.36652	43.07086	59.92519	3.088134	58.85711	-2.30953100
y_{02}	23.45450	56.83676	27.41706	-0.26726	57.3939	32.75311	45.69508300
θ_1	6.20163	4.01596	5.97746	0.86238	3.603526	6.280857	4.51724170
θ_2^1	6.11937	1.36655	6.42411	4.49683	1.74656	5.829859	6.25324470
θ_2^2	0.19304	2.33077	6.53496	3.94685	2.399606	6.038721	2.40389810
θ_2^3	0.44083	2.87104	0.36230	3.51209	2.83412	6.213979	2.93623970
θ_2^4	0.68467	3.39459	0.46906	3.11078	3.260462	0.093753	3.49057510
θ_2^5	0.95835	3.97096	0.57765	2.71011	3.727497	0.255427	4.06507760
θ_2^6	1.35533	4.96349	0.69047	2.26550	4.28393	0.425728	5.20950190
Best	0.036298225	0.0002057	0.00023712	4.07×10^{-12}	0.00088	0.008574	8.84×10^{-29}
Max	-	-	-	-	13.5226	437.5001	62.5000
Mean	-	-	-	-	0.7120	30.6096	2.8596
STD	-	-	-	-	2.6144	68.7252	9.5524

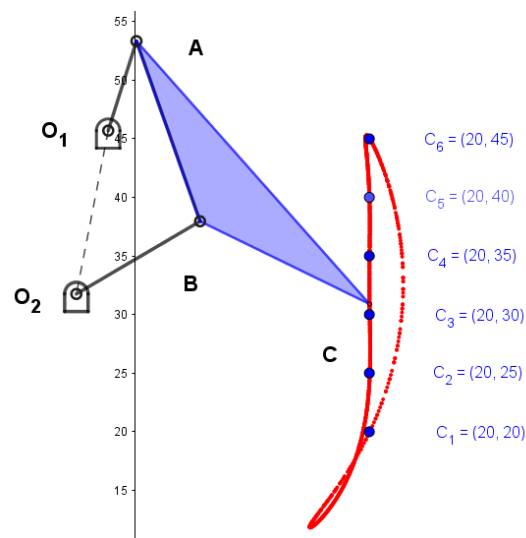


Figure 5. The best path traced by the coupler in Case 1 by using HCDJ.

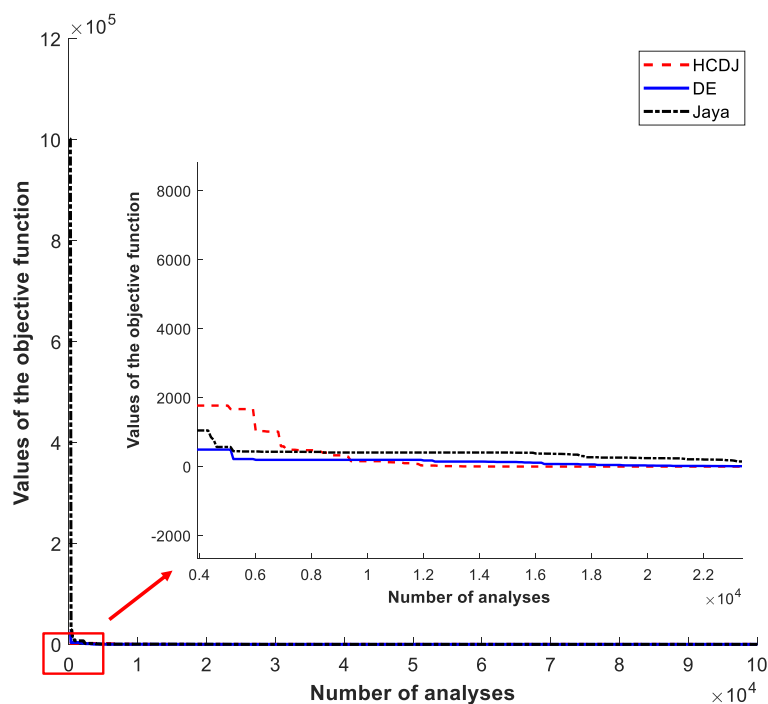


Figure 6. The convergence of objective values in Case 1 by using the DE, Jaya and HCDJ with a logarithm scale for the Y-axis.

4.2. Case 2

In this case, path generation with a prescribed timing of four-bar mechanism is performed, which is also investigated in [2], the inputs for the optimization problem are six coupler points and these points belong to a semi-circular arc. Thus, the design variables’ vector is defined as follows:

$$\mathbf{X} = [r_1 \ r_2 \ r_3 \ r_4 \ a \ b]. \tag{21}$$

The six input coupler points are chosen as follows:

$$\begin{cases} \mathbf{C}^i = [(3, 3); (2.759, 3.363); (2.372, 3.663); \\ (1.890, 3.862); (1.355, 3.943)]', \\ \theta_2^1, \theta_2^2, \theta_2^3, \theta_2^4, \theta_2^5 = [\pi/6; \pi/4; \pi/3; 5\pi/12; \pi/2]. \end{cases} \tag{22}$$

The boundary conditions of the design parameters can be expressed as follows:

$$r_1, r_2, r_3, r_4 \in [0, 50]; a, b \in [-50, 50]. \tag{23}$$

Table 6 shows the optimal solutions obtained by HCDJ and other algorithms. It can see that HCDJ gives the best optimal solutions in all algorithms, $1.92392270 \times 10^{-6}$ for HCDJ. Additionally, Figure 7 shows the best path traced by the coupler in Case 2 with using HCDJ and there is a very good traced path of the coupler (point C in link 3). Furthermore, the convergence speed of the HCDJ, DE, and Jaya are also illustrated in Figure 8. The HCDJ algorithm reaches the optimal solutions much faster than the DE and the Jaya.

