

## Article

# Neuronal Constraint-Handling Technique for the Optimal Synthesis of Closed-Chain Mechanisms in Lower Limb Rehabilitation

José Saúl Muñoz-Reina <sup>1</sup>, Miguel Gabriel Villarreal-Cervantes <sup>1,\*</sup>, Leonel Germán Corona-Ramírez <sup>2</sup>  
and Luis Ernesto Valencia-Segura <sup>1,2</sup>

<sup>1</sup> Instituto Politécnico Nacional, Centro de Innovación y Desarrollo Tecnológico en Cómputo, Ciudad de Mexico 07738, Mexico; jmunozr1002@alumno.ipn.mx (J.S.M.-R.); lvalenciam1700@alumno.ipn.mx (L.E.V.-S.)

<sup>2</sup> Instituto Politécnico Nacional, Unidad Profesional Interdisciplinaria en Ingeniería y Tecnologías Avanzadas, Ciudad de Mexico 07340, Mexico; lgcorona@ipn.mx

\* Correspondence: mvillarrealc@ipn.mx

**Abstract:** The optimal methods for the synthesis of mechanisms in rehabilitation usually require solving constrained optimization problems. Metaheuristic algorithms are frequently used to solve these problems with the inclusion of Constraint-Handling Techniques (CHTs). Nevertheless, the most used CHTs in the synthesis of mechanisms, such as penalty function and feasibility rules, generally prioritize the search for feasible regions over the minimization of the objective function, and it notably influences the exploration and exploitation of the algorithm such that it could induce in the premature convergence to the local minimum and thus the solution quality could deteriorate. In this work, a Neuronal Constraint-Handling (NCH) technique is proposed and its performance is studied in the solution of mechanism synthesis for rehabilitation. The NCH technique uses a neural network to search for the fittest solutions into the feasible and the infeasible region to pass them to the next generation of the evolutionary process of the Differential Evolution (DE) algorithm and consequently improve the obtained solution quality. Two synthesis problems with four-bar and cam-linkage mechanisms are the study cases for developing lower-limb rehabilitation routines. The NCH is compared with four state-of-the-art constraint-handling techniques (penalty function, feasibility rules, stochastic ranking,  $\epsilon$ -constrained method) included into four representative metaheuristic algorithms. The irace package is used for both the algorithm settings and neuronal network training to fairly and meaningfully compare through statistics to confirm the overall performance. The statistical results confirm that, despite changes in the rehabilitation trajectories, the proposal presents the best overall performance among selected algorithms in the studied synthesis problems for rehabilitation, followed by penalty function and feasibility rule.

**Keywords:** constraint-handling technique; metaheuristics; neural networks; optimal synthesis; optimization; rehabilitation mechanism



**Citation:** Muñoz-Reina, J.S.; Villarreal-Cervantes, M.G.; Corona-Ramírez, L.G.; Valencia-Segura, L.E. Neuronal Constraint-Handling Technique for the Optimal Synthesis of Closed-Chain Mechanisms in Lower Limb Rehabilitation. *Appl. Sci.* **2022**, *12*, 2396. <https://doi.org/10.3390/app12052396>

Academic Editor: Vincent A. Cicirello

Received: 27 January 2022

Accepted: 23 February 2022

Published: 25 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

Nowadays, rehabilitation systems based on closed-chain mechanisms are a low-cost alternative for rehabilitation routines [1–3]. In the design of these systems, a mechanism synthesis process is made. The synthesis process consists of determining the link lengths to generate a rehabilitation trajectory in the mechanism coupler link [4]. Among the graphical, analytical, or optimal methods used in the synthesis of mechanisms [5], the optimal method is the most suitable to solve synthesis problems because several design objectives, constraints, and precision points can be handled as an optimization problem. However, the solution quality depends on the numerical methods (optimizers) that solves the optimization problem [6–8].

Numerical methods are frequently used to solve complex problems in the optimal synthesis of mechanisms. These can be classified into indirect and direct search methods [9]. The indirect search methods require derivatives of both the objective function and constraints presenting issues when the optimization problem is non-continuous and in the case of highly nonlinear spaces (the obtained solutions converge to the initial guessing). The most popular indirect search method to solve optimal synthesis problems is the Sequential Quadratic Programming (SQP) [10–14]. Otherwise, the direct search methods do not require a derivative of the objective function and constraints, and the solutions are obtained heuristically [15] allowing for searching for solutions in highly nonlinear or discontinuous spaces.

The direct search methods such as metaheuristic algorithms are currently used to solve the synthesis of mechanisms problems because they are based on a set of solutions called population such that the initial guessing does not influence the obtained solutions. The metaheuristic algorithms are inspired in biological systems as the Genetic Algorithm (GA) [16], Differential Evolution (DE), Malaga University and Mechanism Synthesis Algorithm (MUMSA) [17], in social phenomena as the Particle Swarm Optimization (PSO) [18], inspired by musical composition as the Harmony Search (HS) [19] or in physics as the Inclined Planes Optimization (IPO) [20]. Among these algorithms, GA, DE, PSO, and MUMSA are usually implemented to solve synthesis problems, finding better solutions than indirect search methods in the synthesis of mechanisms [3,7,21–23].

On the other hand, the original versions of metaheuristic algorithms solve unconstrained optimization problems. Nevertheless, the mechanism synthesis problems are formulated as optimization problems with inequality constraints requiring not only the selection of optimizers that search in high nonlinearity objective space, but also the ways of finding into the complex search shapes enclosed by constraints. Therefore, the optimizers and the ways of handling complex constraints are crucial. The efficiency of the optimizers to provide the most suitable solution quality depends on the way of handling constraints. In constrained optimization problems, Constraint-Handling Techniques (CHTs) have been incorporated [24–26]. In the solution of constrained optimization problems, there are five types of constraint-handling techniques; these are: penalty functions, special representations and operators, repair algorithms, separation of objective function and constraints, and hybrid methods [24]. Currently, the Penalty Function (PF), Feasibility Rules (FR), Stochastic Ranking (SR), and  $\epsilon$ -Constraint ( $\epsilon$ C) are popular constraint-handling techniques used in metaheuristic algorithms for the synthesis of mechanisms [27–29]. The PF transforms a constrained optimization problem into an unconstrained one [30]. In the PF, the constraints are penalized and incorporated into the objective function, and the penalty factor in constraints is usually high. Therefore, the search for feasible regions is prioritized over the objective function minimization, for instance, in [5,6,8,17,31–39]. The FR have three rules for handling constraints. In this method, the search for feasible regions is also prioritized over the objective function minimization [3,7,21,40,41]. The SR works by using a pseudo-random ordering of solutions where the first ones in the ranking are selected [29,42]. A probabilistic parameter controls the order of solutions to prioritize the search for feasible regions or the minimization of the objective function. The  $\epsilon$ C relaxes the constraints to increase the probability of selecting solutions close to feasible regions. This constraint-handling technique prioritizes the search for feasible regions over the minimization of the objective functions. The  $\epsilon$ C also has a gradient-based mutation mechanism to find feasible regions using the gradient of unfeasible solutions [29,43]. Through the empirical study of four CHTs (PF, FR, SR, and  $\epsilon$ C) for the solution of mechanism synthesis given in [29], it is confirmed that the most suitable solution quality depends on the way of handling constraints and consequently impacts the efficiency of optimizers. In that work, the feasibility rules usually led to efficient optimization in the solution of mechanism synthesis for rehabilitation.

The summary of metaheuristic algorithms and their constraint-handling techniques used in the mechanism synthesis problem in a broad context and for specific rehabilitation

purposes is given in Table 1. The PF and the FR are the most promising and widely used CHTs for the mechanism synthesis problem for rehabilitation. It is also observed that the performance of SR and  $\epsilon$ C have only been investigated a little. According to recent literature [44], the trend to improve the solution accuracy for the dimensional synthesis problem is to make modifications of metaheuristics by hybridizing different algorithms or modifying their structure, as they are shown in Table 1. However, the efficient and effective use of handling constraints based on neuronal networks has not been investigated in the dimensional synthesis problem for rehabilitation. This research direction can significantly influence the metaheuristic behavior.

**Table 1.** Summary of investigations using metaheuristic algorithms and constraint-handling techniques in mechanism synthesis problems in a broad context and for specific rehabilitation purposes.

Study	Mechanisms	Metaheuristic Algorithms	Constraint-Handling Technique
[5]	four-bar mechanism	Genetic Algorithm (GA)	Penalty Function (PF)
[41]	Hand robot mechanism	Pareto Optimum Evolutionary multi-objective Algorithm (POEMA)	Feasibility Rules (FR)
[31]	four-bar mechanism	Differential Evolution (DE)	PF
[32]	Six-bar mechanism	DE	PF
[33]	four-bar mechanism	DE	PF
[34]	four-bar mechanism	GA-fuzzy logic (GA-FL)	PF
[17]	four-bar and Six-bar mechanisms	Málaga University Mechanism Synthesis Algorithm (MUMSA)	PF
[35]	four-bar mechanism	GA, DE, Particle Swarm Optimization (PSO)	PF
[36]	four-bar mechanism	Ant-gradient (AG)	PF
[45]	four-bar mechanism	GA-DE	PF
[6]	Six-bar mechanism	Cuckoo Search (CS)	PF
[37]	four-bar mechanism	Imperialist Competitive Algorithm (ICA), GA, DE, PSO	PF
[8]	four-bar mechanism	Modified Krill Herd	PF
[46]	four-bar mechanism	Teaching-Learning-Based Optimization (TLBO), GA, PSO	PF
[38]	four-bar mechanism	Hybrid Lagrange Interpolation DE (HLIDE)	PF
[39]	four-bar and Six-bar mechanisms	Hybridization DE with Generalized Reduced Gradient	PF
[47]	four-bar mechanism	DE	FR
[48]	four-bar mechanisms	CS, TLBO, DE, MUMSA auto-adaptive modified DE, combined-mutation DE (CMDE)	-

Study	Mechanism in Rehabilitation	Metaheuristic Algorithms	CHT
[21]	Six-bar mechanism in finger rehabilitation	MUMSA	FR
[22]	cam-linkage mechanism in gait rehabilitation	GA	PF
[7]	four-bar mechanism in gait rehabilitation	DE	FR
[23]	four-bar mechanism in gait rehabilitation and orthotic devices	PSO, TLBO	-
[3]	Eight-bar mechanism in lower limb rehabilitation	DE	FR
[29]	Eight-bar, four-bar and cam-linkage mechanisms in lower limb rehabilitation	DE, PSO, MUMSA, GA	FR, PF, Stochastic-Ranking (SR), $\epsilon$ -Constraint ( $\epsilon$ C)

### 1.1. Contributions

The constraint-handling techniques used in the synthesis of mechanisms generally prioritize the search for feasible regions. According to the results in [29], it is observed that the fast convergence to feasible regions as presented in the penalty function, feasibility rules, and  $\epsilon$ -constraint increase the convergence probability to local minimums in synthesis problems. On the other hand, stochastic ranking controls the search by prioritizing feasible regions and minimizing the objective function through a probabilistic factor. However, the pseudo-random ordering of solutions in stochastic ranking decreases the metaheuristic algorithm capability to explore and exploit search space in the synthesis problem solution. Therefore, to decrease the convergence probability to a local minimum in metaheuristic algorithms, the incorporation of a neuronal network as a constraint-handling technique is presented in this work, and the proposed NCH technique represents the first contribution of the work. This technique searches for solutions in the infeasible or feasible regions based on prior knowledge. The neural network as a constraint-handling technique can find patterns in complex data sets to ensure the obtained solution quality, in particular, problems [49–52]. The proposed NCH technique is applied to the solution of two synthesis problems to develop lower limb rehabilitation routines. One of these mechanisms is the most common one in the literature, the four-bar linkage mechanism. The other mechanism is a new mechanism reported in [22] conformed by a cam and links called cam-linkage mechanism. Sixteen algorithms are selected to make a comparative analysis and confirm the NCH technique's performance. The sixteen algorithms are based on four metaheuristic algorithms using four popular constraint-handling techniques. Therefore, these algorithms and techniques are the most representative of mechanism synthesis problems in the state-of-the-art.

On the other hand, the stochastic parameters in algorithms produce every run, reaching different solutions. The algorithm parameters can significantly influence the performance of those algorithms and the quality of solutions [53,54]. Thus, unlike other research, this work takes into consideration the automatic parameter tuning of all algorithms with the aid of the irace package [55], to make a fair and meaningful comparative statistical analysis. The descriptive and inferential statistical analysis verifies the performance of the proposed NCH technique included into the DE/RAND/1/BIN through the convergence towards a region of the design space where the obtained solutions remain within a useful and competitive margin [56]. Therefore, the analysis provides enough information about the most suitable constraint-handling technique applied to algorithms for solving the mentioned problems and confirms the performance of the proposal with respect to the state-of-the-art CHTs. The fair and meaningful comparative statistical analysis represents the second contribution of this work.

### 1.2. Paper Organization

This work is organized as follows: Section 2 presents the NCH technique implemented in a DE variant called DE/RAND/1/BIN. Section 3 presents two study cases to test the proposed constraint-handling technique performance. The first study case is related to the synthesis of a four-bar linkage mechanism, and the second study case is associated with the synthesis of the cam-linkage mechanism. Section 4 analyzes the NCH technique in a DE algorithm with respect to four popular constraint-handling techniques (FR, SR, EC, and PF) included into four metaheuristic algorithms commonly used in mechanism synthesis (ED, GA, PSO, MUMSA). The analysis is based on descriptive statistics, confidence intervals and inferential statistics. Section 5 presents conclusions of the work.

## 2. Constraint-Handling Technique Based on a Neural Network for the Differential Evolution Algorithm

This work presents the constraint-handling technique based on a neural network. This technique is implemented in a DE variant called DE/RAND/1/BIN and is tested in the

solution of mechanism synthesis problems. In this work, those problems have more than one design objective, and they are formulated by using the weighted sum method [57].

Section 2.1 presents the general statement of the mechanism synthesis problem as a mono-objective optimization problem. Section 2.2 explains the operation of the DE variant DE/RAND/1/BIN, and Section 2.3 shows the proposed constraint-handling technique based on a neural network.

2.1. Statement of the Mechanism Synthesis Problem

Optimization problems for the synthesis of mechanisms usually require satisfying several design objectives and constraints. Multi-objective or mono-objective optimization techniques can solve design problems. Nowadays, mono-objective optimization techniques are commonly used for solving mechanism synthesis problems, where a multi-objective problem is transformed into a mono-objective problem by the weighted sum approach. Therefore, the mechanism synthesis problem related to this work is expressed as follows.

$$\text{Min}_{x^*} \bar{J}(x) \tag{1}$$

subject to:

$$g_j(x) \leq 0 \quad \forall j = \{1, 2, \dots, m\} \tag{2}$$

$$h_k(x) = 0 \quad \forall k = \{1, 2, \dots, r\} \tag{3}$$

$$x_{min} \leq x \leq x_{max} \tag{4}$$

where  $\bar{J}(x) = \sum_{i=1}^n w_i J_i(x)$  is the weighted objective function,  $J_i$  is the  $i$ -th term associated with a design performance index,  $w_i$  is the  $i$ -th weight attributed to each term,  $x \in R^q$  is the design variable vector,  $g_j(x)$  and  $h_k(x)$  are the inequality and equality constraints, respectively, and  $[x_{min}, x_{max}] \in R^q$  are the design variable vector bounds.

