International Journal of Systems Science
Vol. 00, No. 00, 00 Month 20xx, 1–39

RESEARCH ARTICLE

Optimum Synthesis of a Four-Bar Mechanism Using the Modified Bacterial Foraging Algorithm

Efrén Mezura-Montes[a]*, Edgar A. Portilla-Flores[b] and Betania Hernández-Ocaña[a]

[a] *Laboratorio Nacional de Informática Avanzada (LANIA) A.C.*
Rébsamen 80, Centro, Xalapa, Veracruz, 91000, MEXICO
[b] *Centro de Innovación y Desarrollo Tecnológico en Cómputo (CIDETEC-IPN)*
U. Adolfo López Mateos, México D.F., 07700, MEXICO

(*Received 00 Month 20xx; final version received 00 Month 20xx*)

This paper presents the mechanical synthesis of a four-bar mechanism, its definition as a constrained optimization problem in presence of one dynamic constraint and its solution with a swarm intelligence algorithm based on the bacteria foraging process. The algorithm is adapted to solve the optimization problem by adding a suitable constraint-handling technique able to incorporate a selection criterion for the two objectives stated by the kinematic analysis of the problem. Moreover, a diversity mechanism, coupled with the attractor operator used by bacteria, is designed to favor the exploration of the search space. Four experiments are designed to validate the proposed model and to test the performance of the algorithm regarding constraint-satisfaction, sub-optimal solutions obtained, performance metrics, and an analysis of the solutions based on the simulation of the four-bar mechanism. The results are compared with those provided by four algorithms found in the specialized literature used to solve mechanical design problems. Based on the simulation analysis, the solutions obtained by the proposed algorithm lead to a more suitable design based on motion generation and operation quality.

Keywords: Optimum synthesis; four-bar mechanism; nature-inspired optimization; swarm intelligence; evolutionary algorithms.

1. Introduction

Nowadays, in the engineering design framework, one of the most important approaches is the usage of computer aided design (CAD) software. This approach allows the design engineer to obtain virtual 3D models, 2D drawings and numerical control code, among other features. However, the design of modern systems could be complex. One design approach commonly found in the specialized literature is the mechatronic approach, where the design of modern systems can be considered as a mechatronic design. This approach requires that a mechanical system and its control system are designed as an integrated one; then a concurrent design problem is proposed in order to obtain the whole system (Portilla-Flores et al. 2007). An alternative design approach considers assigning the best possible combination of values to a set of parameters that describe the system. In this approach, a kinematic analysis is carried out in a first stage. The kinematic analysis is used to fulfill positions and velocities of the mechanical system and it also allows the system to be described by a set of parameters. Moreover, the design engineer must propose a set of performance functions and constraints to quantify the system behavior. Thereafter, the design

*Corresponding author. Email: emezura@lania.mx Tel. +52 228 841 6100 Fax. +52 228 8416101

ISSN: 0020-7721 print/ISSN 1464-5319 online

DOI: 10.1080/00207721.20xx.CATSid
http://www.informaworld.com

engineer is able to propose several potential solutions (Norton 1997). Precisely in this part of the design approach, the kinematic analysis can be transformed into an optimization problem and then being solved by an optimization method (Nariman-Zadeh et al. 2009, Radovan and Stevan 2009, Saridakis and Dentsoras 2009).

Mathematical programming, such as the Newton method, has been adopted as in the work by Mermetas (2004) where an optimal kinematic design of a planar manipulator with four-bar mechanism was optimized. There, optimum link measurements of the manipulator that maximize the local mobility index depending on the input link location were found. Also, design charts for the optimum manipulator design were obtained.

Nonetheless, based on the inherent difficulty of the optimization problems obtained from the real-world system behavior of mechanical systems, it may be difficult to solve them with traditional mathematical programming methods. Therefore, nature-inspired meta-heuristic algorithms have become a valid option. They can be divided in two main classes (Engelbrecht 2007): (1) evolutionary algorithms (EAs) and (2) swarm intelligence algorithms (SIAs). The first ones emulate the evolution of species and the survival of the fittest. The second ones emulate the social behavior of some simple species searching for food or shelter. EAs comprise different algorithms such as genetic algorithms (GAs) (Eiben and Smith 2003), evolution strategies (ES) (Schwefel 1995), evolutionary programming (Fogel 1999), genetic programming (GP) (Koza et al. 2003) and differential evolution (DE) (Price et al. 2005). SIAs main approaches are the particle swarm optimization (PSO) (Kennedy and Eberhart 2001) and the ant colony optimization (ACO) (Dorigo et al. 1996). PSO is based on the choreographies followed by birds and is able to solve difficult numerical optimization problems (Daneshyari and Yen 2012, Wang et al. 2012). ACO models different behaviors found in ant nests but has been mainly used to solve combinatorial optimization problems. Among those novel SIAs there is one which emulates the foraging behavior of bacteria. It was proposed by Passino (2002) and is called bacterial foraging optimization algorithm (BFOA). The entire foraging process of the Escherichia Coli bacterium is emulated: Chemotaxis (tumble-swim movements), reproduction, and elimination-dispersal. More recently, the modified bacterial foraging optimization algorithm (MBFOA) was proposed by Mezura-Montes and Hernández-Ocaña (2009) with the aim to increase its applicability in engineering design by reducing the number of user-defined parameters and handling constraints as well.

The following are approaches based on EAs for optimizing mechanical systems. In (Mundo and Yan 2007) a method for the kinematic optimization of transmission mechanism was proposed. A motion control of a ball-screw transmission mechanism was developed. The kinematic characteristics of the ball-screw mechanism were analyzed by means of non-dimensional motion equations in order to formulate an optimization problem. A GA was implemented in order to optimize the objective function and a penalty method was used to fulfill the design rules. In (Nariman-Zadeh et al. 2009) a hybrid multi-objective genetic algorithms (GA) was used for Pareto optimum synthesis of four-bar linkages considering the minimization of two objective functions simultaneously. The obtained Pareto fronts demonstrated that trade-offs between these two objectives can be recognized so that a designer can select a desired four-bar linkage from the set of sub-optimal solutions. In (Radovan and Stevan 2009) the synthesis of a four-bar linkage in which the coupler point performs approximately a rectilinear motion was presented. Very high accuracy for motion along a straight line at a large number of given points was achieved by using the variable controlled deviations method and by also employing DE.

From the previously commented works, it was nothed that: (1) different search algorithms, e.g., GAs and DE, have been adapted to obtain the optimal design synthesis of four-bar mechanisms and (2) it was also noted that each mechanical system provides new and specific design challenges. These two findings are the main motivation of this paper.

The objective of the present work is two-fold: (1) to optimize the output motion and energy transmission of a continuously variable transmission (CVT) by designing a four-bar mechanism on the crank-rocker configuration (both explained later in the paper), and (2) to extend the solution of this type of optimal mechanical design problems to other meta-heuristic (MBFOA) not yet applied, to the best of the authors' knowledge.

As it will be detailed later in the paper, the optimization problem obtained from the mechanical system is a constrained bi-objective optimization problem and one of the constraints is dynamic. A set of experiments are carried out to analyze the behavior of the algorithm regarding the feasibility and the treatment of the dynamic constraint, the overall performance of the algorithm based on metrics and, finally, an analysis of the solutions based on simulations. Three algorithms found in the specialized literature are used for comparison purposes and statistical tests are applied to get confidence on the results presented.