Table 6. Optimal results of Case 2.

Design	KK [44]	Cabrera [4]	Ortiz [7]	This Study		
Variables	GA	GA	IOA s-at	DE	Jaya	HCDJ
r_1	3.509643	3.0630424	2.803607	14.49987	12.83291	4.29308120
r_2	1.857606	1.9959624	1.99226	1.972381	2.031706	1.99783240
r_3	4.725835	3.305823	3.030461	34.78773	39.75247	4.73740990
r_4	3.518721	2.524706	2.474117	28.74995	36.2862	2.94173010
a	1.959538	1.645158	1.64413	2.362063	2.3363	1.70740510
b	1.558898	1.708959	1.714536	0.373688	-0.05985	1.64416960
Best	9.5264×10^{-4}	4.08×10^{-6}	4.27×10^{-6}	9.57×10^{-4}	9.68×10^{-4}	$1.92392270 \times 10^{-6}$
Max	-	-	-	0.5190	3.9253	0.4668
Mean	-	-	-	0.1005	0.3506	0.0961
STD	-	-	-	0.1486	0.6218	0.0910

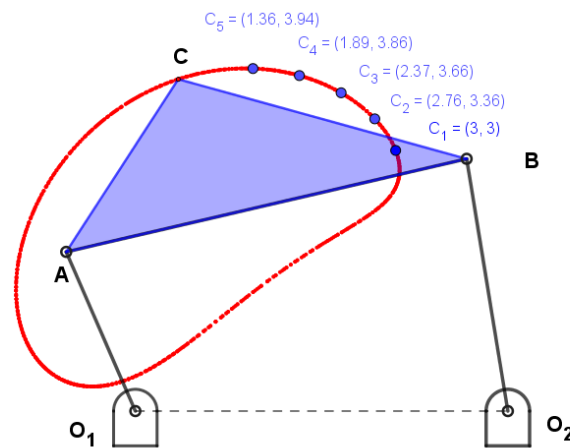


Figure 7. The best path traced by the coupler in Case 2 by using HCDJ.

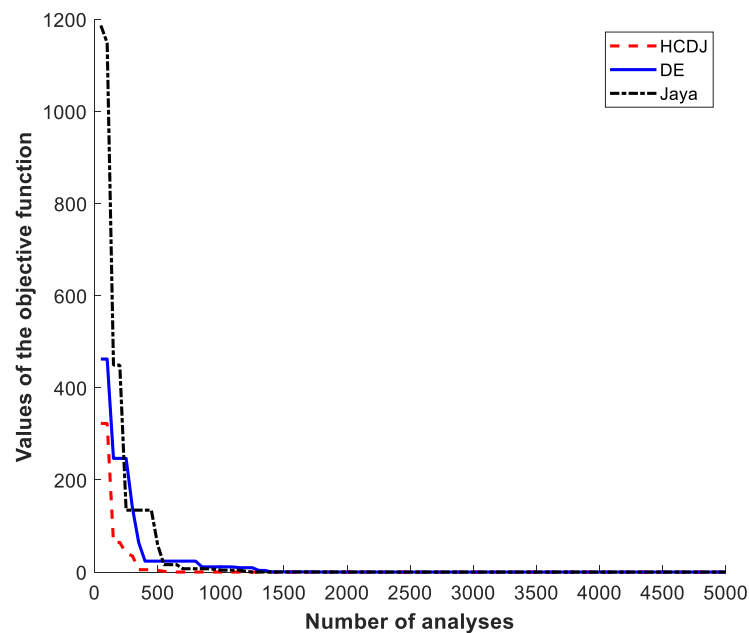


Figure 8. The convergence of objective values in Case 2 by using the DE, Jaya and HCDJ.

4.3. Case 3

For the third case, the coupler point traces a close loop path generation in which 18 coupler points are included and prescribed timing is required. This problem was first presented in [44]. Thus, the vector of design variable is defined as follows:

$$\mathbf{X} = [r_1 \ r_2 \ r_3 \ r_4 \ a \ b \ x_{02} \ y_{02} \ \theta_1 \ \theta_2^1]. \tag{24}$$

The 18 desired coupler points are chosen as follows:

$$\begin{cases} \mathbf{C}_d^i = \left[\begin{array}{l} (0.5, 1, 1); (0.4, 1.1); (0.3, 1.1); (0.2, 1.0); (0.1, 0.9); (0.05, 0.75); \\ (0.02, 0, 6); (0.0, 0.5); (0.0, 0.4); (0.03, 0.3); (0.1, 0.25); (0.15, 0.2); \\ (0.2, 0.3); (0.3, 0.4); (0.4, 0.5); (0.5, 0.7); (0.6, 0.9); (0.6, 1.0) \end{array} \right], \\ \theta_2^2, \theta_2^3, \theta_2^4, \theta_2^5, \theta_2^6 = [\pi/6; \pi/4; \pi/3; 5\pi/12; \pi/2]. \end{cases} \tag{25}$$