2.2. General Overview of the Differential Evolution Algorithm

In this work, the optimization strategy for the solution of synthesis problems is based on the differential evolution variant called DE/RAND/1/BIN [58]. The term "DE" in the name of the variant DE/RAND/1/BIN relates to the Differential Evolution algorithm. The term "RAND" indicates the inclusion of a random mutation with one "1" pair of individuals, meaning that the base vector is randomly selected from the population and one difference vector is added. The last term "BIN" implies the incorporation of a binomial crossover.

DE/RAND/1/BIN initializes with a population  $\mathbf{X}^G = \{X_1^G, X_2^G, \dots, X_{NP}^G\}$  in the generation  $G = 0$  with  $NP$  individuals. Each individual  $X_p^G \forall p = \{1, 2, \dots, NP\}$  in the population  $\mathbf{X}^G$  has  $D$  genotypic characteristics represented by the design variable  $x_{p,q}$  i.e.,  $X_p^G = [x_{p,1}, x_{p,2}, \dots, x_{p,q}] \forall q = \{1, 2, \dots, D\}$ . In the initial generation  $G = 0$ , the design variable vector is determined by random values in the range  $[X_{min}, X_{max}]$ .

The individuals in the population  $\mathbf{X}^G$  undergo the mutation and crossover processes to generate a population of offspring individuals  $\mathbf{U}^G = \{U_1^G, U_2^G, U_3^G, \dots, U_p^G\}$  in each generation  $G$ . These processes aim to improve the fitness of individuals  $\mathbf{X}^G$  for the next generation of parents  $\mathbf{X}^{G+1}$ . The mutation and crossover processes of the variant DE/RAND/1/BIN are detailed in (5):

$$u_{p,q}^G = \begin{cases} \overbrace{\left( \underbrace{x_{r_3,q}^G}_{\text{Base vector}} + F \left( \underbrace{x_{r_1,q}^G - x_{r_2,q}^G}_{\text{Difference vector}} \right) \right)}^{\text{Mutation}}}_{\text{Crossover}} & \text{If } \text{rand}(0, 1) < CR \text{ or } q = q_{rand} \\ x_{p,q}^G & \text{Otherwise} \end{cases} \tag{5}$$

where  $CR \in [0, 1]$  is the crossover factor,  $F \in [0, 1]$  is a scale factor, and  $r_1, r_2$  and  $r_3$  are random numbers that represent the information of three individuals; these must be different from the individual  $p$  and, among them, and  $q_{rand} \in \{1, 2, \dots, D\}$  is a random value that controls the crossover point.

Each new offspring's individual  $U_p^G$  is compared with their parent  $X_p^G$ . Both individuals compete and only the fittest individuals pass on to the next generation  $G + 1$ . In this work, the selection between the parent and the offspring vectors is based on the Neuronal Constraint-Handling (NCH) technique. This technique is described in the following subsection.

### 2.3. Neuronal Constraint-Handling Technique

The Neuronal Constraint-Handling (NCH) technique is based on a feed-forward artificial neural network [59]. This technique selects the most suitable individual between two existing individuals  $X_p^G$  and  $U_p^G$ . The best individual for the next population  $X^{G+1}$  is obtained considering the value of the objective functions  $\bar{J}(x)$  and the sum of infeasible constraint distance  $\phi(x)$  given by (6).

$$\phi(x) = \sum_{j=1}^m \max(0, g_j(x)) \tag{6}$$

The best individual between  $X_p^G$  and  $U_p^G$  is defined based on the neural network architecture. The operation of the NCH technique with DE/RAND/1/BIN algorithm is presented in the block diagram shown in Figure 1. The NCH technique can be divided in three processes to select solutions. In the first process, the value of the objective function  $\bar{J}$  and the value of the sum of infeasible constraint distance  $\phi$  of two individuals are normalized in the range  $[0,1]$ , using the maximum objective function value ( $\max(\bar{J}(X^G))$ ) and the maximum sum of infeasible constraint distance ( $\max(\phi(X^G))$ ) found in the population  $X^G$ . Each normalized value is introduced in the input layer of neurons consisting of four neurons  $a_1^{s=1}, a_2^{s=1}, a_3^{s=1},$  and  $a_4^{s=1}$ . The input neuron values are defined by (7)–(10).

$$a_1^1 = \frac{\bar{J}(X_p^G)}{\max(\bar{J}(X^G))} \tag{7}$$

$$a_2^1 = \frac{\phi(X_p^G)}{\max(\phi(X^G))} \tag{8}$$

$$a_3^1 = \frac{\bar{J}(U_p^G)}{\max(\bar{J}(U^G))} \tag{9}$$

$$a_4^1 = \frac{\phi(U_p^G)}{\max(\phi(U^G))} \tag{10}$$

In the second process, the neuron values of the input layer go to a hidden layer of neurons, where the inputs are analyzed according to a selection rule learned a priori. Hence, the values in the input layer goes to the neurons in the hidden layer  $a_r^s \forall r = \{1, 2, \dots, n_{f_s}\} \wedge s = \{2, 3, \dots, c - 1\}$ , where  $c$  is the number of layers and  $n_{f_s}$  is the number of neurons in the  $s$  layer. In this process, the user can select the number of layers and neurons. The third process selects the best individual between  $X_p^G$  and  $U_p^G$ . One neuron  $a_1^{s=c}$  is established in the output layer, and then, with this information, the best individual is given by the selector (11):

$$X_p^{G+1} = \begin{cases} X_p^G & \text{if } a_1^c < 0 \\ U_p^G & \text{if } a_1^c \geq 0 \end{cases} \tag{11}$$

As can be seen in Figure 1, the neurons  $a_r^s$  in the NCH technique receive the values of previous neurons, but these input values do not go directly to the next neuron. These input

values are introduced using neuronal weights  $\bar{w}_{t,r}^s \in [-100, 100] \forall t = \{1, 2, \dots, n_{f_{s-1}}\}$ , as shown in Figure 2.

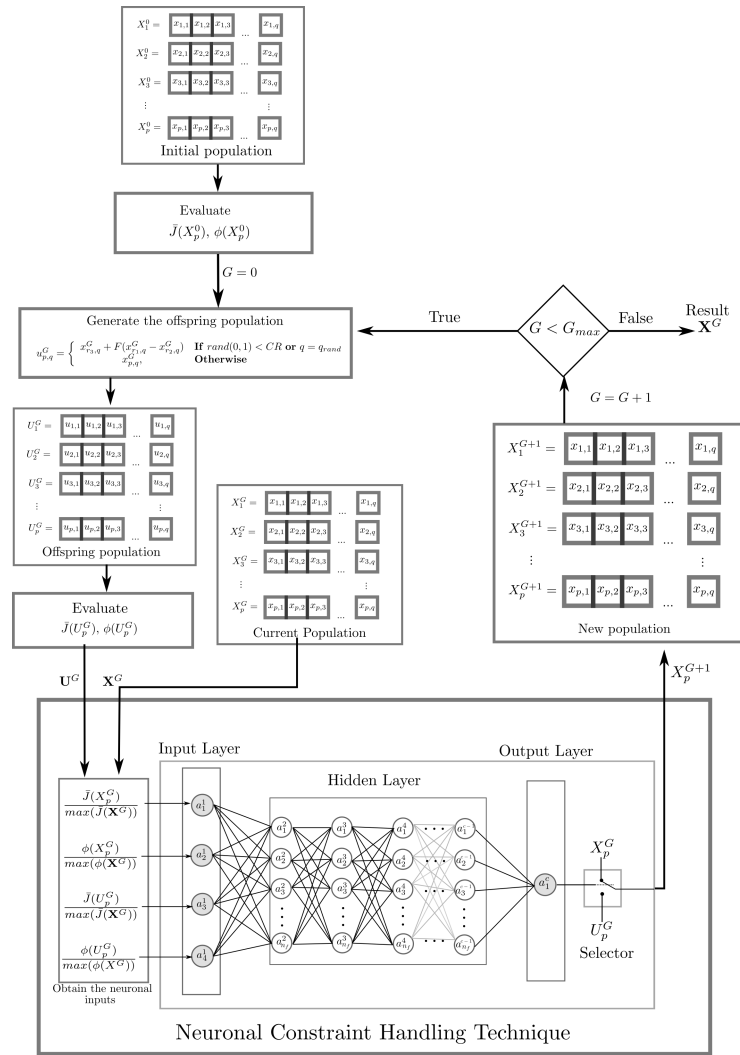


Figure 1. Neuronal constraint-handling technique.

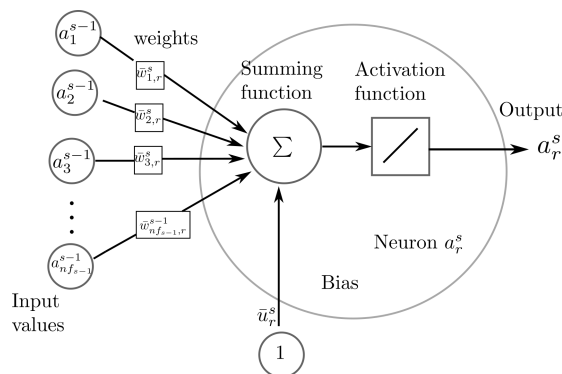


Figure 2. Artificial neuron or node in the NCH technique.

The output value of the neuron  $a_r^s$  is obtained using (12), where  $\bar{u}_r^s \in [-100, 100]$  is a bias value,  $n_{f_{s-1}}$  is the number of neurons in the  $s - 1$  layer, and  $f(\mu) \in [-1, 1]$  is a sigmoid activation function given by (13):

$$a_r^s = f\left(\sum_{t=1}^{n_{f_{s-1}}} (\bar{w}_{t,r}^s a_t^{s-1}) + \bar{u}_r^s\right) \tag{12}$$

$$f(\mu) = \frac{1 - e^{-\mu}}{1 + e^{-\mu}} \tag{13}$$

Algorithm 1 shows the operation of DE/RAND/1/BIN algorithm with the neuronal constraint-handling technique. The proposal is called R1B-NCH.

---

**Algorithm 1** Pseudo-code of the DE/RAND/1/BIN with the inclusion of the NCH technique (R1B-NCH).

---

- 1: Generate an initial population  $\mathbf{X}^0$  with  $NP$  individuals.
  - 2: Evaluate  $\mathbf{X}^0$ .
  - 3:  $G \leftarrow 0$
  - 4: **while**  $G \leq G_{max}$  **do**
  - 5:   **for all**  $X_p^G \in \mathbf{X}^G$  **do**
  - 6:     Generate a child individual  $U_p^G$  based on (5)
  - 7:     Evaluate the fitness of  $U_p^G$ .
  - 8:     Determine for the next generation, the individual  $X_p^{G+1}$  according to the proposed NCH technique (11) between  $X_p^G$  and  $U_p^G$ .
  - 9:   **end for**
  - 10:  $G \leftarrow G + 1$
  - 11: **end while**
  - 12: Select the best individual in the last population i.e.,  $x^* = Best(\mathbf{X}^{G_{max}})$  in the optimization problem (1)–(4).
- 

It is important to note that the selection rule in the NCH technique is based on the neuronal weights  $\bar{w}_{t,r}^s$  and bias  $\bar{u}_r^s$ . Therefore, those weight values must (train) to develop a suitable selection process in the synthesis process of mechanisms.

### 3. Study Cases

This work considers two study cases to test the DE/RAND/1/BIN algorithm with the Neuronal Constraint-Handling technique. The first study case is related to the synthesis of a four-bar linkage mechanism [7,11], and the second study case is associated with the synthesis of the cam-linkage mechanism for lower limb rehabilitation [22]. Both linkage mechanisms are synthesized to follow paths in the coupler point.

#### 3.1. Case 1: Four-Bar Linkage Mechanism

The conceptual design model for the four-bar linkage mechanism is presented in Figure 3, where  $r_1, r_2, r_3,$  and  $r_4$  are the link lengths,  $\theta_1$  is the ground link angle,  $[x_0, y_0]$  is the Cartesian position of the mechanism with respect to the coordinate system  $x - y,$  and  $\theta_2^i \forall i = \{1, 2, \dots, n_f\}$  is the  $i$ -th crank angle.



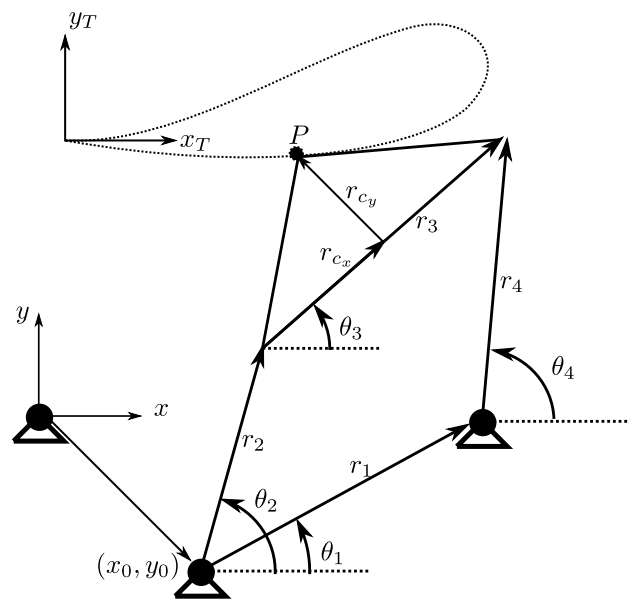


Figure 3. four-bar linkage mechanism.