The paper is organized as follows: Section 2 outlines the mechanism synthesis and the design problem of the four-bar mechanism. A kinematic analysis of the four-bar mechanism is also shown. Section 3 presents the development of the optimal strategy, i.e., the design of the objective funtions and the constraints. Thereafter, the bi-objective optimization problem and the design variables are detailed in Section 4. Section 5 then presents the proposed algorithm based on MBFOA by detailing those added mechanisms so as to get it suitable to solve the bi-objective optimization problem in presence of a dynamic constraint. After that, the experimental design, the obtained results and their corresponding discussions are included in Section 6. Finally, some conclusions and the future work are established in Section 7.

2. Mechanism Synthesis problem

There are several mechanisms using pedaling motion as a supply source for machinery. However, these mechanisms have some inherent difficulties due to the complexity of building a pinion-chain-crown mechanism. In (Pardo and Sánchez 2006) a new CVT configuration was proposed. It was based on a crank-slider mechanism to use pedaling motion. The operational principle of this mechanism is explained as follows: the mechanism converts the input pedaling motion onto the driving wheel rotating input by unilateral transmission. In order to do this, the mechanism uses ratchets that are alternatively engaged inside the links of the output chain to push and pull them. On the other hand, a special mechanism, with the aim to change the effective radius of pulling in the whole mechanical system, is considered. The above mentioned continuously variable transmission is proposed with the goal to introduce a gradual, linear change of velocity on pedaling vehicles. A simplified drawn from this mechanism can be seen in Figure 1

.

FIGURE 1 AROUND HERE.

As it can be observed, in the CVT configuration above mentioned, the special mechanism used for converting the input pedaling motion is very difficult to build. That is due to the particular manufacturing of the shaft that couples the input pedaling motion into the rest of the CVT. Therefore, in this work a four-bar mechanism is used to replace such mechanism. It is important to remark that the selection of such mechanism is based on the required kinematic by the whole system and the fact that the four-bar mechanisms have been widely studied and they are easy to build. A drawing from the CVT configuration proposed in this work can be observed in Figure 2

.

FIGURE 2 AROUND HERE.

An important fact is that the energy transmission from the input to the mechanical system in Figure 2 is related with the rocker motion amplitude of the four-bar mechanism. In other words, the input motion of the crank-slider mechanism is the rocker motion of the first mechanism. On the other hand, it is known that the velocity of the slider is a mathematical relationship of the *sin* of the crank angle (Shigley and Uicker 1995), i.e., the larger the value of this angle, the larger the velocity on the slider of the second mechanism. As a matter of fact, once the range of motion is satisfied, it is necessary to ensure a smooth transmission of force and speed on the joint of the connecting rod and the rocker of the four-bar mechanism. Therefore, getting mechanical elements which allow a large amplitude on the motion of the rocker is a goal of the mechanical design.

When the design engineer carries out the design of mechanisms, both, analysis and synthesis are taking into account. Regarding dimensional synthesis, there are tree problems to be considered: function generation, path generation and motion generation. In a function generation problem, an input motion is correlated with an output motion on a mechanism. The path generation problem is related to the control of a point within a plane, so that it can follows a path or trajectory previously established. Finally, a motion generation problem is defined by the control of a line within a plane, so that a set of sequential positions is fulfilled. This last problem is more general with respect to the path generation problem (Norton 1997).

In this work, the design of the four-bar mechanism precisely belongs to the motion generation problem. Therefore, two approaches are used to carry out the design of the four-bar mechanism: a dimensional synthesis for motion generation and also the performance of the mechanism.

2.1. *Kinematic analysis of the four-bar mechanism*

As it has been previously mentioned, the CVT is based on two mechanisms: one four-bar mechanism and one crank-slider mechanism. A kinematic analysis of the first mechanical system is carried out in order to obtain the behavior of the whole system. A schematic drawing of the four-bar mechanism is shown in Figure 3.

This four-bar mechanism is composed by a reference bar (r_1), a crank bar (r_2), a connecting rod bar (r_3) and a rocker bar (r_4). A set of four important angles has to be considered, beginning with θ_1, described by the angle between the horizontal axis and r_1, θ_2 is the angle between the horizontal axis and r_2, θ_3 is the angle between the horizontal axis and r_3 and θ_4 is the angle between the horizontal axis and r_4. The angle marked as μ is the transmission angle. This special configuration is the so-called crank-rocker.

FIGURE 3 AROUND HERE.

2.1.1. Analysis of position

For the mechanism in Figure 3, a loop-closure equation (Shigley and Uicker 1995) is proposed as follows:

$$\vec{r_1} + \vec{r_4} = \vec{r_2} + \vec{r_3} \tag{1}$$

where each one of the vector is related with each one of the linkages.

On the other hand, if vectors are written in polar form, (1) can be expressed as follows:

$$r_1 e^{j\theta_1} + r_4 e^{j\theta_4} = r_2 e^{j\theta_2} + r_3 e^{j\theta_3} \tag{2}$$

Appliying Newton's formula into (2), it can be writen as:

$$r_1(cos\theta_1 + jsin\theta_1) + r_4(cos\theta_4 + jsin\theta_4) = r_2(cos\theta_2 + jsin\theta_2) + r_3(cos\theta_3 + jsin\theta_3) \tag{3}$$

Separating the real and imaginary part of (3):

$$\begin{aligned} r_1 cos\theta_1 + r_4 cos\theta_4 &= r_2 cos\theta_2 + r_3 cos\theta_3 \\ jr_1 sin\theta_1 + jr_4 sin\theta_4 &= jr_2 sin\theta_2 + jr_3 sin\theta_3 \end{aligned} \tag{4}$$

In order to obtain the angular position θ_3, the left side of the equations system (4) is put in terms of θ_4:

$$\begin{aligned} r_4 cos\theta_4 &= r_2 cos\theta_2 + r_3 cos\theta_3 - r_1 cos\theta_1 \\ r_4 sin\theta_4 &= r_2 sin\theta_2 + r_3 sin\theta_3 - r_1 sin\theta_1 \end{aligned} \tag{5}$$

Taking the square of (5) and adding them, the Freudenstein's equation (Shigley and Uicker 1995) in a compact form is established as follows:

$$A_1 cos\theta_3 + B_1 sin\theta_3 + C_1 = 0 \tag{6}$$

where:

$$A_1 = 2r_3 \left(r_2 cos\theta_2 - r_1 cos\theta_1\right) \tag{7}$$

$$B_1 = 2r_3 \left(r_2 sin\theta_2 - r_1 sin\theta_1\right) \tag{8}$$

$$C_1 = r_1^2 + r_2^2 + r_3^2 - r_4^2 - 2r_1 r_2 cos\left(\theta_1 - \theta_2\right) \tag{9}$$

The angle θ_3 can be explicitly found as a function of the parameters A_1, B_1, C_1 and θ_2. Such solution is obtained by expressing $sin\theta_3$ and $cos\theta_3$ in terms of $tan\left(\frac{\theta_3}{2}\right)$ as follows:

$$sin\theta_3 = \frac{2tan(\frac{\theta_3}{2})}{1+tan^2(\frac{\theta_3}{2})}\ ,\ cos\theta_3 = \frac{1-tan^2(\frac{\theta_3}{2})}{1+tan^2(\frac{\theta_3}{2})} \tag{10}$$

and substituting those in Equation (6), a second order linear equation is obtained:

$$\left[C_1 - A_1\right] tan^2\left(\frac{\theta_3}{2}\right) + \left[2B_1\right] tan\left(\frac{\theta_3}{2}\right) + A_1 + C_1 = 0 \tag{11}$$

Solving Equation (11), the angular position θ_3 is given by Equation (12).