The boundary conditions of the design parameters can be expressed as follows:

$$r_1, r_2, r_3, r_4 \in [0, 50]; a, b, x_{02}, y_{02} \in [-50, 50]; \theta_1 \in [0, 2\pi]. \tag{26}$$

Table 7 shows the optimal solutions obtained by HCDJ and other algorithms. It can be seen that HCDJ gives the best optimal solutions in all algorithms, 0.016077817 for HCDJ. Additionally, Figure 9 shows the best path traced by the coupler in Case 3 using HCDJ, and there is a very good traced path of the coupler (point C in link 3). Furthermore, the convergence speed of the HCDJ, DE, and Jaya are also illustrated in Figure 10. The HCDJ reaches the optimal solutions much faster than the DE and the Jaya.

Table 7. Optimal results of Case 3.

Design Variables	This Study					
	KK [44] GA	Cabrera [4] GA	Ortiz [7] IOA s-at	DE	Jaya	HCDJ
r_1	1.8796600	3.0578780	4.0404350	43.9060510	40.7943060	49.80088600
r_2	0.2748530	0.2378030	0.2452160	0.3914995	0.2588295	0.28817159
r_3	1.1802530	4.8289540	6.3829400	34.1729750	42.1511430	48.47613800
r_4	2.1382090	2.0564560	2.6205320	38.8796410	5.3834310	1.63364200
a	-0.8335920	0.7670380	1.1391060	13.5079350	11.6043990	-23.41281000
b	-0.3787700	1.8508280	1.8661090	11.1797190	6.9199644	7.66448860
x_{02}	1.1320620	1.7768080	1.8918050	-6.3236304	-11.4890630	-14.83615300
y_{02}	0.6634330	-0.6419910	-0.7613390	16.9547380	7.3215994	20.14022300
θ_1	4.3542240	1.0021680	1.1877510	3.3892883	5.1119994	2.52860910
θ_2^1	2.5586250	0.2261860	0.0000000	4.1084598	2.8399163	5.06442240
Best	0.0430000	0.0337480	0.0349885	0.1059867	1.3193569	0.016077817
Max	-	-	-	9.4391	83.1584	2.5112
Mean	-	-	-	1.4682	17.293	0.3031
STD	-	-	-	1.6327	16.0914	0.5407

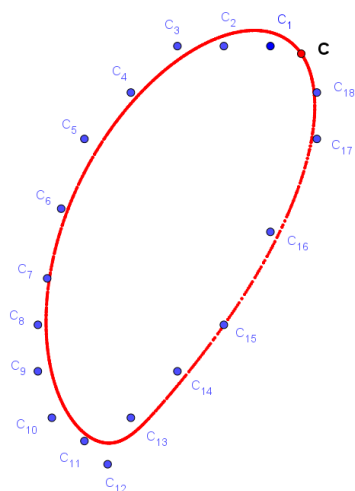


Figure 9. The best path traced by the coupler in Case 3 by using HCDJ.

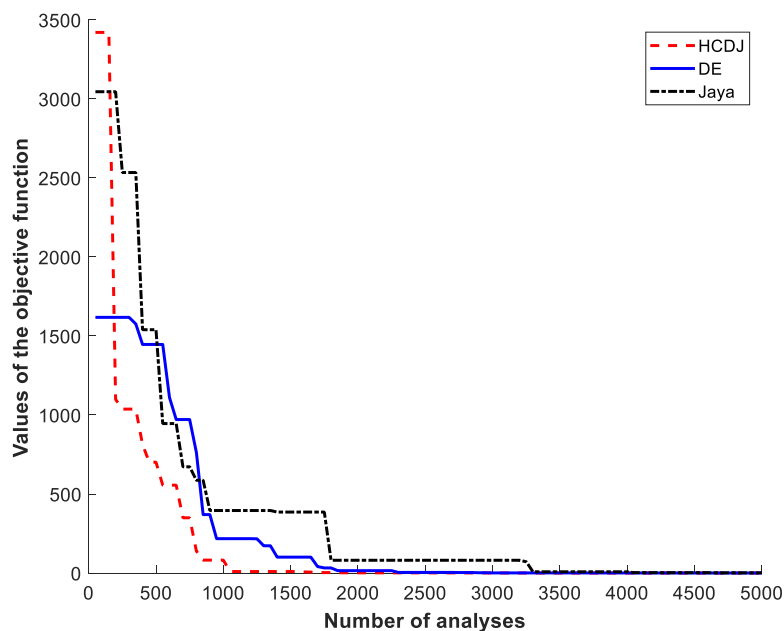


Figure 10. The convergence of objective values in Case 3 by using the DE, Jaya and HCDJ.