The dimensional synthesis problem for this mechanism consists of finding the link lengths that approximate the coordinate  $[x_p^i, y_p^i]$  of the coupler link point  $P$  to the coordinates  $[\bar{x}_p^i, \bar{y}_p^i]$  of the rehabilitation trajectory. Figure 3 shows the design variable vector, and it is defined in (14):

$$x = [r_1, r_2, r_3, r_4, \theta_1, x_0, y_0, r_{cx}, r_{cy}, \theta_2^1, \theta_2^2, \dots, \theta_2^{n_f}] \tag{14}$$

The dimensional synthesis optimization problem is presented in (15)–(19), where the constant values  $w_1 = 1$  and  $w_2 = 0.01$  weight the design goals:

$$\text{Min}_x w_1 \sum_{i=1}^{n_f} [(x_p^i - \bar{x}_p^i)^2 + (y_p^i - \bar{y}_p^i)^2] + w_2 \sum_{i=1}^{n_f} (\theta_2^i - \bar{\theta}_2^i)^2 \tag{15}$$

subject to:

$$g_1(x) \leq 0 : r_2 + r_1 - r_3 - r_4 \leq 0 \tag{16}$$

$$g_2(x) \leq 0 : -r_4 - r_1 + r_2 + r_3 \leq 0 \tag{17}$$

$$g_3(x) \leq 0 : -r_3 - r_1 + r_2 + r_4 \leq 0 \tag{18}$$

$$x_{min} \leq x \leq x_{max} \tag{19}$$

The first design goal minimizes the error between the desired precision points  $[\bar{x}_p^i, \bar{y}_p^i]$  and the path  $[x_p^i, y_p^i]$  of the mechanism’s coupler link point. The second design goal minimizes the angular displacement error between the desired angles  $\bar{\theta}_2^i$  and the crank angles  $\theta_2^i$ . In this problem, the desired angles  $\bar{\theta}_2^i$  are established by (20):

$$\bar{\theta}_2^i = \theta_2^1 + 2\pi \frac{i-1}{n_f-1} \quad \forall i = \{1, 2, \dots, n_f\} \tag{20}$$

The  $i$ -th desired precision point  $[\bar{x}_p^i, \bar{y}_p^i]$  is established based on the ankle trajectory for rehabilitation routine. In this work, two trajectories are considered for the synthesis process representing a change in the human anatomy. The Pearson correlation between both trajectories is 95%. The trajectory points are shown in Figure 4, where Figure 4a,b are considered as trajectory 1 and trajectory 2, respectively, for the synthesis process. The Cartesian coordinates of the trajectory points are shown in Table A1.

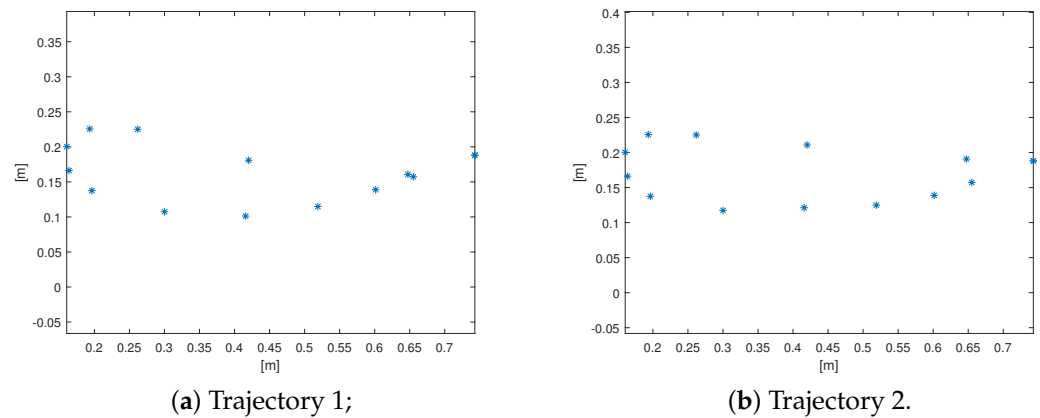


Figure 4. four-bar linkage precision points for the synthesis process.

The synthesis problem has three inequality constraints (16)–(18). These constraints define a crank-rocker mechanism according to the Grashof criterion. The design variable vector  $x \in R^{23}$  for the four-bar linkage mechanism is bounded by  $x_{min}$  and  $x_{max}$  limits. These limits are established according to [29].

### 3.2. Case 2: Cam-Linkage Mechanism

The second optimization problem is the synthesis of a cam-linkage mechanism. The design problem was presented by Shao et al. [22]. The conceptual model for the cam-linkage mechanism is shown in Figure 5, where  $r_1, r_2, \dots, r_8$  are the link lengths;  $\beta, \gamma, \eta$  are related to angular displacements;  $[x_0, y_0]$  is the Cartesian position of the mechanism with respect to the  $x - y$  coordinate system;  $e$  and the angle  $\alpha$  define the displacement position of the slider.

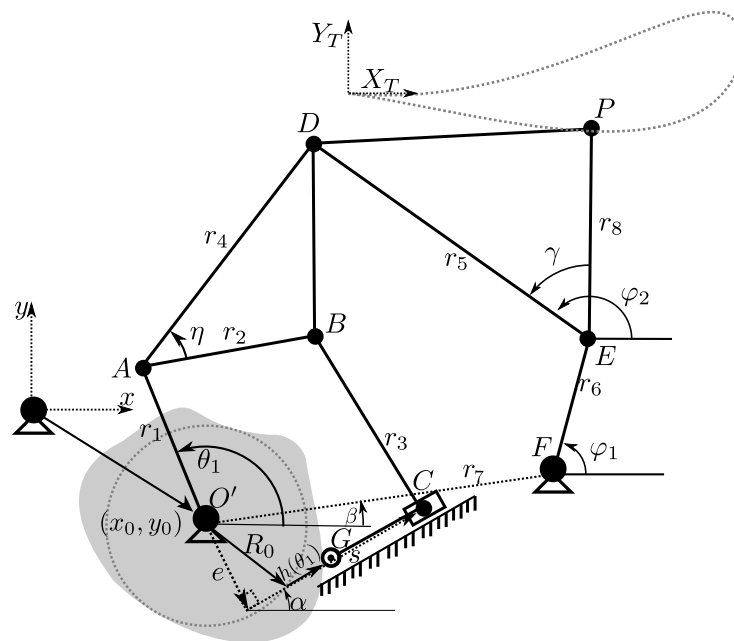


Figure 5. cam-linkage mechanism.

The dimensional synthesis problem for this mechanism consists of finding the link lengths and the cam profile that approximate the coordinates  $[x_p^i, y_p^i]$  of the coupler link point  $P$  to the coordinates  $[\bar{x}_p^i, \bar{y}_p^i]$  of the rehabilitation trajectory. In this work, two rehabilitation trajectories are considered for the synthesis process representing a change in the human anatomy. The Pearson correlation between both trajectories is 97%. The trajectory points are shown in Figure 6, where Figure 6a,b are considered as trajectory 1 and trajectory

2, respectively, for the synthesis process. The Cartesian coordinates of the trajectory points are shown in Tables A2 and A3.

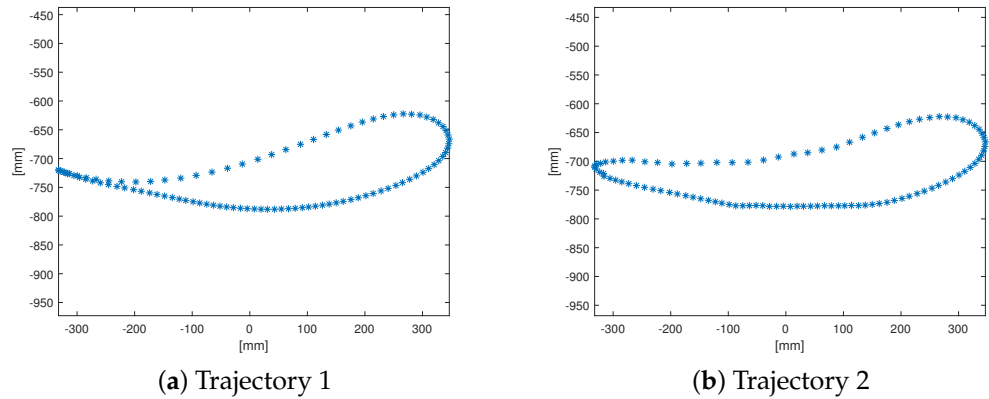


Figure 6. cam-linkage precision points for the synthesis process.

In this synthesis problem, the design variable vector is given by (21):

$$x = [r_1, r_2, r_3, r_4, r_5, r_6, r_7, r_8, \alpha, \beta, \gamma, \eta, x_0, y_0, e] \tag{21}$$

The dimensional synthesis optimization problem is presented in (22)–(34).

$$\text{Min}_x w_1 \frac{\sum_{i=1}^{n_f} (\theta_1^i - \bar{\theta}_1^i)^2}{\Theta_{ref}} + w_2 \frac{R_0}{R_{ref}} \tag{22}$$

subject to:

$$g_1 : r_1 + r_2 - r_3 - \sqrt{s_{min}^2 + e^2} \leq 0 \tag{23}$$

$$g_2 : r_1 + r_3 - r_2 - \sqrt{s_{min}^2 + e^2} \leq 0 \tag{24}$$

$$g_3 : r_1 + \sqrt{s_{max}^2 + e^2} - r_2 - r_3 \leq 0 \tag{25}$$

$$g_4 : r_2^2 + r_3^2 - (d_{AC})_{min}^2 - 2r_2r_3 \cos \pi/6 \leq 0 \tag{26}$$

$$g_5 : -r_2^2 - r_3^2 + (d_{AC})_{max}^2 + 2r_2r_3 \cos(\pi - \pi/6) \leq 0 \tag{27}$$

$$g_6 : r_5^2 + r_6^2 - (d_{DF})_{min}^2 - 2r_5r_6 \cos \pi/6 \leq 0 \tag{28}$$

$$g_7 : -r_5^2 - r_6^2 + (d_{DF})_{max}^2 + 2r_5r_6 \cos(\pi - \pi/6) \leq 0 \tag{29}$$

$$g_8 : (d_{OD})_{max} - r_4 - r_1 \leq 0 \tag{30}$$

$$g_9 : |r_4 - r_1| - (d_{OD})_{min} \leq 0 \tag{31}$$

$$g_{10} : (d_{Be})_{max} - r_3 \leq 0 \tag{32}$$

$$g_{11} : 0.12R_0 - 0.6|\rho|_{min} \leq 0 \tag{33}$$

$$x_{min} \leq x \leq x_{max} \tag{34}$$

The first term of the weighted objective function (22) minimizes the error between the crank angles  $\theta_1^i$  and the desired angles  $\bar{\theta}_1^i \forall i = \{1, 2, \dots, n_f\}$ , where  $n_f$  is the number of precision points, and  $\bar{\theta}_1^i$  is defined by (35):

$$\bar{\theta}_1^i = \theta_1^1 + 2\pi \frac{i-1}{n_f-1} \forall i = \{1, 2, \dots, n_f\} \tag{35}$$

The second term minimizes the base radius  $R_0$  of the cam (see Figure 5). Both design objectives are weighted by  $w_1 = 0.3$  and  $w_2 = 0.7$ , and normalized by  $\Theta_{ref} = 0.4246$  [rad] and  $R_{ref} = 0.3782$  [m] according to [22].

On the other hand, the first three inequality constraints (23)–(25) guarantee a complete revolution of the crank angle  $\theta_2$  (Grashof criterion), where  $s_{min}$  and  $s_{max}$  are the minimum and maximum values of  $s(\theta_1)$  in the precision points. The inequality constraints (26)–(29) provide high efficiency of the force transmission from the input link (crank link) to the output link, where  $d_{AC}$  is the distance between points A and C, and  $d_{DF}$  is the distance between points D and F. The inequality constraints (30)–(32) guarantee the path feasibility in the output link, where  $d_{OD}$  is the distance between points O and D, and  $d_{Be}$  is the distance between points B and e. The inequality constraint (33) defines the minimum curvature radius of the cam profile. The design variable vector  $x$  is bounded by  $x_{min}$  and  $x_{max}$ . For more details about the constraints and boundary limits, see [29].

#### 4. Results

In this section, the behavior of the DE/RAND/1/BIN algorithm with the NCH technique, named R1B-NCH, is compared with four popular constraint-handling techniques included in four metaheuristic algorithms. The metaheuristic algorithms involve the Differential Evolution (DE) variant DE/RAND/1/BIN, Genetic Algorithm (GA), Particle Swarm Optimization (PSO), and the Malaga University Mechanism Synthesis Algorithm (MUMSA). Moreover, Feasibility Rules (FR), Stochastic-Ranking (SR),  $\epsilon$ -Constrained ( $\epsilon$ C), and Penalty Function (PF) are considered as constraint-handling techniques. Therefore, the R1B-NCH algorithm is compared with sixteen different algorithms in the solution of study cases.

As was mentioned in previous sections, two different trajectories in each study case are considered. The good parameter setting of optimization algorithms (algorithm parameter tuning) are found using only trajectory 1 for each study case. The irace package carries out the algorithm parameter's tuning process to get significant results in the comparative analysis process. On the hand, the NCH technique is trained with the same package considering the parameter setting obtained previously and trajectory 1 for each study case.

Once the algorithm parameter setting is obtained and the neuronal constraint handling is trained, the second trajectory in the mechanism synthesis problem is used to show the algorithm performance under changes in the trajectory. This test provides information about the capability of the NCH technique to handle variations in the optimization problem that were considered neither in the algorithm tuning process nor the training process. This second trajectory provides different data than the trained data set provided by the first trajectory, and then this test evaluates the ability of the NCH to handle a data set that was not considered in the training process and the capacity to produce a suitable response in the output of the NCH.

This section is divided into three subsections: the first one presents the experiment conditions obtained by the algorithm tuning and training process. The second one shows algorithm performance analysis per each study case using descriptive, confidence intervals, and inferential statistics. Finally, in the third one, the overall evaluation of the algorithms is discussed to confirm the performance of the proposed NCH technique included in the DE/RAND/1/BIN.

##### 4.1. Experiment Conditions: Algorithm Parameter Tuning and Neuronal Constraint Handling Training Process

This section presents the parameter tuning of algorithms and also the training process of the proposed NCH technique. Those processes are applied at each study case and consider the first trajectory.

Four metaheuristic algorithms (DE/RAND/1/BIN, GA, PSO, and MUMSA) with the use of four constraint-handling techniques (FR, SR,  $\epsilon$ C and PF) are considered in the tuning process. The irace [55] package version 3.3 implemented in the R software version 3.6.2 is used in the parameter setting of algorithms. The irace package finds the most suitable setting of the algorithm parameters given a specific optimization problem through a set of automatic configuration procedures in a particular problem. At each

iteration of configuration procedures, it maximizes the algorithms’ performance across difference demands and uses either the non-parametric Friedman test or the paired *t*-test to identify worse configurations to be discarded. When these procedures end, the best algorithm setting results in a much higher-performing optimization algorithm, and it can be used to solve the problem. For the detailed explanation of the irace procedure, consult [60]. The maximum number of algorithm executions in the irace package for finding the algorithm parameter setting is set as 50,000. Each algorithm in one execution includes a stop criterion based on the objective function evaluation. In this case, 300,000 evaluations of the objective function are considered. The maximum number was empirically chosen through a series of trial and error procedures where the trade-off between the computational time and the convergence to a suitable solution is considered.

The algorithm parameter setting obtained by the irace package is presented in Table 2 for the first study case and Table 3 for the second one.

The training of the NCH technique into the DE/RAND/1/BIN consists of finding the neural weights  $\bar{w}_{i,r}^s$  and the bias  $\bar{u}_r^s$  by using the irace package. In the training process of the NCH technique, the crossover factor *CR* and the scale factor limits  $F_{min}$ ,  $F_{max}$  are fixed with the same configuration of the algorithm DE/RAND/1/BIN using the Feasibility Rules (FR) constraint handling technique (see Tables 2 and 3).