$$\theta_3 = 2arctan\left[\frac{-B_1 \pm \sqrt{B_1^2 + A_1^2 - C_1^2}}{C_1 - A_1}\right] \tag{12}$$

A similar mathematical procedure to obtain θ_4 must be done. The Freudestein's equation from Equation (1) in compact form is given by Equation (13).

$$D_1 cos\theta_4 + E_1 sin\theta_4 + F_1 = 0 \tag{13}$$

where:

$$D_1 = 2r_4\left(r_1 cos\theta_1 - r_2 cos\theta_2\right) \tag{14}$$

$$E_1 = 2r_4\left(r_1 sin\theta_1 - r_2 sin\theta_2\right) \tag{15}$$

$$F_1 = r_1^2 + r_2^2 + r_4^2 - r_3^2 - 2r_1 r_2 cos\left(\theta_1 - \theta_2\right) \tag{16}$$

Therefore, the angular position θ_4 is given by (17)

$$\theta_4 = 2arctan\left[\frac{-E_1 \pm \sqrt{D_1^2 + E_1^2 - F_1^2}}{F_1 - D_1}\right] \tag{17}$$

Equations (12) and (17) must take the signs of radicals according with the four-bar mechanism used. Table 1 shows the corresponding signs. It is important to remark that in this work, the open configuration was used.

TABLE 1 AROUND HERE.

2.1.2. Analysis of velocity

In order to carried out the analysis of velocity, equation (1) is time derivative. It is important to remark that $\theta_1 = constant$, therefore:

$$jr_4\omega_4 e^{j\theta_4} = jr_2\omega_2 e^{j\theta_2} + jr_3\omega_3 e^{j\theta_3} \tag{18}$$

where:

$$\omega_2 = \frac{d\theta_2}{dt} \tag{19}$$

$$\omega_3 = \frac{d\theta_3}{dt} \tag{20}$$

$$\omega_4 = \frac{d\theta_4}{dt} \tag{21}$$

Dividing (18) by $je^{j\theta_3}$:

$$r_4\omega_4 e^{j(\theta_4-\theta_3)} = r_2\omega_2 e^{j(\theta_2-\theta_3)} + r_3\omega_3 \tag{22}$$

Appliying Newton's formula into (22), it can be writen as:

$$r_4\omega_4[cos(\theta_4-\theta_3)+jsin(\theta_4-\theta_3)] = r_2\omega_2[cos(\theta_2-\theta_3)+jsin(\theta_2-\theta_3)] + r_3\omega_3 \tag{23}$$

Taking the imaginary part of (23)

$$r_4\omega_4 sin(\theta_4-\theta_3) = r_2\omega_2 sin(\theta_2-\theta_3) \tag{24}$$

From (24), the angular velocity ω_4 can be calculated as follows:

$$\omega_4 = \left(\frac{r_2}{r_4}\right)\left[\frac{sin(\theta_2-\theta_3)}{sin(\theta_4-\theta_3)}\right]\omega_2 \tag{25}$$

In this mechanism, the input engine is coupled to the shaft `D` and the output motion is coupled to the shaft `C`. That is, the input angular velocity of the mechanism is the angular velocity of the crank (ω_2) and the output angular velocity, is the angular velocity of the rocker (ω_4) which is established in (25). Finally, as the magnitude of the input angular velocity of the four-bar mechanism is constant, the angular position θ_2 is given by Equation (26).

$$\theta_2 = \omega_2 t \tag{26}$$

where $\omega_2 = 6.9$ rad/sec, which is the only parameter in the model, is the constant speed of the input motor.

3. Optimal strategy

Once the kinematic analysis was developed, the design problem must be established as an optimization problem. In order to do this, objective function and constraints must be proposed.

3.1. *Objective functions*

Recalling from Section 2, the four-bar mechanism is designed by using the case of motion generation and its corresponding performance as it is detailed below.

3.1.1. Motion generation

From the kinematic analysis in Subsection 2.1, it was noted that the rocker motion is mechanically constrained to the shaft `C`. Also, the rotational displacement of this element is called θ_4 and it has the maximum and minimum values at the tightening points of the mechanism. The tightening points are when the linkages r_2 and r_3 are collinear. At this points one degree-of-freedom is drop. The tightening points can be observed in Figures 4(a) and 4(b).

FIGURES 4(a) and 4(b) AROUND HERE.

Taking into account the tightening points, the displacement of the angle θ_4 between these positions can be calculated as indicated in Equation 27.

$$\Phi_1 = \theta_{4max} - \theta_{4min} \tag{27}$$

where θ_{4max} and θ_{4min} are computed according with the four-bar configuration as follows:

- Case a) $\theta_1 < 0$:

$$\theta_{4max} = \pi - \left[abs(\theta_1) + arccos\left(\frac{r_1^2 + r_4^2 - (r_3 - r_2)^2}{2r_1r_4}\right)\right] \tag{28}$$

$$\theta_{4min} = \pi - \left[abs(\theta_1) + arccos\left(\frac{r_1^2 + r_4^2 - (r_3 + r_2)^2}{2r_1r_4}\right)\right] \tag{29}$$

- Case b) $\theta_1 = 0$:

$$\theta_{4max} = \pi - arccos\left(\frac{r_1^2 + r_4^2 - (r_3 - r_2)^2}{2r_1r_4}\right) \tag{30}$$

$$\theta_{4min} = \pi - arccos\left(\frac{r_1^2 + r_4^2 - (r_3 + r_2)^2}{2r_1r_4}\right) \tag{31}$$

- Case c) $\theta_1 > 0$:

$$\theta_{4max} = \pi + \left[abs(\theta_1) - arccos\left(\frac{r_1^2 + r_4^2 - (r_3 - r_2)^2}{2r_1r_4}\right)\right] \tag{32}$$

$$\theta_{4min} = \pi + \left[abs(\theta_1) - arccos\left(\frac{r_1^2 + r_4^2 - (r_3 + r_2)^2}{2r_1r_4}\right)\right] \tag{33}$$

It is important to remark that a maximum value for objective function expressed in Equation 27 implies the maximization of the output of the four-bar mechanism.

3.1.2. Operation quality

One of the most popular criteria to evaluate the performance of a four-bar mechanism is the transmission angle (μ). In a general way, the transmission angle is taken as the absolute value of the acute angle of the pair of angles formed at the intersection of the connecting of the two links. This parameter is defined as the angle between the rocker and the connecting rod of the four-bar mechanism. When the rocker and the connecting rod form an acute angle, μ is computed by (34). If the angle between these elements is not acute, then $\mu = 180° - \mu$ (Norton 1997). The transmission angle varies from a minimum to a maximum value along the cycle of r_2. Figures 5(a) and 5(b) show the positions of the minimum and maximum (μ) transmission angle, respectively.

FIGURES 5(a) and 5(b) AROUND HERE.

Therefore, the transmission angle in a four-bar mechanism is defined by Equations (34) and (35):

$$\mu = |\theta_3 - \theta_4| \tag{34}$$

$$\text{if } \mu > \tfrac{\pi}{2} \quad \text{then } \mu = \pi - \mu \tag{35}$$

and the angles θ_3 and θ_4 are defined by Equations 12 and 17.