4.4. Case 4

In the Case 4, a path generation problem with prescribed timing is performed. Six coupler points in a vertical straight line are used as inputs. Then, the vector of design variables is given as follows

$$\mathbf{X} = [r_1 \ r_2 \ r_3 \ r_4 \ a \ b \ x_{02} \ y_{02} \ \theta_1]. \tag{27}$$

The 18 desired coupler points are selected as follows:

$$\begin{cases} \mathbf{C}^i = [(0, 0); (1.9098, 5.8779); (6.9098, 9.5106); \\ (13.09, 9.5106); (18.09, 5.8779); (20, 0)]', \\ \theta_2^1, \theta_2^2, \theta_2^3, \theta_2^4, \theta_2^5, \theta_2^6 = [\pi/6; \pi/3; \pi/2; 2\pi/3; 5\pi/6, \pi]. \end{cases} \tag{28}$$

Lower and upper boundary for design variables is taken in an interval as the following equation which is given as follows:

$$r_1, r_2, r_3, r_4 \in [0, 50]; a, b, x_{02}, y_{02} \in [-50, 50]; \theta_1 \in [0, 2\pi]. \tag{29}$$

Table 8 shows the optimal solutions obtained by HCDJ and other algorithms. It can be seen that HCDJ gives the best optimal solutions in all algorithms, 1.21621220 for HCDJ. Additionally, Figure 11 shows the best path traced by the coupler in Case 4 using HCDJ and there is a very good traced path of the coupler (point C in link 3). Furthermore, the convergence speed of HCDJ, DE, and Jaya is also illustrated in Figure 12. The HCDJ algorithm reaches the optimal solutions much faster than the DE and Jaya algorithms.

Table 8. Optimal results of Case 4.

Design Variables	Cabrera [2]		Ortiz [7]		This Study	
	PSO	DE	IOA	DE	Jaya	HCDJ
r_1	49.994859	50	49.968967	46.619505	47.5628	49.99998000
r_2	5	5	4.785659	1.5729995	10.1759	1.34843150
r_3	5.915643	5.905345	6.491026	1.5900729	26.5621	1.34845070
r_4	49.994867	50	48.393942	46.603155	33.6611	50.00000000
a	18.925715	18.819312	16.444782	11.910771	49.8608	11.38443400
b	0	0	11.988091	4.0847137	34.6944	4.44243570
x_{02}	14.472475	14.373772	12.046587	10.349552	49.1293	10.19475200
y_{02}	-12.494409	-12.444295	-14.774897	-4.2274931	-46.173	-3.69413400
θ_1	0.467287	0.463633	0.038678	6.28E+00	1.03722	6.21587990
Best	5.547239141	5.520687978	2.490688998	1.2900783	5.60164	1.21621220
Max	-	-	-	14.0282	144.9284	442.1287
Mean	-	-	-	7.4516	24.1544	26.0644
STD	-	-	-	4.3290	24.7319	68.4167

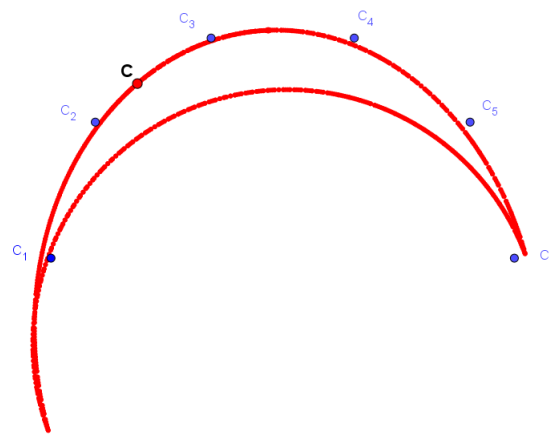


Figure 11. The best path traced by the coupler in Case 4 by using HCDJ.

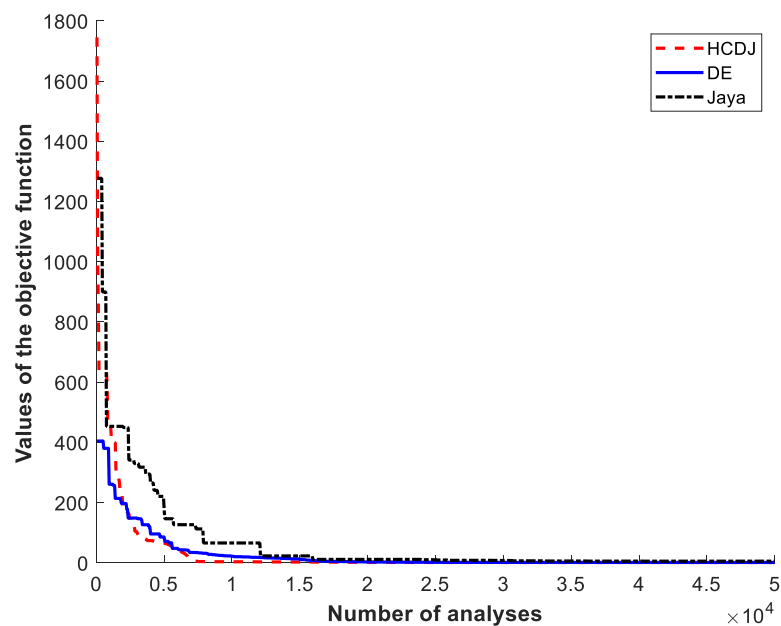


Figure 12. The convergence of objective values in Case 4 by using the DE, Jaya and HCDJ.