Four neurons in the input layer ( $a_r^1 \forall r = \{1, 2, 3, 4\}$ ), three neurons in three hidden layers ( $a_r^s \forall r = \{1, 2, 3\} \wedge s = \{2, 3, 4\}$ ), and one neuron in the output layer ( $a_1^5$ ) are considered. The number of neurons in each layer was obtained by the trial and error procedure where different numbers of neurons are compared to maximize the algorithm efficiency. Based on that set of trials, three layers of neurons are recommended, with three neurons in each layer for the study cases. The neural weights  $\bar{w}_{i,r}^s$  and the bias  $\bar{u}_r^s$  for each study case are shown in Table 4.

**Table 2.** Algorithm parameter setting obtained by an irace package in study case 1.

Algorithm	CHT	Parameters
DE/RAND/1/BIN	FR	$CR = 0.95, F_{min} = 0.13, F_{max} = 0.95$
	SR	$CR = 0.91, F_{min} = 0.23, F_{max} = 0.96, Pf = 0.15$
	EC	$CR = 0.89, F_{min} = 0.20, F_{max} = 0.81, Pg = 0.07, Tc = 590, Rg = 5, cp = 6$
	PF	$CR = 0.95, F_{min} = 0.17, F_{max} = 0.91, \omega_k = 10,000$
GA	FR	$CR = 0.80, MR = 0.06$
	SR	$CR = 0.23, Pf = 0.20, MR = 0.05$
	EC	$CR = 1, Pg = 0.05, Tc = 610, Rg = 4, cp = 8, MR = 0.08$
	PF	$CR = 0.83, MR = 0.11, \omega_k = 10,000$
PSO	FR	$v_{min} = 0.0, v_{max} = 0.01, C1 = 1.29, C2 = 2.04$
	SR	$Pf = 0.69, v_{min} = 0.0, v_{max} = 0.01, C1 = 2.22, C2 = 1.06$
	EC	$Pg = 0.05, Tc = 610, Rg = 4, cp = 8, v_{min} = 0.0, v_{max} = 0.01, C1 = 1.10, C2 = 1.79$
	PF	$v_{min} = 0.05, v_{max} = 0.17, C1 = 2.04, C2 = 1.06, \omega_k = 10,000$
MUMSA	FR	$CR = 0.73, F_{min} = 0.34, F_{max} = 0.78, R = 0.87, MR = 0.05$
	SR	$CR = 0.81, F_{min} = 0.69, F_{max} = 0.79, R = 0.73, MR = 0.08, Pf = 0.31$
	EC	$CR = 0.93, F_{min} = 0.71, F_{max} = 0.99, R = 0.75, MR = 0.03, Pg = 0.08, Tc = 257, Rg = 3, cp = 9$
	PF	$CR = 0.02, F_{min} = 0.13, F_{max} = 0.78, R = 0.04, MR = 0.02, \omega_k = 10,000$

*CR* is the crossover factor,  $F_{min}$  and  $F_{max}$  are the maximum and minimum scale factor limits. *MR* is the mutation rate. *Pf* is a probabilistic factor in SR. *cp* is a parameter to control the speed of reducing relaxation of constraints.  $T_c$  is the maximum number of iterations (generations or time) to relax constraints. *RG* is the number of attempts to improve a solution, and *Pg* is a probabilistic factor to improve a solution using the gradient-based mutation in  $\epsilon C$ .

**Table 3.** Algorithm parameter setting obtained by the irace package in study case 2.

Algorithm	CHT	Parameters
DE/RAND/1/BIN	FR	$CR = 0.87, F_{min} = 0.51, F_{max} = 0.53$
	SR	$CR = 0.91, F_{min} = 0.46, F_{max} = 0.70, Pf = 0.52$
	EC	$CR = 0.92, F_{min} = 0.53, F_{max} = 0.56, Pg = 0.04, Tc = 430, Rg = 5, cp = 9$
	PF	$CR = 0.93, F_{min} = 0.52, F_{max} = 0.63$
GA	FR	$CR = 0.98, MR = 0.11$
	SR	$CR = 0.57, Pf = 0.12, MR = 0.17$
	EC	$CR = 1, Pg = 0.08, Tc = 640, Rg = 4, cp = 6, MR = 0.14$
	PF	$CR = 0.02, Pg = 0.21$
PSO	FR	$v_{min} = 0.05, v_{max} = 0.24, C1 = 0.29, C2 = 3.12$
	SR	$Pf = 0.48, v_{min} = 0.04, v_{max} = 0.23, C1 = 1.60, C2 = 1.07$
	EC	$Pg = 0.06, Tc = 600, Rg = 2, cp = 2, v_{min} = 0.04, v_{max} = 0.26, C1 = 1.90, C2 = 0.86$
	PF	$v_{min} = 0.12, v_{max} = 0.34, C1 = 2.47, C2 = 0.34$
MUMSA	FR	$CR = 0.99, F_{min} = 0.78, F_{max} = 0.86, R = 0.31, MR = 0.03$
	SR	$CR = 0.88, F_{min} = 0.71, F_{max} = 0.74, R = 0.89, MR = 0.03, Pf = 0.41$
	EC	$CR = 0.91, F_{min} = 0.82, F_{max} = 0.85, R = 0.83, MR = 0.03, Pg = 0.03, Tc = 340, Rg = 2, cp = 5$
	PF	$CR = 0.98, F_{min} = 0.66, F_{max} = 0.73, R = 0.89, MR = 0.12$

CR is the crossover factor,  $F_{min}$  and  $F_{max}$  are the maximum and minimum scale factor limits. MR is the mutation rate. Pf is a probabilistic factor in SR. cp is a parameter to control the speed of reducing relaxation of constraints. Tc is the maximum number of iterations (generations or time) to relax constraints. Rg is the number of attempts to improve a solution, and Pg is a probabilistic factor to improve a solution using the gradient-based mutation in eC.

**Table 4.** Algorithm parameters of the NCH obtained by the irace package.

Parameter	Four-Bar	Cam-Linkage	Parameter	Four-Bar	Cam-Linkage
$w_{1,1}^1$	-5.1507	-9.7041	$w_{3,2}^3$	-4.5326	-1.7223
$w_{2,1}^1$	5.7785	4.6228	$w_{1,1}^4$	-6.4775	-9.5093
$w_{3,1}^1$	7.5795	-9.6850	$w_{2,1}^4$	0.7554	-3.7442
$w_{4,1}^1$	5.2518	-3.5547	$w_{3,1}^4$	0.8913	0.0632
$w_{1,2}^1$	8.4485	1.6000	$w_{1,2}^4$	-5.6703	-6.2279
$w_{2,2}^1$	-4.8087	5.5538	$w_{2,2}^4$	5.7473	-5.4144
$w_{3,2}^1$	6.4018	-4.7216	$w_{3,2}^4$	-2.1010	-1.4408
$w_{4,2}^1$	-0.0997	4.5662	$w_{1,1}^5$	5.0501	-5.8695
$w_{1,3}^1$	-6.4759	9.3248	$w_{2,1}^5$	2.7255	-4.0260
$w_{2,3}^1$	-6.9908	0.6948	$w_{3,1}^5$	-5.471	-0.5826
$w_{3,3}^1$	6.9994	-9.1374	$u_1^1$	5.0333	-5.8695
$w_{4,3}^1$	0.9477	-5.2051	$u_2^1$	2.7255	-4.0260
$w_{1,1}^2$	6.4335	-7.9333	$u_3^1$	-5.471	-0.5826
$w_{2,1}^2$	6.7337	-0.4223	$u_1^2$	5.0333	7.6293
$w_{3,1}^2$	9.1879	3.7733	$u_2^2$	-3.2977	-7.3464
$w_{1,2}^2$	2.2342	7.7584	$u_3^2$	-0.5276	-0.2524
$w_{2,2}^2$	-6.4004	-8.9817	$u_1^3$	4.4916	-5.0776
$w_{3,2}^2$	0.2861	-6.4264	$u_2^3$	6.1729	2.9905
$w_{1,1}^3$	7.0605	-7.6129	$u_3^3$	-9.4461	-9.8085
$w_{2,1}^3$	-3.3614	8.5456	$u_1^4$	-3.4358f	-6.9378
$w_{3,1}^3$	3.2812	-3.3421	$u_2^4$	0.8191	-4.6794
$w_{1,2}^3$	-4.3912	2.1478	$u_3^4$	-1.1605	4.4777
$w_{2,2}^3$	8.4263	7.2631	$u_1^5$	-1.7526	-2.6718
$F_{max}$	0.95	0.53	$F_{min}$	0.13	0.51
CR	0.95	0.87			

4.2. Algorithm Performance Analysis

Thirty executions are carried out to evaluate the metaheuristic algorithm performance (DE/RAND/1/BIN, GA, PSO, MUMSA) for each constraint-handling technique (FR, SR,

$\epsilon$ C, PF, NCH). The acronyms given by the sixteen algorithms are related to the algorithms' acronym, followed by the constraint-handling technique through a dash. For instance, in the case of algorithms related to the DE/RAND/1/BIN, the acronyms are expressed as R1B-FR, R1B-SR, R1B- $\epsilon$ C, and R1B-PF.

The algorithm parameters presented in Table 2, and Tables 3 and 4 are considered in the executions. Sixty individuals, particles, or chromosomes ( $NP = 60$ ) and 600,000 objective function evaluations are established as a stop condition for each algorithm.

The results obtained by each algorithm consider the thirty best objective function values obtained through the thirty executions (one value per each execution). This set of values conforms to one sample for the statistical analysis described in this section for a particular algorithm.

In this section, the samples of all algorithms are analyzed by using descriptive statistics, confidence intervals, and inferential statistics.

#### 4.2.1. Descriptive Statistics

The samples related to thirty executions of algorithms are analyzed by using descriptive statistics metrics, such as the mean, the media, the standard deviation (std), and the maximum (max) and minimum (min) values of the objective function. In addition, the number of Non-Feasible Solutions (NFS) at the end of the optimization process through the thirty executions is also included to know whether the constraint-handling technique shows some issue to leave the infeasible region.

##### Study case 1:

Table 5 shows the descriptive statistics of samples for study case 1 using trajectory 1 in the synthesis process. The columns are related to the analyzed algorithm, the descriptive statistics metrics, and the NFS for each sample. Boldface indicates the minimum value for each column. According to the results in Table 5, all metaheuristic algorithms find feasible solutions through the thirty executions (see NFS metric). R1B-NCH, R1B-SR, and R1B-PF obtain the three best solutions in that order (see column "min"). R1B-NCH gives the best algorithm performance finding solutions. The obtained solution in R1B-SR and R1B-PF presents an increment of 37.88% and 41.94%, respectively, with respect to R1B-NCH (see column "min"). In addition, the mean and minimum results are less in R1B-NCH than other algorithms, and the standard deviation, the median, and maximum values present an average behavior.

Table 6 shows the descriptive statistics of the samples for study case 1 using trajectory 2 in the synthesis process. The columns of such tables contain the same terms explained above. In this analysis, the R1B-NCH algorithm reliability is studied under trajectory changes to be followed by the four-bar linkage mechanism. According to the results in Table 6, all metaheuristic algorithms find feasible solutions (see "NFS" column). MUMSA- $\epsilon$ C, R1B-FR, and R1B-NCH obtain the three best solutions in that order. In spite of R1B-NCH presenting the third position, the difference with respect to the MUMSA- $\epsilon$ C is minimal at around 2.59% (see "min" column). Moreover, the R1B-NCH algorithm has the best mean value, indicating the consistency of results through different executions. The standard deviation, the median, the maximum, and minimum values of the objective function have average performance in the R1B-NCH with respect to other algorithms.

**Table 5.** Descriptive statistics of the obtained results for each metaheuristic algorithm in study case 1 with trajectory 1.

Algorithm-CHT	Mean	Std.	Median	Min	Max	NFS
R1B-FR	0.02619	0.01033	0.03075	0.003966	0.04198	0
GA-FR	0.03709	0.006465	0.03808	0.02067	0.05163	0
PSO-FR	0.03626	0.01057	0.03566	0.02025	0.05834	0
MUMSA-FR	0.03233	0.007363	0.03338	0.01265	0.0421	0
R1B-SR	0.03088	0.008759	0.03394	0.002988	0.04118	0
GA-SR	0.03578	0.01002	0.03773	0.01142	0.05438	0
PSO-SR	0.02846	0.007408	<b>0.02765</b>	0.004439	0.04094	0
MUMSA-SR	0.03104	0.00883	0.0341	0.01653	0.04237	0
R1B-εC	0.03395	<b>0.004956</b>	0.03378	0.02432	0.04756	0
GA-εC	0.04107	0.009803	0.04155	0.01859	0.0544	0
PSO-εC	0.03441	0.007179	0.03341	0.02315	0.0567	0
MUMSA-εC	0.02708	0.01299	0.0295	0.003726	0.04752	0
R1B-PF	0.02486	0.01167	0.03135	0.003076	<b>0.03391</b>	0
GA-PF	0.0397	0.008355	0.038	0.02922	0.05966	0
PSO-PF	0.03371	0.01283	0.03478	0.005304	0.06198	0
MUMSA-PF	0.02407	0.009266	0.02789	0.009579	0.03866	0
R1B-NCH	<b>0.02255</b>	0.01324	0.02978	<b>0.002167</b>	0.03585	0

**Table 6.** Descriptive statistics of the obtained results for each metaheuristic algorithm in study case 1 with trajectory 2.

Algorithm-CHT	Mean	Std.	Median	Min	Max	NFS
R1B-FR	0.02357	0.01494	0.03304	0.002539	0.04297	0
GA-FR	0.03521	0.01041	0.03816	0.01467	0.05109	0
PSO-FR	0.03724	<b>0.00713</b>	0.03643	0.02442	0.05452	0
MUMSA-FR	0.03205	0.009535	0.03412	0.01266	0.048	0
R1B-SR	0.02685	0.01408	0.03331	0.002685	0.04368	0
GA-SR	0.04522	0.01005	0.04495	0.02564	0.06512	0
PSO-SR	0.0353	0.01015	0.03663	0.003119	0.05145	0
MUMSA-SR	0.03064	0.01027	0.03104	0.01039	0.05104	0
R1B-εC	0.02956	0.009226	0.03264	0.01179	0.04592	0
GA-εC	0.03973	0.009862	0.03818	0.02489	0.05877	0
PSO-εC	0.03608	0.009744	0.03674	0.01091	0.05446	0
MUMSA-εC	0.02675	0.01264	0.03091	<b>0.002507</b>	0.05102	0
R1B-PF	0.02509	0.01224	<b>0.02461</b>	0.002746	0.04512	0
GA-PF	0.03458	0.01353	0.04058	0.005144	0.05076	0
PSO-PF	0.03452	0.01104	0.03651	0.01113	0.05355	0
MUMSA-PF	0.02689	0.007722	0.02891	0.00876	<b>0.03767</b>	0
R1B-NCH	<b>0.02336</b>	0.01577	0.03441	0.002572	0.04166	0

Study case 2:

Table 7 shows the descriptive statistics for study case 2 using trajectory 1 in the synthesis process. The columns of such tables contain the same terms explained above. According to the results in Table 7, all metaheuristic algorithms find feasible solutions except for the PSO-FR and R1B-SR algorithm (see “NFS” column). R1B-NCH, MUMSA-FR, and MUMSA-εC obtain the three best solutions (see “min” column). R1B-NCH gives the best algorithm performance finding solutions due to the mean, the median, and the min value of the best objective function being less than other algorithms. The standard deviation and the maximum value of the objective function also present a lower value than the average in the proposal.