It is important to remark that, as the transmission angle decreases, the mechanical advantage is reduced and even a small amount of friction will cause the mechanism to lock. A value close to 90° along the cycle of the crank (r_2) is acceptable to ensure good quality of the mechanism (Norton 1997). On the other hand, from Figure 5 it can be observed that the minimum and maximum values for the transmission angles are defined in Equations 36 and 37.

$$\mu_{max} = arccos\left[\frac{r_3^2 + r_4^2 - (r_1 + r_2)^2}{2r_3r_4}\right] \tag{36}$$

$$\mu_{min} = arccos\left[\frac{r_3^2 + r_4^2 - (r_1 - r_2)^2}{2r_3r_4}\right] \tag{37}$$

Then, a function is established in Equation 38 in order to evaluate the performance of the mechanism.

$$\Phi_2 = (\mu_{max} - 90°)^2 + (\mu_{min} - 90°)^2 \tag{38}$$

A minimum value of the relationship established in Equation 38, produces a best performance of the four-bar mechanism.

3.2. *Constraints*

One of the most important issues to be considered in the mechanism design is ensuring that the crank is able to complete a cycle. In order to fulfill the kinematic necessities, a set of constraints is proposed. Such set is related with the mechanism size, symmetry, transmission angle and mobility criterion.

3.2.1. Grashof's law

The Grashof's law provides that: for a plane four-bar linkage, the sum of the lengths shortest and largest links can not be greater than the sum of the lengths of the two remaining links, if a continuous relative rotation between two elements is desired (Norton 1997). Denoting as s and l the shortest and the largest links of the four-bar mechanism and as p and q the other two links, Grashof's law is established as detailed in Equation 39.

$$s + l \leq p + q \tag{39}$$

In the problem tackled in this work, Grashof's law is given by:

$$r_2 + r_3 \leq r_1 + r_4 \tag{40}$$

In addition, to ensure that the solution method produces Grashof mechanisms, it must be fulfill the conditions established in Equations 41 and 42.

$$r_1 \leq r_3 \tag{41}$$

$$r_4 \leq r_3 \tag{42}$$

3.2.2. System size

One of the design considerations used in the four-bar mechanism is the maximum size of the links. Due to the available space, the length is determined between $0.05m$ and $0.5m$, so that the restrictions are presented in Equations 43 to 46.

$$0.05 \leq r_1 \leq 0.5 \tag{43}$$

$$0.05 \leq r_2 \leq 0.5 \tag{44}$$

$$0.05 \leq r_3 \leq 0.5 \tag{45}$$

$$0.05 \leq r_4 \leq 0.5 \tag{46}$$

On the other hand, the angle between the horizontal axis and the reference bar (r_1) is limited between $45°$ and $-45°$, as pointed out in Equation 47.

$$-45° \leq \theta_1 \leq 45° \tag{47}$$

3.2.3. Transmission angle

As it was pointed out before, the transmission angle μ must be greater than $45°$ along the crank cycle, so that it must be fulfill Equation 48

$$45° \leq \mu(r_1, r_2, r_3, r_4, t) \tag{48}$$

where the transmission angle is calculated using Equation 34. It is worth remarking that this design constraint is a dynamic constraint. This fact is because the constraint must be calculated along the whole crank cycle. The presence of this type of constraints makes difficult the usage of a mathematical programming method in order to obtain a solution of the optimization problem unless such constraints are static. In (Mezura-Montes et al. 2008) a mathematical programming method was used in order to solve a mechatronic design problem. From that work, some implementation issues were highlighted regarding constraints: in a general way, linear approximations of the constraints were used by the mathematical programming method, this fact kept the approach from generating feasible solutions. Furthermore, the mathematical programming method required gradient calculation of the constraints.

3.2.4. Rocker's displacement symmetry

With an output whose behavior is symmetric, two advantages have to be noticed. The first one consists in an equal momentum required at each one of the tightening points. The other advantage is presented in an equal displacement through both planar movement axis. To ensure this equality constraint, Equation 49 is established.

$$180^\circ - \theta_{4max} = \theta_{4min} \tag{49}$$

4. Design of the mechanism

Once the criteria and constraints were established, the set of parameters to apply the kinematic design approach must be proposed.

4.1. *Design variables*

In order to carry out the parametric four-bar design, the vector $\vec{x}$ is defined in Equation 50.

$$\vec{x} = (x_1, x_2, x_3, x_4, x_5)^T = (r_1, r_2, r_3, r_4, \theta_1)^T \tag{50}$$

where the first four variables correspond to the lengths of the four-bar links $[m]$ and the last variable is the angle between the horizontal axis and the reference bar $[rad]$. With this set of parameters, the four-bar mechanism is completely described.

4.2. *Optimization problem*

In order to obtain the mechanical four-bar parameter optimal values, a constrained bi-objective optimization problem with one dynamic constraint is proposed in Equations 51 to 58:

$$\underset{\vec{x} \in R^5}{opt} \quad \Phi = [\Phi_1(\vec{x}), \Phi_2(\vec{x})] \tag{51}$$

with

$$\Phi_1(\vec{x}) = -(\theta_{4max} - \theta_{4min})^2 \tag{52}$$

$$\Phi_2(\vec{x}) = (\mu_{max} - \frac{\pi}{2})^2 + (\mu_{min} - \frac{\pi}{2})^2 \tag{53}$$

subject to:

$$g_1(\vec{x}) = x_2 + x_3 - x_1 - x_4 \leq 0 \quad (54)$$

$$g_2(\vec{x}) = x_1 - x_3 \leq 0 \quad (55)$$

$$g_3(\vec{x}) = x_4 - x_3 \leq 0 \quad (56)$$

$$g_4(\vec{x}, t) = \frac{\pi}{4} - \mu(x, t) \leq 0 \quad (57)$$

$$h_1(\vec{x}) = \pi - \theta_{4max} - \theta_{4min} = 0 \quad (58)$$

$$0.05 \leq x_i \leq 0.5, i = 1, \ldots, 4, \text{ and } -\frac{\pi}{4} \leq x_5 \leq \frac{\pi}{4} \quad (59)$$

where $\Phi_1(\vec{x})$ (the output of the four-bar mechanism) must be maximized and $\Phi_2(\vec{x})$ (the differences of the transmission angle values with respect to 90°) must be minimized. $\Phi_1(\vec{x})$, the output of the mechanism, must be equal or greater than zero. Therefore, this value is squared. Moreover, with the aim to get the two objective functions minimized $\Phi_1(\vec{x})$ is multiplied by -1 in Equation 52. Finally, it is important to notice that the time interval used to evaluate dynamic constraints and the angles θ_3 and θ_4 is the time interval of one crank cycle.

5. Multi-Objective Modified Bacterial Foraging Optimization Algorithm (MOMBFOA)

As it was mentioned in Section 1, MBFOA is based on BFOA. Both of them are based on social and cooperative behaviors bacteria have when looking for regions with high nutrient levels (Bremermann 1974). This section starts by explaining the foraging behavior of bacteria. After that, the original BFOA and its modified version for engineering design problems MBFOA are detailed. Finally, the modifications made to MBFOA to solve the four-bar mechanism design problem is presented.

5.1. *Bacteria foraging behavior*

Each bacterium tries to maximize its obtained energy per each unit of time spent on the foraging process while avoiding noxious substances. Moreover, bacteria can communicate among them. A swarm of bacteria S behaves as follows (Passino 2002):

(1) Bacteria are randomly distributed in the map of nutrients.
(2) Bacteria move towards high-nutrient regions in the map by means of chemotaxis (tumble and swimming). Those located in regions with noxious substances or low-nutrient regions will die and disperse, respectively. Bacteria in high-nutrient regions will reproduce (split).
(3) Bacteria are now located in promising regions of the map of nutrients as they try to attract other bacteria by generating chemical attractors. 1G
(4) Bacteria are now located in the highest-nutrient region.
(5) Bacteria now disperse as to look for new nutrient regions in the map.