4.5. Case 5

In the fifth case, an elliptical path generation problem without prescribed timing is investigated. The path consists of 10 target points. The elliptical path with a major axis of 20 units and a minor one of 16 units is considered. The center’s coordinate is at (10, 10) and the major axis is kept horizontal.

The vector of design variables is given as follows:

$$X = [r_1 \ r_2 \ r_3 \ r_4 \ a \ b \ x_{02} \ y_{02} \ \theta_1 \ \theta_2^1 \ \theta_2^2 \ \dots \ \theta_2^{10}]. \tag{30}$$

The desired coupler points are chosen as follows:

$$C^i = \left[\begin{array}{l} (20, 10); (17.66, 15.142); (11.736, 17.878); \\ (5, 16.928); (0.60307, 12.736); (0.60307, 7.2638), \\ (5, 3.0718), (11.736, 2.1215), (17.66, 4.8577), (20, 10) \end{array} \right]. \tag{31}$$

The boundary conditions of the design parameters can be expressed as follows:

$$r_1, r_2, r_3, r_4 \in [5, 80]; a, b, x_{02}, y_{02} \in [-80, 80]; \theta_1, \theta_2^1, \theta_2^2, \dots, \theta_2^{10} \in [0, 2\pi]. \tag{32}$$

Table 9 shows the optimal solutions obtained by HCDJ and other algorithms. It can be seen that HCDJ and DE in [8] give the best optimal solutions in all algorithms, 4.201438×10^{-4} for HCDJ and 4.01992×10^{-4} for DE [8]. Additionally, Figure 13 shows the best path traced of the coupler in Case 5 using HCDJ, and there is a very good traced path of the coupler (point C in link 3). Furthermore, the convergence speed of the HCDJ, DE, and Jaya are also illustrated in Figure 14. The HCDJ algorithm reaches the optimal solutions much faster than the DE and the Jaya algorithms.

Table 9. Optimal results of Case 5.

Design Variables	Cabrera [6] MUMSA	Ortiz [7] IOA s-at	WY Lin [8] 10%–20%–70%	DE	Jaya	This Study HCDJ
r_1	79.5160680	65.4287710	80.0000000	73.4494770	41.3002340	79.999856000
r_2	9.7239730	8.0163870	8.0456620	8.3019102	9.2637486	8.123200800
r_3	45.8425240	47.2216550	50.8190200	52.4311980	72.7938790	50.870344000
r_4	51.4328480	44.1365600	42.2080100	35.0021110	42.8027070	42.360533000
a	8.2139220	−11.5708580	−10.6369700	−10.5394310	6.9607656	−10.853763000
b	−2.9539575	−1.9049140	−2.2910900	4.5542557	1.2414044	0.091433127
x_{02}	2.0211090	10.6354140	8.4948130	5.9241807	15.3054700	7.537903100
y_{02}	13.2165878	−1.6754770	−0.7579678	−0.6075295	14.6833360	−0.522541410
θ_1	5.5969445	3.8673300	3.8892100	4.3045025	3.1834274	4.012119500
θ_2^1	0.6376873	2.4199310	2.4494370	2.0623267	3.1556054	2.342859900
θ_2^2	1.3255329	3.1092670	3.1539690	2.7790775	3.8621079	3.053512700
θ_2^3	2.0080339	3.8129500	3.8371140	3.4847400	4.6327586	3.741746400
θ_2^4	2.6955659	4.5064400	4.5201710	4.1731807	5.3273744	4.425545700
θ_2^5	3.3845794	5.1811390	5.2047990	4.8544466	6.0138412	5.108671600
θ_2^6	4.0829376	5.8834200	5.8985360	5.5419381	0.4538612	5.798382900
θ_2^7	4.7984548	0.2962630	0.3162040	6.2310760	1.0808289	0.212868930
θ_2^8	5.5117057	0.9911530	1.0235550	0.6409663	1.7972805	0.916585540
θ_2^9	6.2127919	1.7077870	1.7389920	1.3516693	2.4996946	1.630413500
θ_2^{10}	0.6371866	2.4188650	2.4494370	2.0622545	3.1550369	2.342815200
Best	0.004766469	0.01909700	4.01992×10^{-4}	0.002336965	0.48257144	4.201438×10^{-4}
Max	-	-	-	12.5292	828.1948	99.2594
Mean	-	-	-	0.7305	138.4789	2.3325
STD	-	-	-	2.0771	190.6784	14.0713

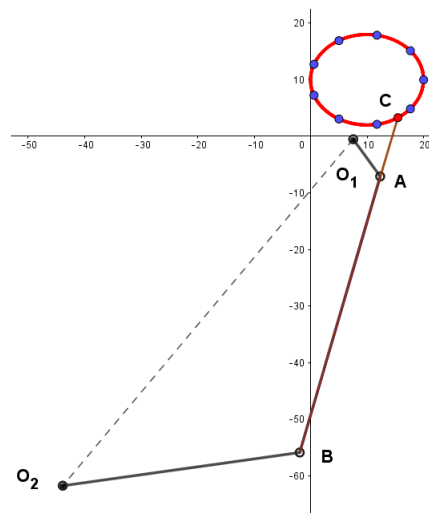


Figure 13. The best path traced by the coupler in Case 5 by using HCDJ.