**Table 7.** Descriptive statistics of the obtained results for each metaheuristic algorithm in study case 2 with the trajectory 1.

Algorithm-CHT	Mean	Std.	Median	Min	Max	NFS
R1B-FR	0.7046	0.05785	0.6858	0.6778	0.9428	0
GA-FR	19.72	34.7	7.387	2.206	145.5	0
PSO-FR	5.092	2.945	4.244	2.232	13.56	1
MUMSA-FR	1.641	0.8234	1.158	0.6581	3.145	0
R1B-SR	0.7533	—	0.7533	0.7533	<b>0.7533</b>	29
GA-SR	3.123	1.743	2.455	1.892	8.379	0
PSO-SR	3.407	4.932	2.441	1.334	28.89	0
MUMSA-SR	1.779	1.324	1.206	0.6633	6.056	0
R1B-εC	0.6886	<b>0.04365</b>	0.6751	0.6625	0.8178	0
GA-εC	5.154	2.55	3.98	2.339	11.08	0
PSO-εC	2.336	1.067	2.17	0.983	6.829	0
MUMSA-εC	1.797	1.694	0.8433	0.6595	5.484	0
R1B-PF	0.7961	0.3161	0.7118	0.6722	2.422	0
GA-PF	2.714	0.4742	2.506	2.077	3.639	0
PSO-PF	2.91	1.138	2.899	1.388	7.702	0
MUMSA-PF	2.901	2.317	2.341	0.7097	10.47	0
R1B-NCH	<b>0.6823</b>	0.05252	<b>0.6632</b>	<b>0.658</b>	0.8157	0

Table 8 shows the descriptive statistics for study case 2 using trajectory 2 in the synthesis process. The columns of such tables contain the same terms explained above. In this analysis, the R1B-NCH algorithm reliability is studied under a trajectory change in the cam-linkage mechanism. According to the results in Table 8, all metaheuristic algorithms find feasible solutions except for PSO-FR, R1B-SR, and PSO-εC (see “NFS” column). MUMSA-FR, R1B-NCH, and MUMSA-SR obtain the three best solutions in that order. The R1B-NCH presents a very slight increment of around 0.68% with respect to the obtained result given in MUMSA-FR (see “min” column). The R1B-NCH also obtains the best values in the mean, the standard deviation, the median, and the maximum value of the objective function.

**Table 8.** Descriptive statistics of the obtained results for each metaheuristic algorithm in study case 2 with trajectory 2.

Algorithm-CHT	Mean	Std.	Median	Min	Max	NFS
R1B-FR	0.7011	0.08787	0.6694	0.6592	0.9423	0
GA-FR	6.38	6.634	3.33	2.098	35.07	0
PSO-FR	3.705	2.125	3.331	1.706	13.79	2
MUMSA-FR	1.601	1.468	0.8563	<b>0.6418</b>	5.802	0
R1B-SR	—	—	—	—	—	30
GA-SR	3.77	2.143	2.737	1.889	7.992	0
PSO-SR	5.722	11.51	2.966	1.03	62.69	0
MUMSA-SR	1.681	1.233	1.042	0.649	4.779	0
R1B-εC	0.686	0.06462	0.6591	0.6556	0.8979	0
GA-εC	5.369	2.458	4.001	1.8	8.895	0
PSO-εC	2.596	1.027	2.506	1.394	7.299	1
MUMSA-εC	1.755	1.594	0.9007	0.6536	5.441	0
R1B-PF	0.7394	0.1097	0.691	0.6796	1.056	0
GA-PF	2.836	0.8672	2.561	2.065	6.465	0
PSO-PF	2.983	1.476	2.779	1.151	7.049	0
MUMSA-PF	3.183	5.02	2.204	0.6996	28.75	0
R1B-NCH	<b>0.6588</b>	<b>0.03913</b>	<b>0.6487</b>	0.6462	<b>0.8027</b>	0

It is observed in Tables 5–8 that the R1B-NCH presents an outstanding mean performance (see “mean” columns) in the search for solutions through study cases in spite of presenting changes in the optimization problems (rehabilitation trajectories) where the algorithm setting and the NCH technique were tuned and trained, respectively.

However, descriptive statistics cannot estimate the best algorithm behaviors for future executions. Therefore, the analysis of results by 95% confidence intervals and inferential statistics [56] is carried out in the next section.

#### 4.2.2. Confidence Intervals and Inferential Statistics

This section confirms the best algorithm behavior to solve each study case considered in this work. The best algorithm behavior is obtained by analyzing the sample (thirty best values of the objective function) obtained in each algorithm execution by 95% confidence intervals [61] and using the non-parametric Friedman test [56].

The 95% Confidence Interval (CI) [61] is a range of values for a selected sample of a study, where, with a 95% confidence, the interval contains the population’s true mean. Therefore, the confidence interval values closest to zero indicate the most promising algorithm due to the high probability that the mean behavior of the algorithm is inside such interval. Moreover, in what follows, for each study case, the use of inferential statistics through the Friedman test of the algorithm samples with the best CI will confirm the algorithm performance.

Study case 1:

Table 9 shows the confidence interval values obtained by each algorithm for study case 1 using both trajectories, and Figure 7 shows its graphical representation. It is observed that the R1B-NCH, the R1B-PF, and the MUMSA-PF have confidence intervals closer to zero (to the left) for trajectory 1. The R1B-NCH presents the confidence interval lower limit value closest to zero, and its confidence interval upper limit does not exceed the MUMSA-PF confidence interval upper limit. The results using trajectory 2 show that R1B-FR, R1B-PF, and R1B-NCH have confidence intervals closer to zero. The R1B-NCH presents the confidence interval lower limit value closest to zero, and the upper limit does not exceed the confidence interval upper limit of R1B-FR.

**Table 9.** Confidence interval values of algorithm for study case 1 in both trajectories.

Trajectory 1					
Limits	R1B-FR	R1B-SR	R1B-εC	R1B-PF	R1B-NCH
Low	0.0223	0.0276	0.0321	0.0205	0.0176
Up	0.0300	0.0341	0.0358	0.0292	0.0275
	GA-FR	GA-SR	GA-εC	GA-PF	
Low	0.0347	0.0320	0.0374	0.0366	
Up	0.0395	0.0395	0.0447	0.0428	
	PSO-FR	PSO-SR	PSO-εC	PSO-PF	
Low	0.0323	0.0257	0.0317	0.0289	
Up	0.0402	0.0312	0.0371	0.0385	
	MUMSA-FR	MUMSA-SR	MUMSA-εC	MUMSA-PF	
Low	0.0296	0.0277	0.0222	0.0206	
Up	0.0351	0.0343	0.0319	0.0275	

Table 9. Cont.

Trajectory 2					
Limits	R1B-FR	R1B-SR	R1B-εC	R1B-PF	R1B-NCH
Low	0.0180	0.0216	0.0261	0.0205	0.0175
Up	0.0291	0.0321	0.0330	0.0297	0.0292
Limits	GA-FR	GA-SR	GA-εC	GA-PF	
Low	0.0313	0.0415	0.0360	0.0295	
Up	0.0391	0.0490	0.0434	0.0396	
Limits	PSO-FR	PSO-SR	PSO-εC	PSO-PF	
Low	0.0346	0.0315	0.0324	0.0304	
Up	0.0399	0.0391	0.0397	0.0386	
Limits	MUMSA-FR	MUMSA-SR	MUMSA-εC	MUMSA-PF	
Low	0.0285	0.0268	0.0220	0.0240	
Up	0.0356	0.0345	0.0315	0.0298	

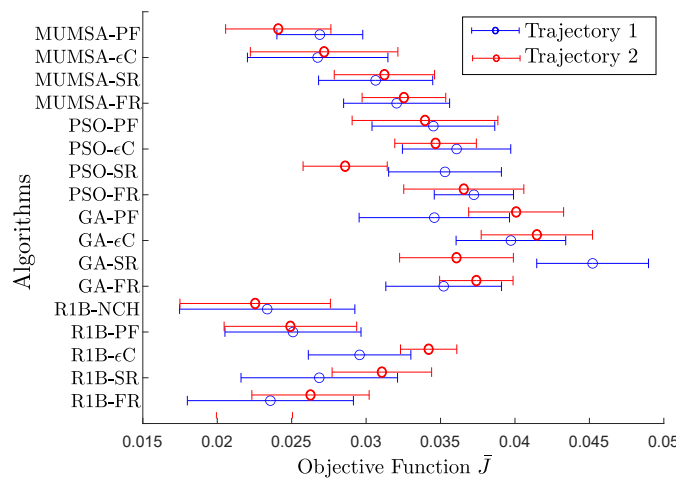


Figure 7. Graphical representation of confidence intervals for algorithms solving the study case 1 in both trajectories.

The 95%-confidence Friedman test is carried out in those three outstanding algorithms per each trajectory, i.e., in R1B-NCH, R1B-PF, and MUMSA-PF for trajectory 1; and, in R1B-FR, R1B-PF, and R1B-NCH for trajectory 2. According to the Friedman test in those groups of three algorithms, there is no statistically significant difference in the comparisons because the returned *p*-value is greater than 0.05. It is also confirmed in the multiple comparisons with the Bonferroni post-hoc error correction method of the Friedman test given in Table 10 that the associated *p*-value is greater than 0.05, indicating the no confirmation of the superiority of one algorithm in the comparison. In this paper, a tie is declared in the comparison when that situation occurs. Thus, in all comparisons presented in Table 10, the algorithms draw.

In the mechanism synthesis problem, it is common to obtain the optimal solution by setting different (usually thirty) executions to the same problem [47]. However, there may be significant differences among the best solutions obtained through different executions, and the most promising solution obtained by the optimizer in the mechanism synthesis problem is given through the best solution among executions [62].

**Table 10.** *p*-values for tests in the multiple comparisons among the three outstanding algorithms in study case 1 using both trajectories. All algorithms draw two times in both trajectories.

Trajectory 1	
Hypotesis	Bonferroni
MUMSA-PF vs. R1B-NCH	1
MUMSA-PF vs. R1B-PF	1
R1B-NCH vs. R1B-PF	1

Trajectory 2	
Hypotesis	Bonferroni
R1B-FR vs. R1B-NCH	1
R1B-FR vs. R1B-PF	0.59012
R1B-NCH vs. R1B-PF	0.90510

Then, to know whether the proposed R1B-NCH empirically presents a high probability of finding the best results in the kinematic synthesis problem for lower limb rehabilitation, the following is defined: As the confidence interval indicates whether these executions are repeated in several tests, the proportion of calculated 95% confidence intervals that is included the population’s true mean value would tend toward 95%; then, the confidence interval closest to zero will indicate the population’s true mean value is closer to the reduction of the objective function, and, hence, it is assumed that this interval presents a high probability of granting the best solution (minimum value) through thirty executions of the algorithm (executions commonly used for finding the best solution in the mechanism synthesis problem).

In order to verify the previous assumption, the other four sets of thirty executions are performed for the two groups of three algorithms (R1B-NCH, R1B-PF, and MUMSA-PF for trajectory 1; and R1B-FR, R1B-PF, and R1B-NCH for trajectory 2). In Table 11, the minimum values from the thirty executions per set are displayed. It is observed that, in each column related to the set of data for the corresponding trajectories, the minimum value, marked in boldface, is obtained by the proposed R1B-NCH.

**Table 11.** Minimum value of the objective function through different sets of thirty algorithms’ executions in study case 1 in both trajectories.

Trajectory 1				
Algorithm-CHT	Set 1	Set 2	Set 3	Set 4
R1B-NCH	<b>0.00219</b>	<b>0.00305</b>	<b>0.00216</b>	<b>0.00214</b>
R1B-PF	0.00234	0.00848	0.00401	0.00215
MUMSA-PF	0.01340	0.01142	0.01166	0.00976

Trajectory 2				
Algorithm-CHT	Set 1	Set 2	Set 3	Set 4
R1B-NCH	<b>0.00255</b>	<b>0.00257</b>	<b>0.00266</b>	<b>0.00263</b>
R1B-PF	0.00257	0.00307	0.00341	0.00354
R1B-FR	0.00270	0.00260	0.00271	0.00353

According to this analysis for study case 1, it is observed that the R1B-NCH algorithm presents a high probability to find the best solution through thirty executions of the algorithm in the mechanism synthesis problem for lower limb rehabilitation.

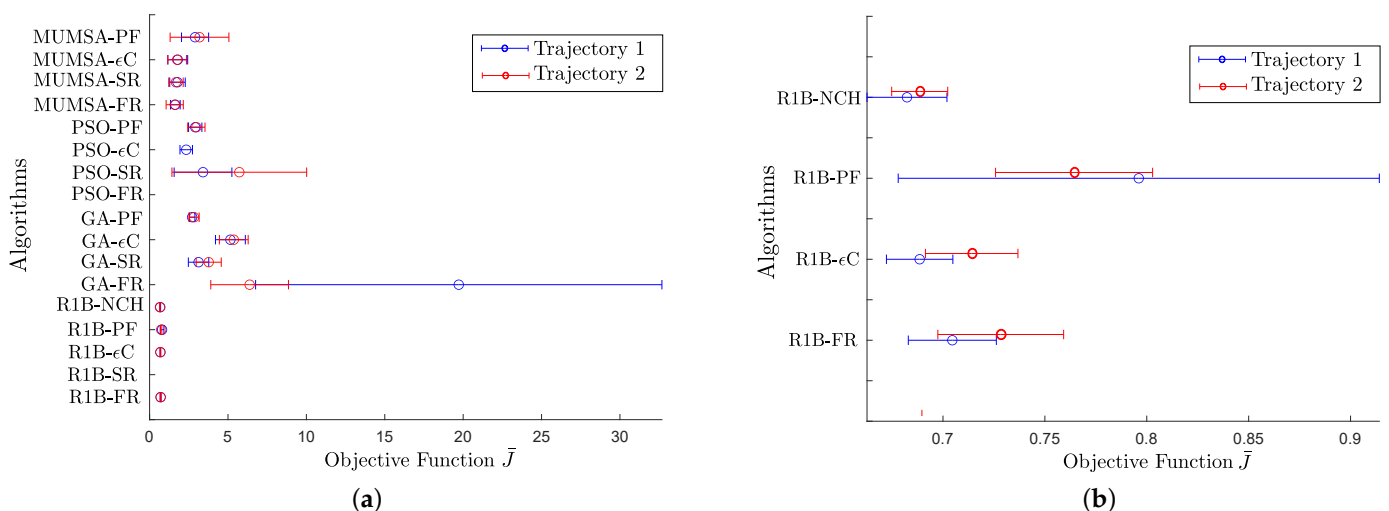
Study case 2:

Table 12 shows the confidence intervals obtained by each algorithm for study case 2 using both trajectories. Figure 8a shows the graphical representation of the confidence interval, and Figure 8b shows the close-up view of the confidence intervals closer to

zero. According to the results for both trajectories, R1B-FR, R1B- $\epsilon$ C, and R1B-NCH have confidence intervals closer to zero, where R1B-NCH presents the confidence interval lower limit value closest to zero and its upper limit does not exceed the R1B- $\epsilon$ C confidence interval upper limit.