5.2. *BFOA*

Four main processes can summarize the bacterial foraging behavior: (1) chemotaxis, (2) swarming, (3) reproduction and (4) elimination-dispersal. Based on them, Passino (Passino

2002) proposed BFOA to solve unconstrained numerical single-objective optimization problems. The algorithm is detailed in Table 2

.

TABLE 2 AROUND HERE.

A bacterium i represents a potential solution to the optimization problem (i.e., a vector of real numbers which includes the n design variables of the optimization problem and identified as $\vec{x}$ in Section 4.1) $[x_1, x_2, \ldots, x_n]$, and it is defined as $\theta^i(j,k,l)$, where j is its chemotactic loop value, k is its reproduction loop value, and l is its elimination-dispersal loop value. The tumble within the chemotactic loop of each bacterium i was modeled by Passino with the generation of a random search direction $\phi(i)$ as presented in Equation 60.

$$\phi(i) = \frac{\Delta(i)}{\sqrt{\Delta(i)^T \Delta(i)}} \tag{60}$$

where $\Delta_(i)$ is a randomly generated vector of size n with elements within the following interval: $[-1, 1]$. After that, each bacterium i modifies its positions by a swimming step as indicated in Equation 61.

$$\theta^i(j+1,k,l) = \theta^i(j,k,l) + C(i)\phi(i) \tag{61}$$

The swim will be repeated N_s times if and only if the new position is better than the previous one, i.e., (assuming minimization) $f(\theta^i(j+1,k,l)) < f(\theta^i(j,k,l))$. Otherwise, a new tumble is computed. The chemotactic loop stops when the chemotactic loop limit N_c is reached for all bacteria in the swarm. The swarming process is based on a modification to the fitness landscape by making more attractive the boundaries of the location of a given bacterium with the aim to attract more bacteria to such region. This process requires the definition of a set of parameter values by the user. (Passino 2002). The reproduction process consists on sorting all bacteria in the swarm $\theta^i(j,k,l), \forall i,\ i = 1, \ldots, N_b$ based on their objective function value $f(\theta^i(j,k,l))$ and eliminating half of them with the worst value. The remaining half will be duplicated so as to maintain a fixed population size. The elimination-dispersal process consists on eliminating each bacteria $\theta^i(j,k,l), \forall i,\ i = 1, \ldots, N_b$ with a probability $0 \leq P_{ed} \leq 1$.

5.3. *MBFOA*

MBFOA was proposed with the aim of adapting BFOA for solving constrained numerical single-objective optimization problems. Furthermore, the original algorithm was simplified and some parameters were eliminated so as to make it easier to use in engineering design problems (Mezura-Montes and Hernández-Ocaña 2009).

MBFOA adopted a penalty-function-free constraint-handling technique to bias the search to the feasible region of the search space, as well as a generational loop (G) where three inner processes are carried out: chemotaxis, reproduction and elimination-dispersal. Based on the fact that in MBFOA there are no reproduction and elimination-dispersal loops because they are replaced by the generational loop, a bacterium i in generation G and in chemotactic step j is defined as $\theta^i(j,G)$.

The four modifications in MBFOA are detailed as follows:

(1) The constraint handling mechanism modifies the selection criteria used in the tumble-swim operator and in the sorting carried out in the reproduction process. The only criterion used in the original BFOA was the objective function value. Instead, in MBFOA a set of three rules proposed by Deb (2000) are employed. They are the following:
   a) Between two feasible bacteria, the one with the best objective function value is preferred.
   b) Between one feasible bacterium and one infeasible bacterium, the feasible one is preferred.
   c) Between two infeasible bacteria, the one with the lowest sum of constraint violation is preferred.

   The sum of constraint violation is computed as: $\sum_{i=1}^{m} max(0, g_i(\vec{x}))$, where m is the number of constraints of the problem. Each equality constraint is converted into an inequality constraint: $\parallel h_i(x) \parallel - \in \leq 0$, where $\in$ is the tolerance allowed (a very small value).

(2) The chemotactic process consists, as in BFOA, on tumble-swim movements carried out by bacteria in the current swarm. The tumble movement (i.e., search direction chosen at random) is computed in the same way as in BFOA (see Equation 60). The swim movement is similar as that used in BFOA as well, but based on the modified nomenclature it is now defined in Equation 62.

$$\theta^i(j+1, G) = \theta^i(j, G) + C(i)\phi(i) \tag{62}$$

   where $\theta^i(j{+}1,G)$ is the new position of bacterium i (new solution), $\theta^i(j,G)$ is the current position of bacterium i.
   Unlike BFOA, where the stepsize values in vector $C(i)$ must be provided by the user, in MBFOA they are calculated by considering the valid limits per each design variable k (Mezura-Montes and Hernández-Ocaña 2009) as indicated in Equation 63.

$$C(i)_k = R * (\frac{\Delta\vec{x}_k}{\sqrt{n}}), k = 1, ..., n \tag{63}$$

   where $\Delta\vec{x}_k$ is the difference between upper (U_k)and lower (L_k) limits for design parameter x_k: $U_k - L_k$, n is the number of design variables and R is a user-defined percentage of the value used by the bacteria as stepsize.
   Furthermore, instead of the swarming process which requires four user-defined parameters, an attractor movement was added to MBFOA so as to let each bacterium in the swarm to follow the bacterium located in the most promising region of the search space in the current swarm, i.e., the best current solution. The attractor movement, proposed by (Mezura-Montes and Hernández-Ocaña 2009) for MBFOA, is presented in Equation 64.

$$\theta^i(j+1, G) = \theta^i(j, G) + \beta(\theta^B(G) - \theta^i(j, G)) \tag{64}$$

   where $\theta^i(j+1,G)$ is the new position of bacterium i, $\theta^i(j,G)$ is the current position of bacterium i, $\theta^B(G)$ is the current position of the best bacterium in the swarm so far at generation G, and β defines the closeness of the new position of bacterium i with respect to the position of the best bacterium $\theta^B(G)$. The attractor movement applies twice in a chemotactic loop, while in the remaining steps the tumble-swim movement is carried out. The aim is to promote a balance between exploration and exploitation in the search. Finally, if the value of a design variable generated by the tumble-swim or attractor operators is outside the valid limits defined by the optimization problem, the following modification, taken from (Kukkonen and Lampinen 2006) is made so as to get the value within the valid range as showed in Equation 65.

$$x_i = \begin{cases} 2 * L_i - x_i & \text{if } x_i < L_i \\ 2 * U_i - x_i & \text{if } x_i > U_i \\ x_i & \text{otherwise} \end{cases} \tag{65}$$

where x_i is the design variable value i generated by any bacterium movement, and L_i and U_i are the lower and upper limits for design variable i, respectively.

(3) The reproduction process in MBFOA is similar as that used in BFOA, i.e., the swarm is sorted and the first half survives while the second half is eliminated. The first half is also duplicated to maintain a fixed size of the swarm at each cycle. However, in MBFOA the criteria to sort the swarm of bacteria are the three rules of the constraint-handling technique and not only the objective function value as in BFOA.

(4) The elimination-dispersal process eliminates only the worst bacterium, based on the three criteria defined in the constraint-handling technique, and a new randomly generated bacterium is inserted as a replacement.

The complete pseudocode of MBFOA is detailed in Table 3

.

TABLE 3 AROUND HERE.