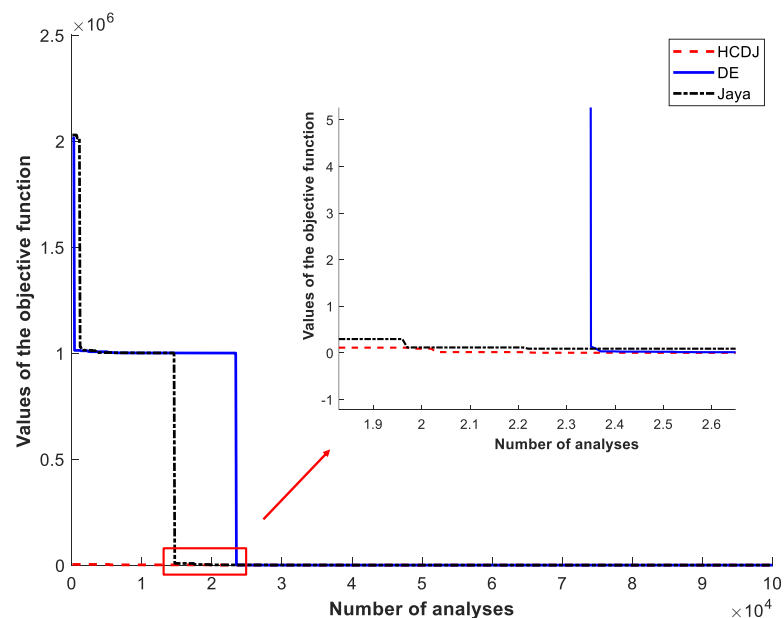


Figure 14. The convergence of objective values in Case 5 by using the DE, Jaya and HCDJ.

5. Discussion

The obtained results in five cases show that the optimal results of HCDJ are better than those of other algorithms. Except in Case 5, the best error in [8] is slightly lighter than those of HCDJ. In addition, the statistical results in terms of the maximum, mean, and standard deviation of best errors in 50 runs are also reported. HCDJ and DE are more stable than Jaya in all five cases. However, the statistical values of classical DE are smaller than those of HCDJ in Cases 1, 4, and 5. In contrast, the performances of HCDJ, in terms of statistical results, are better than those of DE in Cases 2 and 3. It should be noted that the synthesis of the mechanism is an engineering problem, and the key objective is to find the best and most optimal results. In addition, as stated by “No free lunch theorems for optimization” in [45], no optimization algorithm is the best for all problems. Thus, HCDJ outperforms other methods in the dimensional synthesis of the four-bar mechanism in terms of optimal solutions.

6. Conclusions

This study proposed a newly hybrid-combined algorithm, called **HCDJ**, as a combination of the classical DE and Jaya algorithms for the optimally dimensional design of four-bar mechanisms with symmetrical motions. The combined algorithm has a good global search to improve the optimal solution quality by using modified initialization, a hybrid-combined mutation between the classical DE and Jaya algorithm, and the elitist selection. The modified initialization generates initial individuals that are satisfied with Grashof's condition and consequential constraints. In the hybrid-combined mutation, three different groups of mutations are combined. DE/best/1 and DE/best/2, DE/current to best/1 and Jaya operator, and DE/rand/1, and DE/rand/2 belong to the first, second, and third groups, respectively. In the second group, DE/current to best/1 is hybrid with the Jaya operator. Additionally, in the selection stage, the best candidates are produced for the next generation by using the elitist selection technique. Five numerical examples, including two path generations with prescribed timing and three without prescribed timing, are performed to find the optimal designs of the four-bar mechanisms. The obtained solutions of HCDJ are compared with those of the original DE, Jaya, and other algorithms existing in the literature. The optimal results using the HCDJ algorithm have indicated that it can achieve better performances in terms of the solution accuracy than the original DE and Jaya, even in many other algorithms. Accordingly, the proposed HCDJ algorithm is expected to apply not only to symmetrical motion mechanisms, but also asymmetrical motions of mechanisms and various engineering problems.

Author Contributions: S.N.-V.: Conceptualization, methodology, investigation, resources, software, writing—original draft, funding acquisition. Q.X.L.: investigation, writing—review and editing. N.X.-M.: conceptualization, methodology, software, writing—original draft, writing—review and editing, supervision. T.T.N.N.: conceptualization, methodology, software, writing—original draft, writing—review and editing, project administration, supervision. All authors have read and agreed to the published version of the manuscript.