**Table 12.** Confidence interval values of algorithms for the study case 2 in both trajectories.

Trajectory 1					
Limits	R1B-FR	R1B-SR	R1B- $\epsilon$ C	R1B-PF	R1B-NC
Low	0.6830	-	0.6723	0.6780	0.6627
Up	0.7262	-	0.7049	0.9141	0.7019
GA					
Limits	GA-FR	GA-SR	GA- $\epsilon$ C	GA-PF	
Low	6.7614	2.4716	4.2015	2.5364	
Up	32.6767	3.7737	6.1055	2.8906	
PSO					
Limits	PSO-FR	PSO-SR	PSO- $\epsilon$ C	PSO-PF	
Low	-	1.5653	1.9376	2.4854	
Up	-	5.2484	2.7343	3.3350	
MUMSA					
Limits	MUMSA-FR	MUMSA-SR	MUMSA- $\epsilon$ C	MUMSA-PF	
Low	1.3340	1.2845	1.1647	2.0357	
Up	1.9489	2.2735	2.4295	3.7658	
Trajectory 2					
Limits	R1B-FR	R1B-SR	R1B- $\epsilon$ C	R1B-PF	R1B-NC
Low	0.6683	-	0.6618	0.6984	0.6442
Up	0.7339	-	0.7101	0.7803	0.6735
GA					
Limits	GA-FR	GA-SR	GA- $\epsilon$ C	GA-PF	
Low	3.9029	2.9697	4.4512	2.5120	
Up	8.8576	4.5703	6.2872	3.1597	
PSO					
Limits	PSO-FR	PSO-SR	PSO- $\epsilon$ C	PSO-PF	
Low	-	1.4245	2.2053	2.4316	
Up	-	10.0193	2.9865	3.5342	
MUMSA					
Limits	MUMSA-FR	MUMSA-SR	MUMSA- $\epsilon$ C	MUMSA-PF	
Low	1.0532	1.2205	1.1600	1.3086	
Up	2.1492	2.1417	2.3506	5.0576	



**Figure 8.** Graphical representation of confidence intervals for algorithms solving the study case 2 in both trajectories. (a) Original size. (b) Close-up view.

The 95%-confidence Friedman test is performed in those three outstanding algorithms, i.e., in R1B-FR, R1B-εC, and R1B-NCH for both trajectories. According to the Friedman test in those groups of three algorithms, there exist a statistically significant difference among them in both trajectories because the *p*-value associated is less than the proposed significance level. Hence, to determine if one algorithm outperforms the other, inferential statistics given by the Friedman test for multiple comparisons with the Bonferroni post-hoc error correction method is carried out for finding the accurate pairwise comparisons. In Table 13, such tests are given, and in boldface, the winner in the comparison is highlighted. According to the number of wins, the results indicate that R1B-NCH is the most promising optimizer in the optimal synthesis for the study case 2 in both trajectories because it wins in all comparisons.

**Table 13.** *p*-values for tests in the multiple comparisons among the three outstanding algorithms in study case 2 using both trajectories. R1B-NCH wins two times in both trajectories. R1B-EC wins one time in trajectory 1.

Trajectory 1	
Hypotesis	Bonferroni
R1B-EC vs. R1B-FR	<b>9.0179</b> × 10 <sup>-4</sup>
R1B-EC vs. <b>R1B-NCH</b>	<b>4.2514</b> × 10 <sup>-2</sup>
R1B-FR vs. <b>R1B-NCH</b>	<b>3.8933</b> × 10 <sup>-9</sup>
Trajectory 2	
Hypotesis	Bonferroni
R1B-EC vs. R1B-FR	8.4557 × 10 <sup>-2</sup>
R1B-EC vs. <b>R1B-NCH</b>	<b>1.8685</b> × 10 <sup>-5</sup>
R1B-FR vs. <b>R1B-NCH</b>	<b>5.7132</b> × 10 <sup>-11</sup>

In order to empirically confirm the superiority of the proposed R1B-NCH, to obtain the optimal solution by setting different (usually thirty) executions to the same problem, other four sets of thirty executions are performed for the two groups of three algorithms (R1B-FR, R1B-εC, and R1B-NCH for both trajectories). In Table 14, the minimum values from the thirty executions per set are displayed. It is observed that, in each column, the minimum value, marked in boldface, is given by the expected algorithm (based on Table 13), i.e., the proposed R1B-NCH. Note that these boldface values cannot be reached by the other algorithms (R1B-FR and R1B-εC), confirming the statistical analysis presented above.

**Table 14.** Minimum value of the objective function through different sets of thirty algorithms' executions in study case 2 in both trajectories.

Trajectory 1				
Algorithm-CHT	Set 1	Set 2	Set 3	Set 4
R1B-NCH	<b>0.65626</b>	<b>0.65628</b>	<b>0.65606</b>	<b>0.65637</b>
R1B-FR	0.67261	0.67827	0.67514	0.67639
R1B-εC	0.67012	0.66972	0.66852	0.66230
Trajectory 2				
Algorithm-CHT	Set 1	Set 2	Set 3	Set 4
R1B-NCH	<b>0.64382</b>	<b>0.64224</b>	<b>0.64307</b>	<b>0.64441</b>
R1B-FR	0.64948	0.65921	0.65997	0.65742
R1B-εC	0.65648	0.65463	0.65531	0.64569

#### 4.2.3. Overall Evaluation of the Proposed NCH through Study Cases

The overall evaluation of the most promising algorithms given in Section 4.2.2 is discussed next.

In the first study case and according to Table 10, the most promising algorithms considering both trajectories are: R1B-NCH, R1B-PF, MUMSA-PF, and R1B-FR. In that case, all algorithms draw with each other. Thus, each algorithm presents two draws. In the second study case and according to Table 13, the most promising algorithms are: R1B-NCH, R1B-EC, and R1B-FR. In such a case, the R1B-NCH wins four times, and the R1B-EC wins one time. With this information, a summary of the number of wins and draws for both study cases, and trajectories are presented in Table 15. According to this table, the overall evaluation of the most promising algorithms is analyzed, and it is confirmed that the most promising one is the proposed R1B-NCH because it wins and draws four times. The next best behaviors of the algorithms to solve the synthesis problem are related to R1B-PF and R1B-FR in that order.

**Table 15.** Number of wins and draws in the overall performance of the most promising algorithms.

Algorithm-CHT	Win	Draw	Total
R1B-NCH	4	4	8
R1B-PF	0	4	4
R1B-FR	0	3	3
R1B-EC	1	1	2
MUMSA-PF	0	2	2

Then, the NCH technique increases the exploration and exploitation capabilities in the ED/RAND/1/BIN algorithm to solve the considered mechanism synthesis problems. The obtained solution (mechanism) could improve the lower limb rehabilitation machine.

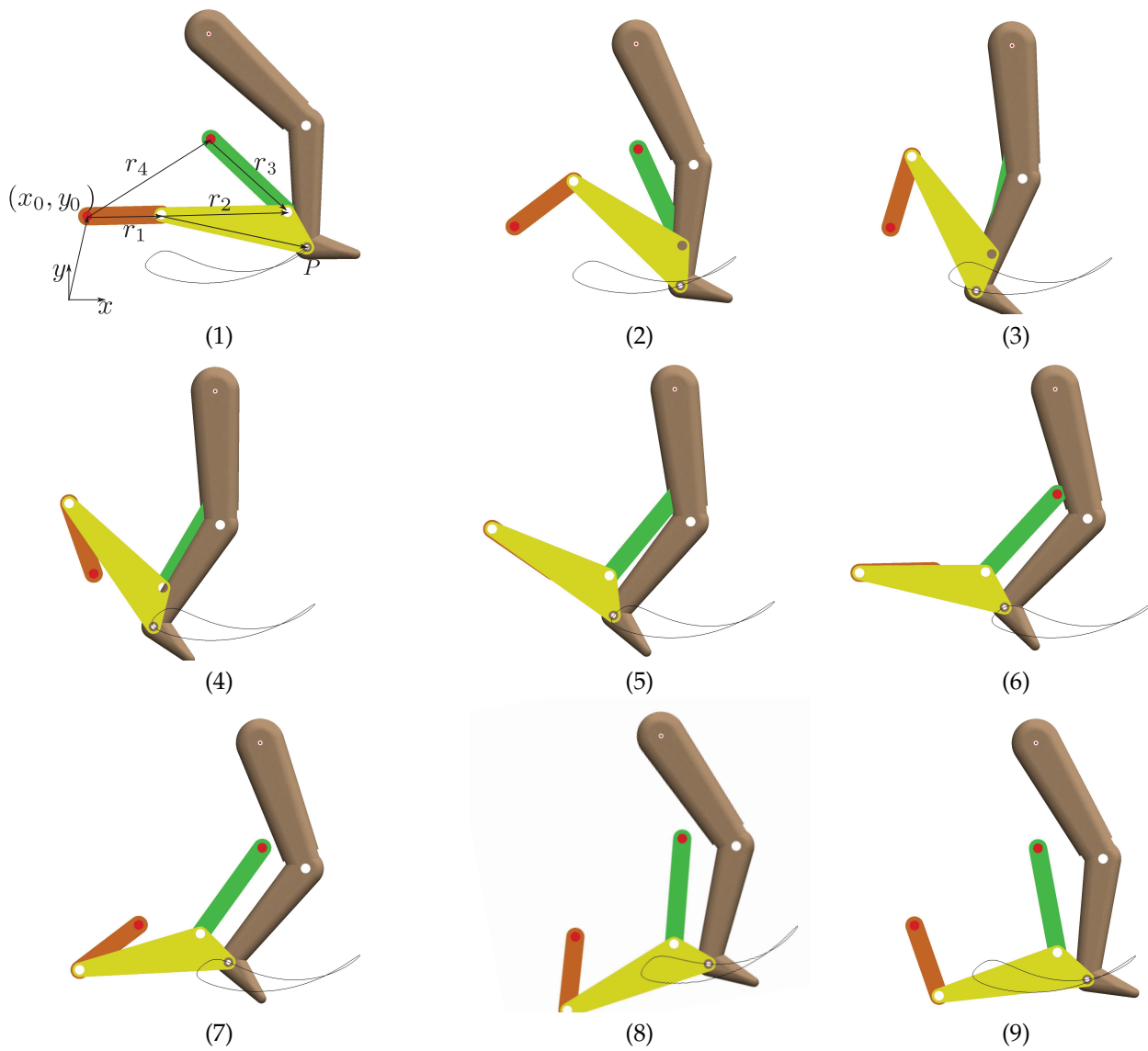
Finally, the best solutions obtained by the R1B-NCH algorithm for each study case are displayed in Tables 16 and 17. In order to validate the obtained mechanism for rehabilitation purposes, the Computer-Aided Design (CAD) of the mechanism in different phases of the crank movement is shown in Figures 9 and 10. Through the figure sequences given by the number 1 to the sequence number 9, it is observed that the coupler point of the mechanism in both figures can provide the proposed rehabilitation routine showing in a continuous line. Furthermore, if we fixed the human ankle joint in the coupler point of the mechanism, the mechanism can reproduce the natural movement of the leg, as is shown in those figures.

**Table 16.** Design variable vector obtained by R1B-NCH for study case 1.

Design variable	$x_1$ [m]	$x_2$ [m]	$x_3$ [m]	$x_4$ [m]	$x_5$ [rad]	$x_6$ [m]	$x_7$ [m]	$x_8$ [m]
Value	0.5993	0.3056	0.5188	0.4359	0.5584	−0.0820	0.3773	0.5959
Design variable	$x_9$ [m]	$x_{10}$ [rad]	$x_{11}$ [rad]	$x_{12}$ [rad]	$x_{13}$ [rad]	$x_{14}$ [rad]	$x_{15}$ [rad]	$x_{16}$ [rad]
Value	−0.1457	−0.8068	−1.1432	0.6441	0.7786	0.9475	1.1921	1.5975
Design variable	$x_{17}$ [rad]	$x_{18}$ [rad]	$x_{19}$ [rad]	$x_{20}$ [rad]	$x_{21}$ [rad]	$x_{22}$ [rad]	$x_{23}$ [rad]	
Value	2.0538	2.5685	3.0521	−2.6381	−1.9459	−1.1755	−0.8090	

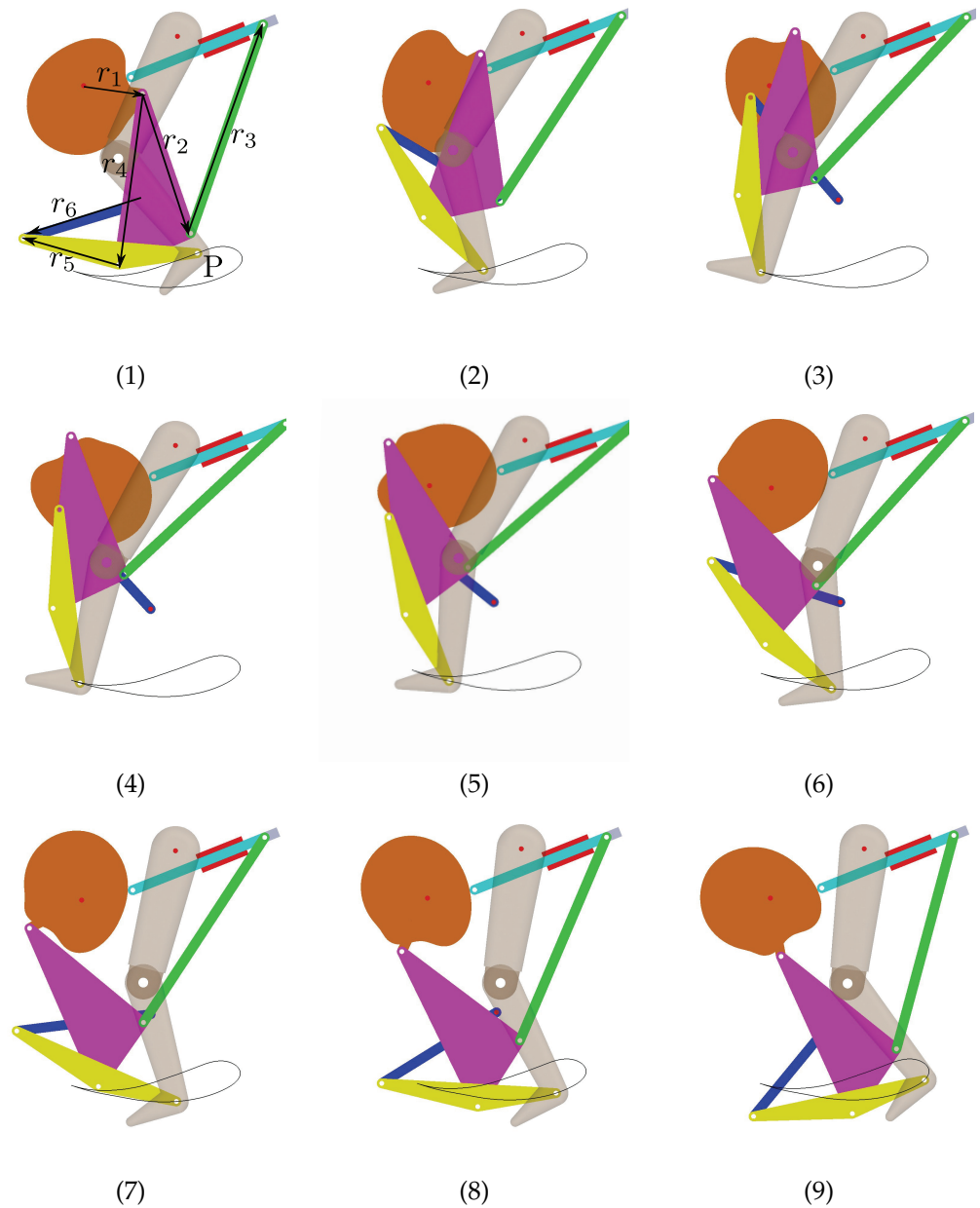
**Table 17.** Design variable vector obtained by R1B-NCH for study case 2.