5.4. *MOMBFOA*

Recalling from Section 4, the problem defined and tackled in this paper is a constrained bi-objective optimization problem with one dynamic constraint. Therefore, MBFOA was modified so as to get the multi-objective MBFOA (MOMBFOA). The details are presented in the following subsections.

5.5. *Pareto dominance and constraint-handling mechanism*

It is well-documented in the specialized literature that the most suitable criterion to be added to an EA or a SIA so as to solve either a two- or three-objective optimization problem is *Pareto dominance* (Deb 2002), which is defined as follows: a vector of objectives $\Phi = [\Phi_1, \dots, \Phi_K]$ is said to Pareto dominate $\Phi' = [\Phi'_1, \dots, \Phi'_K]$ (denoted by $\Phi \preceq \Phi'$) if and only if Φ is partially less than Φ', i.e. $\forall i \in \{1, \dots, K\},\ \Phi_i \leq \Phi'_i \wedge \exists i \in \{1, \dots, K\} : \Phi_i < \Phi'_i$. The set of all the Pareto non-dominated solutions is called the *Pareto optimal set*. The objective function values of those solutions contained in the Pareto optimal set constitute the *Pareto front* of the problem. In this paper, as stated in Section 4, the mechanical design problem has two objectives, then $K = 2$.

The solution of a multi-objective problem can be defined as follows: If the feasible region of the search space is named as $\mathcal{F}$, either the EA or the SIA will look for the Pareto optimal set: $\mathcal{P}^* := \{\vec{v} \in \mathcal{F} \mid \neg\exists\ \vec{s} \in \mathcal{F}\ \ \Phi(\vec{s}) \preceq \Phi(\vec{v})\}$.

In this paper a real-world problem is being solved, then $\mathcal{P}^*$ is unknown, a sub-optimal Pareto set, including sub-optimal trade-off solutions for the mechanical design problem is the solution sought.

5.5.1. Selection criteria

As it was found in the specialized literature, constraint-handling in numerical multi-objective optimization problem has received less attention than the single-objective optimization problem (Yen 2009, Ray et al. 2009). Nevertheless, there are competitive proposals such as the combination of the feasibility rules described in Section 5.3 with Pareto dominance for non-dominated sorting (Deb et al. 2002). There are other similar proposals (Oyama et al. 2007), where the third rule considered Pareto dominance in the constraint search space when two solutions are infeasible. Based on its similarity with the constraint-handling technique used in MBFOA, the feasibility rules described with Pareto dominance (Deb et al. 2002) is adopted in MOMBFOA. This criteria will apply in the tumble-swim movements and in the sorting of the swarm. The set of criteria are defined as follows:

a) Between two feasible bacteria, the one which dominates the other is preferred. If both bacteria are feasible and non-dominated each other, one is chosen at random.
b) Between one feasible bacterium and one infeasible bacterium, the feasible one is preferred.
c) Between two infeasible bacteria, the one with the lowest sum of constraint violation is preferred.

5.5.2. Dynamic constraint-handling

One particular aspect of the four-bar mechanism optimal design problem introduced in Section 3.2.3 and stated in Section 4 is the presence of a dynamic inequality constraint ($g_{14}(\vec{x})$), which keeps the transmission angle μ over 45° along the whole crank cycle. Although the usage of EAs and SIAs in dynamic optimization problems has been documented in the specialized literature (Yang et al. 2007), the treatment of dynamic constraints has been scarcely investigated. In (Kumar-Singh et al. 2009) a framework for unconstrained dynamic optimization was adapted to solve problems with dynamic constraints. In (Nguyen and Yao 2009) constraints were handled by a penalty function and a special operator to convert infeasible solutions into feasible ones. The authors concluded that a special treatment to the constraints is more suitable than keeping or maintaining diversity in the population as in unconstrained dynamic optimization (Yang et al. 2007). Motivated by this last finding, in this paper a special treatment is applied to the dynamic constraint as follows: the evaluation of such constraint is discretized in such a way it is evaluated along the trajectory described by the four-bar mechanism, i.e., one crank cycle. The number of inner evaluations of the dynamic constraint is 900. After that, the sum of violations for each inner evaluation is computed and added to the sum of constraint violation for a given bacterium (see Section 5.3).

5.6. *External archive*

Inspired in state-of-the-art algorithms for multi-objective optimization (Zitzler and Thiele 1999, Knowles and Corne 2000) an external archive to store only feasible non-dominated bacteria is considered in MOMBFOA. The archive, which is empty at the beginning of the optimization process, is updated at each MOMBFOA cycle by inserting a copy of those feasible bacteria from the swarm after the chemotactic process. Each time a set of bacteria enters the file, a non-dominance checking is carried out in the archive to only keep non-dominated bacteria.

5.7. *Diversity mechanism*

With the aim to maintain a suitable diversity in the swarm of bacteria, as recommended for EAs and SIAs to solve multi-objective optimization problems (Deb 2002, Coello Coello et al. 2002), and taking care of not adding additional parameters as considered in the constraint-handling mechanism integrated with Pareto dominance, the *crowding distance* (Deb 2002) was adopted and employed in non-dominated feasible solutions in the external archive. *Crowding distance* values are higher for those more isolated solutions in the objective space. Therefore, bacteria with higher *crowding distance* values in the archive will be preferred. Table 4 details the calculation of such distance

.

TABLE 4 AROUND HERE.

5.8. *Attractor Operator*

Recalling from Section 5.3 and Equation 64, an operator was added to MBFOA to allow the best bacterium in the swarm to attract other bacteria during their chemotactic process. In single-objective optimization the selection of the best bacterium is straightforward because one unique solution is sought. However, in multi-objective optimization, as indicated in Section 5.5, a set of solutions is looked for. Therefore, the selection of the best bacterium must be designed to bias the search to a sub-optimal Pareto front. Hence, in MOMBFOA the leader will be chosen from the external archive and it will be the bacterium with the highest crowding distance value, i.e., the solution located in the most isolated region of the current Pareto front. The goal is to promote the sampling of more solutions in such region and extend the front. The updated attractor operator is shown in Equation 66.

$$\theta^i(j+1,G) = \theta^i(j,G) + \beta(\theta^{BC_r}(G) - \theta^i(j,G)) \tag{66}$$

where $\theta^i(j+1,G)$ is the new position of the bacterium i, $\theta^i(j,G)$ is the current position of bacterium i, and $\theta^{BC_r}(G)$ is the current position of the bacterium with the best crowding distance in generation G taken from the external file. β is the stepsize of the attractor movement. If the archive is empty, i.e., no feasible non-dominated solutions have been found, the leader is chosen from the current swarm and based on the criteria defined in Section 5.5.1, i.e., the bacterium with the lowest sum of constraint violation. The aim there is to bias the search to the feasible region.

5.9. *Reproduction*

Based on the fact that in MBFOA the reproduction process consists on a cloning-like process where an identical copy of a bacterium is generated as offspring, the diversity in the swarm might decrease fast. On the other hand, as in multi-objective optimization problems a set of solutions is seek, diversity must be maintained. Therefore, in MOMBFOA only the best bacterium in the swarm will reproduce with one copy which will replace the second worst bacterium, based on a sorting process computed by considering the criteria defined in Section 5.5.1, i.e., either the bacterium with the second to last highest sum of constraint violation (if the swarm contains either no feasible bacteria or a combination of feasible and infeasible bacteria) or a dominated bacteria (if the swarm has either only feasible bacteria or just one infeasible bacteria). The complete MOMBFOA pseudocode is presented in Table 5

.

TABLE 5 AROUND HERE.

6. Results and discussion

To evaluate the behavior and performance of MOMBFOA four experiments were designed.