Funding: This research is funded by Sejong University.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Freudenstein, F. Advanced mechanism design: Analysis and synthesis. *Mech. Mach. Theory* **1985**, *20*, 81. [[CrossRef](#)]
2. Acharyya, S.; Mandal, M. Performance of EAs for four-bar linkage synthesis. *Mech. Mach. Theory* **2009**, *44*, 1784–1794. [[CrossRef](#)]
3. Zhang, C.; Norton, P.R.; Hammonds, T. Optimization of parameters for specified path generation using an atlas of coupler curves of geared five-bar linkages. *Mech. Mach. Theory* **1984**, *19*, 459–466. [[CrossRef](#)]
4. Cabrera, J.; Simon, A.; Prado, M. Optimal synthesis of mechanisms with genetic algorithms. *Mech. Mach. Theory* **2002**, *37*, 1165–1177. [[CrossRef](#)]
5. Nguyen-Van, S.; Nguyen, T.T.N.; Nguyen-Dinh, N.; Lieu, Q.X. Truss Optimization Under Frequency Constraints by Using a Combined Differential Evolution and Jaya Algorithm. In *Advances in Engineering Research and Application*; Springer International Publishing: Berlin/Heidelberg, Germany, 2020; pp. 861–873. [[CrossRef](#)]
6. Cabrera, J.; Ortiz, A.; Nadal, F.; Castillo, J. An evolutionary algorithm for path synthesis of mechanisms. *Mech. Mach. Theory* **2011**, *46*, 127–141. [[CrossRef](#)]
7. Ortiz, A.; Cabrera, J.; Nadal, F.; Bonilla, A. Dimensional synthesis of mechanisms using Differential Evolution with auto-adaptive control parameters. *Mech. Mach. Theory* **2013**, *64*, 210–229. [[CrossRef](#)]
8. Lin, W.; Hsiao, K. A new differential evolution algorithm with a combined mutation strategy for optimum synthesis of path-generating four-bar mechanisms. *Proc. Inst. Mech. Eng. Part C J. Mech. Eng. Sci.* **2016**, *231*, 2690–2705. [[CrossRef](#)]
9. Jones, G.; Willett, P.; Glen, R.C.; Leach, A.R.; Taylor, R. Development and validation of a genetic algorithm for flexible docking 1 Edited by F. E. Cohen. *J. Mol. Biol.* **1997**, *267*, 727–748. [[CrossRef](#)]
10. Nguyen, T.; Kurtenbach, S.; Hüsing, M.; Corves, B. A general framework for motion design of the follower in cam mechanisms by using non-uniform rational B-spline. *Mech. Mach. Theory* **2019**, *137*, 374–385. [[CrossRef](#)]