Design variable	$x_1$ [mm]	$x_2$ [mm]	$x_3$ [mm]	$x_4$ [mm]	$x_5$ [mm]
Value	222.2690	592.4821	895.9455	686.9534	400.3920
Design variable	$x_6$ [mm]	$x_7$ [mm]	$x_8$ [mm]	$x_9$ [rad]	$x_{10}$ [rad]
Value	506.1880	509.3375	683.1163	0.4579	−1.0462
Design variable	$x_{11}$ [rad]	$x_{12}$ [rad]	$x_{13}$ [mm]	$x_{14}$ [mm]	$x_{15}$ [mm]
Value	−0.1572	−0.4755	−295.5667	14.9513	33.3667



**Figure 9.** CAD representation of the best mechanism movement obtained by R1B-NCH for study case 1.





**Figure 10.** CAD representation of the best mechanism movement obtained by R1B-NCH for study case 2.

## 5. Conclusions

In this work, the Neuronal Constraint-Handling (NCH) technique is proposed and included in the differential evolution variant ED/RAND/1/BIN algorithm. The proposal is validated in the solution of synthesis problems of the four-bar and cam-linkage mechanisms (two study cases) for developing lower-limb rehabilitation routines.

The constraint-handling techniques FR, SR,  $\epsilon$ C, and PF are included in the metaheuristic algorithms DE/RAND/1/BIN, GA, PSO, and MUMSA for comparative purposes.

The NCH performance is fairly and meaningfully compared with sixteen metaheuristic algorithms using the irace package for both the algorithm settings and neuronal network training, and the statistical analysis for the confirmation of the overall performance.

The statistical results confirm that the proposed R1B-NCH presents the best overall performance in spite of using different trajectories than were used to tune the algorithms and train the NCH technique. Furthermore, the results indicate that the NCH technique balances the exploration and exploitation capabilities in the ED/RAND/1/BIN algorithm

to solve the considered mechanism synthesis problems and thus obtains better design solutions.

Future work involves the use and the comparative study of other neuronal network architecture such as recurrent neural networks and symmetrically connected neural networks as constraint handling techniques. Recent authors' work indicates that the use of the relative angle method in the development of the kinematic motion in the synthesis problem of mechanism indirectly increases the exploration capability of the algorithms, i.e., the statement of the optimization problem highly influences in the optimizer. The use of NCH techniques with the relative angle method could benefit the search and improve the obtained results. Another research direction involves using the NCH technique in multi-objective search approaches for the mechanism synthesis problems in rehabilitation routines.

**Author Contributions:** Conceptualization, J.S.M.-R. and M.G.V.-C.; Data curation, J.S.M.-R.; Formal analysis, J.S.M.-R.; Funding acquisition, M.G.V.-C. and L.G.C.-R.; Investigation, J.S.M.-R. and M.G.V.-C.; Methodology, J.S.M.-R. and M.G.V.-C.; Project administration, M.G.V.-C.; Resources, L.G.C.-R.; Software, J.S.M.-R.; Supervision, M.G.V.-C.; Validation, J.S.M.-R., M.G.V.-C. and L.E.V.-S.; Visualization, J.S.M.-R. and L.E.V.-S.; Writing—original draft, J.S.M.-R. and M.G.V.-C.; Writing—review and editing, J.S.M.-R. and M.G.V.-C. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by Secretaría de Investigación y Posgrado del Instituto Politécnico Nacional (SIP-IPN) Grant Nos. 20180196, 20220255, 20181312, 20196226, 20201406, and 20221407.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The data presented in this study are available on request from the corresponding author.

**Acknowledgments:** The authors acknowledge support from the Secretaría de Investigación y Posgrado (SIP) and the Comisión de Operación y Fomento de Actividades Académicas (COFAA) of the Instituto Politécnico Nacional (IPN). The first and fourth authors acknowledge support from the Mexican Consejo Nacional de Ciencia y Tecnología (CONACyT) through a scholarship to pursue graduate studies at CIDETEC-UPIITA-IPN.

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

## Acronyms

NCH	Neuronal Constraint-Handling
DE	Differential Evolution
DE/RAND/1/BIN	DE variant with random mutation and binomial crossover
SQP	Sequential Quadratic Programming
GA	Genetic Algorithm
PSO	Particle Swarm Optimization
MUMSA	Malaga University Mechanism Synthesis Algorithm
POEMA	Pareto Optimum Evolutionary multi-objective Algorithm
GA-FL	GA-Fuzzy Logic
AG	Ant-Gradient
CS	Cuckoo Search
ICA	Imperialist Competitive Algorithm
TLBO	Teaching-Learning-Based Optimization
HLIDE	Hybrid Lagrange Interpolation DE
CMDE	Combined-Mutation DE
CHT	Constraint-Handling Techniques

**Nomenclature**

$\bar{J}$	Weighted objective function
$J_i$	$i$ -th objective function
$x$	Design variable vector
$g_j$	$j$ -th inequality constraint
$h_k$	$k$ -th equality constraint
$x_{min}$ & $x_{max}$	Upper and lower design variable vector bounds
$X^G$	Population of individuals in a $G$ generation
$X_p^G$	$p$ -th individual in the $X^G$ population
$U^G$	Offspring individuals in a $G$ generation
$\phi$	Constraint distance
$f(\mu)$	Sigmoid function
$a_r^s$	$r$ -th neuron in $s$ -th layer in NCH technique
$\hat{w}_{t,r}^s$	$t$ -th wight for $a_r^s$ neuron
$a_r^s$	$r$ -th bias in $s$ -th layer for $a_r^s$ neuron
$w_i$	$i$ -th weight in the objective function
$[\bar{x}_p^i, \bar{y}_p^i]$	Desired path points
$[x_p^i, y_p^i]$	Mechanism path points
$r_i$	$i$ -th length link
$\theta_2^i$	$i$ -th crank angle
$[x_0, y_0]$	Ground link origin
$[r_{c_x}, r_{c_y}]$	Lengths in the coupler link
$\beta, \gamma$ & $\eta$	Link angles
$e$	Slider displacement
$\alpha$	Slider angular position
$R_0$	Cam base radius
$\Theta_{ref}$	Normalization angle parameter
$R_{ref}$	Normalization radius parameter
$d_{ij}$	Distance between $i$ to $j$ point
CR	Crossover factor
$F_{min}$ & $F_{max}$	Maximum and minimum scale factor limit
MR	Mutation rate
Pf	Probabilistic factor in SR
cp	Control the relaxation of constraints
Tc	Maximum iterations to relax constraints
RG	Number of attempts to improve a solution
Pg	Probabilistic factor to improve a solution

**Appendix A. Trajectories for Rehabilitation in the Study Cases**

**Table A1.** Cartesian coordinates of the precision points for both trajectories in study case 1.

Trajectory 1		Trajectory 2	
$x$ [m]	$y$ [m]	$x$ [m]	$y$ [m]
0.7429	0.188	0.7429	0.188
0.6551	0.1573	0.6551	0.1573
0.6014	0.1388	0.6014	0.1388
0.5189	0.1149	0.5189	0.1249
0.4159	0.1012	0.4159	0.1212
0.3001	0.1074	0.3001	0.1174
0.1964	0.1375	0.1964	0.1375
0.1639	0.1662	0.1639	0.1662
0.1605	0.2003	0.1605	0.2003

**Table A1.** *Cont.*

Trajectory 1		Trajectory 2	
<i>x</i> [m]	<i>y</i> [m]	<i>x</i> [m]	<i>y</i> [m]
0.1934	0.2256	0.1934	0.2256
0.2619	0.2251	0.2619	0.2251
0.4201	0.1808	0.4201	0.2108
0.6474	0.1607	0.6474	0.1907
0.7429	0.188	0.7429	0.188

**Table A2.** Cartesian coordinates of the precision points for trajectory 1 in study case 2.

<i>x</i> [mm]	<i>y</i> [mm]	<i>x</i> [mm]	<i>y</i> [mm]	<i>x</i> [mm]	<i>y</i> [mm]
−315.8706	−725.914	102.286	−784.1517	249.9019	−624.0724
−305.426	−729.4993	113.9412	−782.7289	232.1635	−626.9287
−295.562	−732.633	126.2456	−780.9599	214.5033	−631.2082
−284.9638	−735.7146	137.8298	−779.0587	195.4565	−636.575
−272.9513	−738.8459	150.0497	−776.7977	175.5689	−643.126
−259.5651	−742.091	162.2326	−774.345	154.0295	−650.5493
−245.4816	−745.2649	175.691	−771.2814	132.9875	−658.2092
−230.7078	−748.3559	188.404	−768.0873	110.6898	−666.7843
−217.2474	−750.9006	201.7067	−764.4419	88.04307	−675.2332
−201.8657	−754.024	214.2527	−760.6806	62.68539	−684.4142
−188.5033	−756.6797	228.0001	−756.1684	37.99372	−693.0726
−173.8345	−759.6436	240.3332	−751.8006	13.98859	−701.2471
−161.1907	−762.2906	253.8222	−746.6116	−13.02161	−709.5296
−147.8624	−764.9876	265.2127	−741.7514	−38.76389	−716.7951
−134.5299	−767.7142	277.7069	−736.0049	−65.7052	−723.5015
−122.5271	−770.2517	289.3726	−730.1277	−93.18724	−729.2118
−111.1662	−772.6216	300.7921	−723.7715	−119.8744	−733.994
−98.40761	−774.9819	310.6746	−717.4391	−147.5638	−737.2401
−86.97769	−777.2058	320.2523	−710.5785	−172.2932	−739.7084
−74.81197	−779.1819	328.2496	−703.7132	−197.8191	−740.6221
−63.29198	−781.0032	335.1887	−696.3953	−222.7483	−740.2338
−51.73248	−782.6248	339.9644	−689.448	−244.9169	−738.7976
−40.13729	−784.046	343.6156	−681.9446	−266.9157	−735.9746
−28.51026	−785.2659	345.5832	−674.264	−284.0916	−733.2587
−16.16956	−786.2028	346.533	−666.2336	−299.6447	−729.774
−4.491148	−787.0124	344.1531	−659.0253	−313.0036	−726.1448
7.891441	−787.5247	340.7879	−651.3812	−321.6559	−723.6518
19.6055	−787.922	335.573	−644.3309	−328.838	−721.1686
30.63857	−788.0627	327.8983	−637.91	−332.1062	−720.0498
43.05355	−787.9534	318.5382	−632.043	−332.155	−720.179
54.77622	−787.6019	307.4652	−627.8585	−329.6154	−721.2705
67.19192	−787.0725	295.572	−624.2872	−323.2063	−723.8421
78.90697	−786.3287	281.0296	−622.9766	−316.7307	−726.2375
91.2885	−785.2526	266.4392	−622.3635		

**Table A3.** Cartesian coordinates of the precision points for trajectory 2 in study case 2.

$x$ [mm]	$y$ [mm]	$x$ [mm]	$y$ [mm]	$x$ [mm]	$y$ [mm]
−315.8706	−725.914	102.286	−777.1517	249.9019	−624.0724
−305.426	−729.4993	113.9412	−776.7289	232.1635	−626.9287
−295.562	−732.633	126.2456	−776.9599	214.5033	−631.2082
−284.9638	−735.7146	137.8298	−776.0587	195.4565	−636.575
−272.9513	−738.8459	150.0497	−774.7977	175.5689	−643.126
−259.5651	−742.091	162.2326	−773.345	154.0295	−650.5493
−245.4816	−745.2649	175.691	−771.2814	132.9875	−658.2092
−230.7078	−748.3559	188.404	−768.0873	110.6898	−666.7843
−217.2474	−750.9006	201.7067	−764.4419	88.04307	−675.2332
−201.8657	−754.024	214.2527	−760.6806	62.68539	−680.4142
−188.5033	−756.6797	228.0001	−756.1684	37.99372	−685.0726
−173.8345	−759.6436	240.3332	−751.8006	13.98859	−687.2471
−161.1907	−762.2906	253.8222	−746.6116	−13.02161	−692.5296
−147.8624	−764.9876	265.2127	−741.7514	−38.76389	−697.7951
−134.5299	−767.7142	277.7069	−736.0049	−65.7052	−701.5015
−122.5271	−770.2517	289.3726	−730.1277	−93.18724	−702.2118
−111.1662	−772.6216	300.7921	−723.7715	−119.8744	−701.994
−98.40761	−774.9819	310.6746	−717.4391	−147.5638	−703.2401
−86.97769	−777.2058	320.2523	−710.5785	−172.2932	−703.7084
−74.81197	−777.1819	328.2496	−703.7132	−197.8191	−704.6221
−63.29198	−777.0032	335.1887	−696.3953	−222.7483	−702.2338
−51.73248	−776.6248	339.9644	−689.448	−244.9169	−700.7976
−40.13729	−777.046	343.6156	−681.9446	−266.9157	−697.9746
−28.51026	−778.2659	345.5832	−674.264	−284.0916	−698.2587
−16.16956	−778.2028	346.533	−666.2336	−299.6447	−699.774
−4.491148	−778.0124	344.1531	−659.0253	−313.0036	−701.1448
7.891441	−778.5247	340.7879	−651.3812	−321.6559	−703.6518
19.6055	−777.922	335.573	−644.3309	−328.838	−705.1686
30.63857	−778.0627	327.8983	−637.91	−332.1062	−708.0498
43.05355	−777.9534	318.5382	−632.043	−332.155	−710.179
54.77622	−777.6019	307.4652	−627.8585	−329.6154	−713.2705
67.19192	−777.0725	295.572	−624.2872	−323.2063	−718.8421
78.90697	−777.3287	281.0296	−622.9766	−316.7307	−721.2375
91.2885	−777.2526	266.4392	−622.3635		