(1) Analysis of the constraint satisfaction.
(2) Comparison of sub-optimal Pareto fronts.
(3) Comparison based on performance metrics.
(4) Analysis of the solutions obtained in the sub-optimal Pareto set per each compared algorithm.

For comparison purposes, four meta-heuristic algorithms were chosen to solve the four-bar mechanism optimal design problem: (1) NSGA-II (Deb et al. 2002), (2) SPEA2 (Zitzler et al. 2002), (3) the differential evolution for mechanical design (DEMD) (Portilla-Flores et al. 2011), and (4) the multi-objective particle swarm optimization algorithm MOPSO (Reyes-Sierra and Coello Coello 2006). NSGA-II and SPEA2 were chosen based on the fact that they are the most popular EAs to solve multi-objective optimization problems. Moreover, NSGA-II uses crowding distance as a diversity mechanism while SPEA2 uses an external file to store nondominated solutions. DEMD was precisely proposed to solve multi-objective mechanical design problems and it also uses crowding distance as diversity mechanism as well as an external archive for elitism purposes. MOPSO is based on particle swarm optimization and has proved to be very competitive to solve multi-objective optimization problems in presence of constraints Reyes-Sierra and Coello Coello (2006). The four algorithms adopted the same constraint-handling technique used by MOMBFOA for the dynamic constraint.

10 independent runs were carried out by each compared algorithm (MOMBFOA, NSGA-II, SPEA2, DEMD, and MOPSO) in each experiment. The set of parameter values remained fixed for each algorithm in all experiments and $500,000$ evaluations was defined as the termination criteria for each independent run.

The parameter values for each algorithm were obtained after preliminary experiments by considering different combinations and choosing the one which provided the best performance. A summary of the preliminary tests is presented in Table 6, where the hypervolume metric was adopted as a criterion to choose the best performance of each algorithm. This decision was based on the fact that it does not require the real Pareto front for a single algorithm analysis. The preliminary tests were by no means exhaustive. This was because of the computational time required by each single run for each compared algorithm derived mostly from the objective function and constraints (including the dynamic one) evaluations of this real-world optimization problem. The results in Table 6 do not include those which provided the best results for each algorithm. They are presented in Table 8.

TABLE 6 AROUND HERE

The values for MOMBFOA were: $S_b = 200$, $GMAX = 125$, $N_c = 20$ (the tumble-swim operator works in all chemotactic steps for each bacterium with the exception of steps 15 and 20, where the attractor operator is used) , $\beta = 1.7$ and $R = 1.8$E-3. Particulary for MOMBFOA, previous experiments were carried out to choose the best combination of input parameter values. Different combination of values per parameter were defined and tested in 5 runs and based on the filtered Pareto fronts obtained the above mentioned values were adopted in the experiments. It is important to remark that the most critical parameter to be fine-tuned is R, and part of the future work relies on its analysis.

For NSGA-II the parameter values were: popsize = 100, GMAX = 5000, crossover rate = 0.8, mutation rate = 0.3. For DEMD the values were: NP= 100, $GMAX = 5000$, $CR \in [0.8, 1]$ generated at random, and $F \in [0.3, 0.9]$ also generated at random. The randomness for F and CR were adopted because DEMD performs better with such algorithm. For SPEA2 the parameters values were: popsize = 200, GMAX = 2500, crossover rate = 1.0 and mutation rate = 1.0. Finally, for MOPSO the parameters adopted were popsize = 100, GMAX = 5000, $\phi_1 = 2.5$, and $\phi_2 = 1.7$. The remaining values were kept fixed with suggested values: repository size = 100, $\alpha = 0.1$, ngrid = 10, $\beta = 4$, and $\gamma = 1$.

The five algorithms were coded in Matlab R2009b and were run in a laptop computer with 3GB RAM, Intel Core Duo processor at 2.8GHz, and Microsoft Windows XP OS.

6.1. *Constraint satisfaction*

The experimental design related with MOMBFOA's behavior when dealing with the constraints of the problems consisted on the following : (1) An analysis of the population at each generation with respect to the satisfaction of the dynamic constraint, (2) an analysis of the population at each generation with respect to the capability to generate feasible solutions, and (3) the capability to generate feasible non-dominated solutions,

The results for the first experiment are summarized in Figure 6, where the average number of solutions in the population per generation which satisfy the dynamic equality constraint on 10 independent runs are plotted.

FIGURE 6 AROUND HERE.

From Figure 6, three stages can be distinguished in the behavior of the approach: (1) between generations 0 and 5, where MBFOA quickly gets more than half of its population with solutions satisfying the dynamic constraint, (2) between generations 6 and 40, where another 20% of the population also satisfies the dynamic constraint, and finally (3) between generations 41 and 125, where another 15% of the population satisfies the aforementioned constraint but with little "ups" and "downs", i.e., slight decreasing in the number of solutions satisfying the constraint.

The overall behavior in Figure 6 shows that the proposed approach favors the satisfaction of the dynamic constraint. However, this effect cannot be isolated from the capability of the approach to generate feasible solutions. Therefore, the average number of feasible solutions per generation on 10 independent runs are presented in Figure 7, where it is observed that, after generation 22, half of the population is feasible. After that, between generations 23 and 80, another 40% of the population becomes feasible. At the end, between generations 81 and 125, almost the remaining 10% of the population becomes feasible.

FIGURE 7 AROUND HERE.

Finally, the average number of feasible non-dominated solutions stored in the external file per each generation in 10 independent runs are plotted in Figure 8.

A similar behavior with respect to those found in the previous two experiments is observed, but with a significant lower number of solutions. Between generations 0 and 20, six feasible non-dominated solutions are stored in the external file. In the remaining generations, a slower but increasing behavior is observed, where another six feasible non-dominated solutions are added to the file.

FIGURE 8 AROUND HERE.

The overall results of the three experiments regarding the constraint satisfaction suggest that MBFOA, coupled with the proposed constraint-handling mechanism designed to also deal with the dynamic constraint of the problem, quickly locates some bacteria of the population in the feasible region of the search space. This behavior is an attractive property of the approach for mechanical engineers because a solution can be provided by requiring a low number of evaluations. The way almost the whole population gradually enters the feasible region is another convenient behavior observed in MBFOA because maintaining infeasible solutions in the population has been reported in the specialized literature as convenient to avoid local fronts (Ray et al. 2009). This same effect was observed specifically for the dynamic equality constraint, where some solutions remained infeasible even in the last generations.

6.2. *Sub-optimal Pareto fronts comparison*

The comparison of the final sub-optimal Pareto fronts obtained by each algorithm is presented in Figure 9, where the filtered fronts are plotted. Those fronts were obtained by applying non-dominance checking to all the solutions from the 10 fronts obtained in the 10 independent runs per algorithm.

FIGURE 9 AROUND HERE.

From the graphical analysis it is clear that both, MOMBFOA and DEMD outperformed NSGA-II, MOPSO, and SPEA2. MOMBFOA found some feasible solutions in the central and left regions in its front, which dominate solutions located in the central part of the DEMD's front. On the other hand, DEMD was able to generate more solutions in the upper left region of the objective space, while MOMBFOA reached the lower right region. The results of this experiment suggests better solutions provided by MOMBFOA. NSGA-II generated three well-defined regions of a disconnected sub-optimal Pareto front while SPEA2 generated a widely spaced front with the lowest value for the second objective function. It is important to remark that those few solutions obtained by MOPSO are infeasible (constraints g_2 and h_1 are violated).