11. Eqra, N.; Abiri, A.H.; Vatankhah, R. Optimal synthesis of a four-bar linkage for path generation using adaptive PSO. *J. Braz. Soc. Mech. Sci. Eng.* **2018**, *40*, 1–11. [[CrossRef](#)]
12. Trelea, I.C. The particle swarm optimization algorithm: convergence analysis and parameter selection. *Inf. Process. Lett.* **2003**, *85*, 317–325. [[CrossRef](#)]
13. de Bustos, I.F.; Urkullu, G.; Marina, V.G.; Ansola, R. Optimization of planar mechanisms by using a minimum distance function. *Mech. Mach. Theory* **2019**, *138*, 149–168. [[CrossRef](#)]
14. Sabaapour, M.R.; Yoon, J. A novel method for optimal path synthesis of mechanisms based on tracking control of shadow robot. *Mech. Mach. Theory* **2019**, *131*, 218–233. [[CrossRef](#)]
15. Sancibrian, R.; Sedano, A.; Sarabia, E.G.; Blanco, J.M. Hybridizing differential evolution and local search optimization for dimensional synthesis of linkages. *Mech. Mach. Theory* **2019**, *140*, 389–412. [[CrossRef](#)]
16. Romero, N.N.; Campos, A.; Martins, D.; Vieira, R.S. A new approach for the optimal synthesis of four-bar path generator linkages. *SN Appl. Sci.* **2019**, *1*, 1–8. [[CrossRef](#)]
17. Bureerat, S.; Slesongsom, S. Constraint handling technique for four-bar linkage path generation using self-adaptive teaching-learning-based optimization with a diversity archive. *Eng. Optim.* **2021**, *53*, 513–530. [[CrossRef](#)]
18. Varedi-Koulaei, S.; Rezagholizadeh, H. Synthesis of the four-bar linkage as path generation by choosing the shape of the connecting rod. *Proc. Inst. Mech. Eng. Part C J. Mech. Eng. Sci.* **2020**, *234*, 2643–2652. [[CrossRef](#)]
19. Rao, R.V. Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems. *Int. J. Ind. Eng. Comput.* **2016**, *7*, 19–34. [[CrossRef](#)]
20. Frans, R.; Arfiadi, Y. Sizing, Shape, and Topology Optimizations of Roof Trusses Using Hybrid Genetic Algorithms. *Procedia Eng.* **2014**, *95*, 185–195. [[CrossRef](#)]
21. Sonmez, M. Performance Comparison of Metaheuristic Algorithms for the Optimal Design of Space Trusses. *Arab. J. Sci. Eng.* **2018**, *43*, 5265–5281. [[CrossRef](#)]
22. Ghavidel, S.; Azizivahed, A.; Li, L. A hybrid Jaya algorithm for reliability–redundancy allocation problems. *Eng. Optim.* **2017**, *50*, 698–715. [[CrossRef](#)]
23. Luo, X.; Cao, L.; Wang, L.; Zhao, Z.; Huang, C. Parameter identification of the photovoltaic cell model with a hybrid Jaya-NM algorithm. *Optik* **2018**, *171*, 200–203. [[CrossRef](#)]
24. Slesongsom, S.; Bureerat, S. Four-bar linkage path generation through self-adaptive population size teaching-learning based optimization. *Knowl.-Based Syst.* **2017**, *135*, 180–191. [[CrossRef](#)]
25. Storn, R.; Price, K. Differential Evolution—A Simple and Efficient Heuristic for global Optimization over Continuous Spaces. *J. Glob. Optim.* **1997**, *11*, 341–359. [[CrossRef](#)]
26. Ali, M.Z.; Awad, N.H.; Suganthan, P.N. Multi-population differential evolution with balanced ensemble of mutation strategies for large-scale global optimization. *Appl. Soft Comput.* **2015**, *33*, 304–327. [[CrossRef](#)]
27. Ho-Huu, V.; Nguyen-Thoi, T.; Nguyen-Thoi, M.; Le-Anh, L. An improved constrained differential evolution using discrete variables (D-ICDE) for layout optimization of truss structures. *Expert Syst. Appl.* **2015**, *42*, 7057–7069. [[CrossRef](#)]
28. Mirjalili, S.; Lewis, A. The Whale Optimization Algorithm. *Adv. Eng. Softw.* **2016**, *95*, 51–67. [[CrossRef](#)]
29. Angira, R.; Babu, B. Optimization of process synthesis and design problems: A modified differential evolution approach. *Chem. Eng. Sci.* **2006**, *61*, 4707–4721. [[CrossRef](#)]
30. Babu, B.; Angira, R. Modified differential evolution (MDE) for optimization of non-linear chemical processes. *Comput. Chem. Eng.* **2006**, *30*, 989–1002. [[CrossRef](#)]
31. Deng, C.; Zhao, B.; Deng, A.; Liang, C. Modified Differential Evolution for Hard Constrained Optimization. In Proceedings of the 2009 International Conference on Computational Intelligence and Software Engineering, Wuhan, China, 11–13 December 2009. [[CrossRef](#)]
32. Kaelo, P.; Ali, M. A numerical study of some modified differential evolution algorithms. *Eur. J. Oper. Res.* **2006**, *169*, 1176–1184. [[CrossRef](#)]
33. Maulik, U.; Saha, I. Modified differential evolution based fuzzy clustering for pixel classification in remote sensing imagery. *Pattern Recognit.* **2009**, *42*, 2135–2149. [[CrossRef](#)]
34. Mohamed, A.W.; Sabry, H.Z. Constrained optimization based on modified differential evolution algorithm. *Inf. Sci.* **2012**, *194*, 171–208. [[CrossRef](#)]
35. Sayah, S.; Zehar, K. Modified differential evolution algorithm for optimal power flow with non-smooth cost functions. *Energy Convers. Manag.* **2008**, *49*, 3036–3042. [[CrossRef](#)]
36. Lin, W.Y. A GA–DE hybrid evolutionary algorithm for path synthesis of four-bar linkage. *Mech. Mach. Theory* **2010**, *45*, 1096–1107. [[CrossRef](#)]
37. Trivedi, A.; Srinivasan, D.; Biswas, S.; Reindl, T. Hybridizing genetic algorithm with differential evolution for solving the unit commitment scheduling problem. *Swarm Evol. Comput.* **2015**, *23*, 50–64. [[CrossRef](#)]
38. Sayah, S.; Hamouda, A. A hybrid differential evolution algorithm based on particle swarm optimization for nonconvex economic dispatch problems. *Appl. Soft Comput.* **2013**, *13*, 1608–1619. [[CrossRef](#)]
39. Zhang, C.; Ning, J.; Lu, S.; Ouyang, D.; Ding, T. A novel hybrid differential evolution and particle swarm optimization algorithm for unconstrained optimization. *Oper. Res. Lett.* **2009**, *37*, 117–122. [[CrossRef](#)]

40. Lieu, Q.X.; Do, D.T.; Lee, J. An adaptive hybrid evolutionary firefly algorithm for shape and size optimization of truss structures with frequency constraints. *Comput. Struct.* **2018**, *195*, 99–112. [[CrossRef](#)]
41. Zheng, Y.J.; Xu, X.L.; Ling, H.F.; Chen, S.Y. A hybrid fireworks optimization method with differential evolution operators. *Neurocomputing* **2015**, *148*, 75–82. [[CrossRef](#)]
42. Qin, A.; Huang, V.; Suganthan, P. Differential Evolution Algorithm With Strategy Adaptation for Global Numerical Optimization. *IEEE Trans. Evol. Comput.* **2009**, *13*, 398–417. [[CrossRef](#)]
43. Padhye, N.; Bhardawaj, P.; Deb, K. Improving differential evolution through a unified approach. *J. Glob. Optim.* **2012**, *55*, 771–799. [[CrossRef](#)]
44. Kunjur, A.; Krishnamurty, S. A robust multi-criteria optimization approach. *Mech. Mach. Theory* **1997**, *32*, 797–810. [[CrossRef](#)]
45. Wolpert, D.H.; Macready, W.G. No free lunch theorems for optimization. *IEEE Trans. Evol. Comput.* **1997**, *1*, 67–82. [[CrossRef](#)]