## References

1. Maciejasz, P.; Eschweiler, J.; Gerlach-Hahn, K.; Jansen-Troy, A.; Leonhardt, S. A survey on robotic devices for upper limb rehabilitation. *J. Neuroeng. Rehabil.* **2014**, *11*, 3. [\[CrossRef\]](#)
2. Díaz, I.; Gil, J.J.; Sánchez, E. Lower-limb robotic rehabilitation: Literature review and challenges. *J. Robot.* **2011**, *2011*, 759764. [\[CrossRef\]](#)
3. Muñoz-Reina, J.S.; Villarreal-Cervantes, M.G.; Corona-Ramírez, L.G. Integrated design of a lower limb rehabilitation mechanism using differential evolution. *Comput. Electr. Eng.* **2021**, *92*, 107103. [\[CrossRef\]](#)
4. Norton, R.L. *Design of Machinery: An Introduction to the Synthesis and Analysis of Mechanisms and Machines*; McGraw-Hill/Higher Education: New York, NY, USA, 2008.
5. Cabrera, J.; Simon, A.; Prado, M. Optimal synthesis of mechanisms with genetic algorithms. *Mech. Mach. Theory* **2002**, *37*, 1165–1177. [\[CrossRef\]](#)
6. Bulatović, R.R.; Đorđević, S.R.; Đorđević, V.S. Cuckoo Search algorithm: A metaheuristic approach to solving the problem of optimum synthesis of a six-bar double dwell linkage. *Mech. Mach. Theory* **2013**, *61*, 1–13. [\[CrossRef\]](#)
7. Calva-Yáñez, M.B.; Niño-Suarez, P.A.; Portilla-Flores, E.A.; Aponte-Rodríguez, J.A.; Santiago-Valentín, E. Reconfigurable mechanical system design for tracking an ankle trajectory using an evolutionary optimization algorithm. *IEEE Access* **2017**, *5*, 5480–5493. [\[CrossRef\]](#)
8. Bulatović, R.R.; Miodragović, G.; Bošković, M.S. Modified Krill Herd (MKH) algorithm and its application in dimensional synthesis of a four-bar linkage. *Mech. Mach. Theory* **2016**, *95*, 1–21. [\[CrossRef\]](#)
9. Rao, S.S. *Engineering Optimization-Theory and Practice*, 4th ed.; Wiley: Hoboken, NJ, USA, 2009.

10. Ávila Hernández, P.; Cuenca-Jiménez, F. Design and synthesis of a 2 DOF 9-bar spatial mechanism for a prosthetic thumb. *Mech. Mach. Theory* **2018**, *121*, 697–717. [[CrossRef](#)]
11. Ji, Z.; Manna, Y. Synthesis of a pattern generation mechanism for gait rehabilitation. *J. Med. Devices* **2008**, *2*, 1–8. [[CrossRef](#)]
12. Wang, H.; Wu, J.; Wang, Y.; Ren, L.; Zhang, D.; Lu, H. Research on the lower limb gait rehabilitation. In Proceedings of the 2014 IEEE International Conference on Mechatronics and Automation, Tianjin, China, 3–6 August 2014; pp. 1243–1247.
13. Tsuge, B.Y.; McCarthy, J.M. Synthesis of a 10-bar linkage to guide the gait cycle of the human leg. In *Proceedings of the ASME 2015 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*; American Society of Mechanical Engineers Digital Collection: New York, NY, USA, 2015; pp. 1–7.
14. Tsuge, B.Y.; Plecnik, M.M.; Michael McCarthy, J. Homotopy directed optimization to design a six-bar linkage for a lower limb with a natural ankle trajectory. *J. Mech. Robot.* **2016**, *8*, 061009. [[CrossRef](#)]
15. Conway, B.A. A survey of methods available for the numerical optimization of continuous dynamic systems. *J. Optim. Theory Appl.* **2012**, *152*, 271–306. [[CrossRef](#)]
16. Holland, J.H. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*; MIT Press: Cambridge, MA, USA, 1992.
17. Cabrera, J.; Ortiz, A.; Nadal, F.; Castillo, J. An evolutionary algorithm for path synthesis of mechanisms. *Mech. Mach. Theory* **2011**, *46*, 127–141. [[CrossRef](#)]
18. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the ICNN'95-International Conference on Neural Networks, Perth, WA, Australia, 27 November–1 December 1995; Volume 4, pp. 1942–1948.
19. Yang, X.S. Harmony Search as a Metaheuristic Algorithm. In *Music-Inspired Harmony Search Algorithm: Theory and Applications*; Geem, Z.W., Ed.; Springer: Berlin/Heidelberg, Germany, 2009; pp. 1–14.
20. Hamed, M.M.; Abdy, H.; Zahiri, S.H. IPO: An inclined planes system optimization algorithm. *Comput. Inform.* **2016**, *35*, 222–240.
21. Bataller, A.; Cabrera, J.; Clavijo, M.; Castillo, J. Evolutionary synthesis of mechanisms applied to the design of an exoskeleton for finger rehabilitation. *Mech. Mach. Theory* **2016**, *105*, 31–43. [[CrossRef](#)]
22. Shao, Y.; Xiang, Z.; Liu, H.; Li, L. Conceptual design and dimensional synthesis of cam-linkage mechanisms for gait rehabilitation. *Mech. Mach. Theory* **2016**, *104*, 31–42. [[CrossRef](#)]
23. Singh, R.; Chaudhary, H.; Singh, A.K. A novel gait-based synthesis procedure for the design of 4-bar exoskeleton with natural trajectories. *J. Orthop. Transl.* **2018**, *12*, 6–15. [[CrossRef](#)]
24. Coello, C.A.C. Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: A survey of the state of the art. *Comput. Methods Appl. Mech. Eng.* **2002**, *191*, 1245–1287. [[CrossRef](#)]
25. Cervantes-Culebro, H.; Cruz-Villar, C.A.; Peñalosa, M.G.M.; Mezura-Montes, E. Constraint-handling techniques for the concurrent design of a five-bar parallel robot. *IEEE Access* **2017**, *5*, 23010–23021. [[CrossRef](#)]
26. Miranda-Varela, M.E.; Mezura-Montes, E. Constraint-handling techniques in surrogate-assisted evolutionary optimization. An empirical study. *Appl. Soft Comput.* **2018**, *73*, 215–229. [[CrossRef](#)]
27. Mezura-Montes, E.; Coello Coello, C.A. Constraint-handling in nature-inspired numerical optimization: Past, present and future. *Swarm Evol. Comput.* **2011**, *1*, 173–194. [[CrossRef](#)]
28. He, X.S.; Fan, Q.W.; Karamanoglu, M.; Yang, X.S. Comparison of Constraint-Handling Techniques for Metaheuristic Optimization. In *Proceedings of the Computational Science-ICCS 2019*; Rodrigues, J.M.F., Cardoso, P.J.S., Monteiro, J., Lam, R., Krzhizhanovskaya, V.V., Lees, M.H., Dongarra, J.J., Sloat, P.M., Eds.; Springer International Publishing: Cham, Switzerland, 2019; pp. 357–366.
29. Muñoz-Reina, J.S.; Villarreal-Cervantes, M.G.; Corona-Ramírez, L.G. Empirical Study of Constraint-Handling Techniques in the Optimal Synthesis of Mechanisms for Rehabilitation. *Appl. Sci.* **2021**, *11*, 8739. [[CrossRef](#)]
30. Smith, A.E.; Coit, D.W.; Baeck, T.; Fogel, D.; Michalewicz, Z. Penalty functions. *Handb. Evol. Comput.* **1997**, *97*, 1–12.
31. Shiakolas, P.; Koladiya, D.; Kebrle, J. On the optimum synthesis of four-bar linkages using differential evolution and the geometric centroid of precision positions. *Inverse Probl. Eng.* **2002**, *10*, 485–502. [[CrossRef](#)]
32. Bulatović, R.R.; Dordević, S.R. On the optimum synthesis of a four-bar linkage using differential evolution and method of variable controlled deviations. *Mech. Mach. Theory* **2009**, *44*, 235–246. [[CrossRef](#)]
33. Shiakolas, P.; Koladiya, D.; Kebrle, J. On the optimum synthesis of six-bar linkages using differential evolution and the geometric centroid of precision positions technique. *Mech. Mach. Theory* **2005**, *40*, 319–335. [[CrossRef](#)]
34. Laribi, M.; Mlika, A.; Romdhane, L.; Zeghloul, S. A combined genetic algorithm-fuzzy logic method (GA-FL) in mechanisms synthesis. *Mech. Mach. Theory* **2004**, *39*, 717–735. [[CrossRef](#)]
35. Acharyya, S.; Mandal, M. Performance of EAs for four-bar linkage synthesis. *Mech. Mach. Theory* **2009**, *44*, 1784–1794. [[CrossRef](#)]
36. Smaili, A.; Diab, N. Optimum synthesis of hybrid-task mechanisms using ant-gradient search method. *Mech. Mach. Theory* **2007**, *42*, 115–130. [[CrossRef](#)]
37. Ebrahimi, S.; Payvandy, P. Efficient constrained synthesis of path generating four-bar mechanisms based on the heuristic optimization algorithms. *Mech. Mach. Theory* **2015**, *85*, 189–204. [[CrossRef](#)]
38. Zhang, K.; Huang, Q.; Zhang, Y.; Song, J.; Shi, J. Hybrid Lagrange interpolation differential evolution algorithm for path synthesis. *Mech. Mach. Theory* **2019**, *134*, 512–540. [[CrossRef](#)]
39. 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](#)]

40. Deb, K. An efficient constraint handling method for genetic algorithms. *Comput. Methods Appl. Mech. Eng.* **2000**, *186*, 311–338. [[CrossRef](#)]
41. Cabrera, J.; Nadal, F.; Munoz, J.; Simon, A. multi-objective constrained optimal synthesis of planar mechanisms using a new evolutionary algorithm. *Mech. Mach. Theory* **2007**, *42*, 791–806. [[CrossRef](#)]
42. Runarsson, T.P.; Yao, X. Stochastic ranking for constrained evolutionary optimization. *IEEE Trans. Evol. Comput.* **2000**, *4*, 284–294. [[CrossRef](#)]
43. Takahama, T.; Sakai, S.; Iwane, N. Constrained optimization by the  $\epsilon$  constrained hybrid algorithm of particle swarm optimization and genetic algorithm. In *Proceedings of the Australasian Joint Conference on Artificial Intelligence*; Springer: Berlin/Heidelberg, Germany, 2005; pp. 389–400.
44. Huang, Q.; Yu, Y.; Zhang, K.; Li, S.; Lu, H.; Li, J.; Zhang, A.; Mei, T. Optimal synthesis of mechanisms using repellency evolutionary algorithm. *Knowl.-Based Syst.* **2022**, *239*, 107928. [[CrossRef](#)]
45. Lin, W.Y. A GA–DE hybrid evolutionary algorithm for path synthesis of four-bar linkage. *Mech. Mach. Theory* **2010**, *45*, 1096–1107. [[CrossRef](#)]
46. Singh, R.; Chaudhary, H.; Singh, A.K. Defect-free optimal synthesis of crank-rocker linkage using nature-inspired optimization algorithms. *Mech. Mach. Theory* **2017**, *116*, 105–122. [[CrossRef](#)]
47. Valencia-Segura, L.E.; Villarreal-Cervantes, M.G.; Corona-Ramírez, L.G.; Cuenca-Jiménez, F.; Castro-Medina, R. Optimum synthesis of four-bar mechanism by using relative angle method: A comparative performance study. *IEEE Access* **2021**, *9*, 132990–133010. [[CrossRef](#)]
48. Lin, W.Y.; Hsiao, K.M. Cuckoo search and teaching–learning–based optimization algorithms for optimum synthesis of path-generating four-bar mechanisms. *J. Chin. Inst. Eng.* **2017**, *40*, 66–74. [[CrossRef](#)]
49. Rafiq, M.; Bugmann, G.; Easterbrook, D. Neural network design for engineering applications. *Comput. Struct.* **2001**, *79*, 1541–1552. [[CrossRef](#)]
50. Kiranyaz, S.; Avci, O.; Abdeljaber, O.; Ince, T.; Gabbouj, M.; Inman, D.J. 1D convolutional neural networks and applications: A survey. *Mech. Syst. Signal Process.* **2021**, *151*, 107398. [[CrossRef](#)]
51. Chua, L.O.; Yang, L. Cellular neural networks: Applications. *IEEE Trans. Circuits Syst.* **1988**, *35*, 1273–1290. [[CrossRef](#)]
52. Widrow, B.; Rumelhart, D.E.; Lehr, M.A. Neural networks: Applications in industry, business and science. *Commun. ACM* **1994**, *37*, 93–106. [[CrossRef](#)]
53. Kazikova, A.; Pluhacek, M.; Senkerik, R. Why tuning the control parameters of metaheuristic algorithms is so important for fair comparison? *Mendel* **2020**, *26*, 9–16. [[CrossRef](#)]
54. Huang, C.; Li, Y.; Yao, X. A Survey of Automatic Parameter Tuning Methods for Metaheuristics. *IEEE Trans. Evol. Comput.* **2020**, *24*, 201–216. [[CrossRef](#)]
55. López-Ibáñez, M.; Dubois-Lacoste, J.; Cáceres, L.P.; Birattari, M.; Stützle, T. The irace package: Iterated racing for automatic algorithm configuration. *Oper. Res. Perspect.* **2016**, *3*, 43–58. [[CrossRef](#)]
56. Derrac, J.; García, S.; Molina, D.; Herrera, F. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm Evol. Comput.* **2011**, *1*, 3–18. [[CrossRef](#)]
57. Marler, R.T.; Arora, J.S. The weighted sum method for multi-objective optimization: New insights. *Struct. Multidiscip. Optim.* **2010**, *41*, 853–862. [[CrossRef](#)]
58. 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](#)]
59. Sanger, T.D. Optimal unsupervised learning in a single-layer linear feedforward neural network. *Neural Netw.* **1989**, *2*, 459–473. [[CrossRef](#)]
60. de Souza, M.; Ritt, M.; López-Ibáñez, M. Capping methods for the automatic configuration of optimization algorithms. *Comput. Oper. Res.* **2022**, *139*, 105615. [[CrossRef](#)]
61. Tan, S.H.; Tan, S.B. The Correct Interpretation of Confidence Intervals. *Proc. Singap. Healthc.* **2010**, *19*, 276–278. [[CrossRef](#)]
62. 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](#)]