6.3. *Comparison based on performance metrics*

Performance metrics in nature-inspired multi-objective optimization are a valuable tool to analyzing the performance of algorithms (Zitzler et al. 2003). In this work, three metrics will be employed to analyze the results found by each compared algorithm and they are described next:

- **Two Set Coverage (C-Metric):** This binary performance measure estimates the coverage proportion, in terms of percentage of dominated solutions, between two sub-optimal Pareto sets. Given two sets A and B, both containing only feasible non-dominated solutions, the C-Metric is formally defined as:

$$C(A,B) = \frac{|\{u \in B | \exists v \in A : v \preceq u\}|}{|B|} \tag{67}$$

The C-measure value means the portion of solutions in B being dominated by any solution in A.

- **Hypervolume (Hv):** Having a sub-optimal Pareto set PF_{known}, and a reference point in objective space z_{ref}, this performance measure estimates the *Hypervolume* attained by it. The Hypervolume corresponds to the non-overlapping volume of all the hypercubes formed by the reference point (z_{ref}) and every vector in the Pareto set approximation. This is formally defined as:

$$HV = \{\cup_i vol_i | vec_i \in PF_{known}\} \tag{68}$$

where vec_i is a non-dominated vector from the sub-optimal Pareto set, and vol_i is the volume for the hypercube formed by the reference point and the non-dominated vector vec_i. A High value for this measure indicates that the sub-optimal Pareto set is closer to the true Pareto front and that it covers a wider extension of it.

- **Generational Distance (GD):** It is a way of estimating how far are the elements of the sub-optimal Pareto front obtained by one algorithm with respect to those of the true Pareto front of the problem. It is defined as follows:

$$GD = \frac{\sqrt{\sum_{i=1}^{N} d_i^2}}{N} \tag{69}$$

where N is the number of non-dominated solutions in the sub-optimal Pareto front obtained by the algorithm, and d_i is the Euclidean distance between each point in such front and its nearest member of the true Pareto front. Due to the fact that a real-world problem is being solved in this paper, the true Pareto front will be approximated by applying non-dominance checking to the combined set of the accumulated Pareto fronts obtained in a set of independent runs by each compared algorithm. A value of $GD = 0$ points out that all the elements of the sub-optimal front generated by an algorithm are in the true Pareto front. As the GD value increases, so is the distance between the sub-optimal and true Pareto fronts. Therefore, lower values indicate a better performance of an algorithm.

The statistical results obtained for the C-Metric are summarized in Table 7. Such results are based on all the pairwise combinations of the ten independent runs executed by each algorithm (each one of the ten sub-optimal Pareto fronts of one algorithm was compared with each one the ten fronts obtained by the other algorithm). Mann-Whitney non-parametric test was used to get confidence on the differences observed in the samples of runs (Derrac et al. 2011)

.

TABLE 7 AROUND HERE.

The results in Table 7 show that MOMBFOA and DEMD outperform NSGA-II, MOPSO, and SPEA2 in the four-bar mechanism optimization problem. It is worth noting that NSGA-II and SPEA2 showed a similar performance based on the C-metric and that MOPSO was competitive only with respect to NSGA-II. In the same way, such metric is unable to determine the superiority between MOMBFOA and DEMD.

The results for the Hypervolume metric on 10 independent runs are summarized in Table 8. The reference point to calculate that metric was (0,0.7). Based on the fact that the Mann-Whitey test gives 95% confidence on the differences observed in such table, DEMD outperformed the other four algorithms while MOMBFOA outperformed NSGA-II, SPEA2, and MOPSO. Those findings confirm the observations made in Figure 9, where DEMD generated a larger sub-optimal Pareto front with more solutions with respect to that generated by MOMBFOA

.

TABLE 8 AROUND HERE.

To further analyze the behavior of the four algorithms with respect to this metric, Figure 10 shows the average hypervolume value from 2 independent runs at different evaluation numbers. The values confirm the superiority of DEMD with respect to the hypervolume metric.

FIGURE 10 AROUND HERE.

Table 9 summarizes the statistical values on 10 independent runs by each compared algorithm for the Generational Distance metric. The Mann-Whitney test indicates that there is no significant difference between the results of MOMBFOA and DEMD. The same applies for the results between NSGA-II and SPEA2. In the remaining pairwise combinations the differences are significant; it means that the MOMBFOA and DEMD provide a similar better performance with respect to those observed by NSGA-II, SPEA2 and MOPSO.

TABLE 9 AROUND HERE.

6.4. *Analysis of solutions*

With the aim to provide a more detailed review of the solutions provided by MOMBFOA and the four other compared algorithms (DEMD, NSGA-II, SPEA2, and MOPSO) some of them, located in the region of the objective space with more solutions generated by the compared algorithms, were analyzed and subject to simulation so as to get more information on their suitability in the solution of the four-bar mechanism. The solutions chosen were those located at the middle of each front within the region with more solutions. Figure 11 remarks in a rectangle such region in the objective space ($\Phi_1 \in [0.4, 0.8]$ and $\Phi_1 \in [0.3, 0.5]$).

FIGURE 11 AROUND HERE.

Table 10 encloses those five representative solutions by each compared algorithm. Those solutions, except with that infeasible one provided by MOPSO, were introduced in a simulator of the four-bar mechanism. The inputs for the simulator are the five design variables defined in Section 4.1 (x_1, x_2, x_3, x_4, and x_5)
.

TABLE 10 AROUND HERE.

From a mechanical point of view, angle θ_4 must have a displacement over the vertical axis, i.e., by considering a reference angle of 90° (1.5707 radians) θ_4 must have a positive/negative value variation. In the same way, the transmission angle μ must apply a strong impulse force to the bar where angle θ_4 is located, i.e., the μ angle must oscillate by starting with a high value (close to 1 radian), decreasing to a similar value (1 radian) and be closer with respect to the θ_4 angle value. As a consequence, the movement of the mechanism is improved.

Regarding the value taken by the displacement angle θ_4, its variation must be between −45° and 45°, i.e., between 135° (2.3561 radians) and 45° (0.7854 radians) with respect to the reference angle value of 90° (1.5707 radians). On the other hand, the transmission angle μ, for a good performance of the four-bar mechanism, must have a minimum value of 45°.

The simulation results for the displacement angle θ_4 and the transmission angle μ, based on the representative solutions in Table 10, are showed in Figure 12, where the lower and upper horizontal dashed lines represent the lower 45° (0.7854 radians) and upper 135° (2.3561 radians) limits allowed for θ_4 and μ angles, and the horizontal dotted line in the middle of the graph indicates the reference value of 90° (1.5707 radians).

FIGURE 12 AROUND HERE.

Based on the information in Figure 12, four out of five algorithms obtained valid solutions, i.e., the lower and upper limits for both angles are not violated. However, The solution provided by SPEA2 is at the edge of the lower limit value for the μ angle (0.7854 radians). Regarding closeness between angles, the solutions by MOMBFOA and DEMD where the best ones, while the solutions by NSGA-II and SPEA2 show an increasing distance after 0.3 seconds.

The solutions obtained by MOMBFOA and DEMD start and end the μ angle value with 1 radian. Meanwhile, the solutions reached by NSGA-II and MOMBFOA start and end with lower values for the μ angle. Finally, between the solution by MOMBFOA and the one obtained by DEMD, the first one has the higher amplitude for the displacement of the rocker θ_4, i.e., it reaches a value of 2.0 radians. Furthermore, the transmission angle μ remains more time over 90° (the ideal behavior). This observations confirm the comments made to Figure 9, where even with a lower number of solutions, MOMBFOA was able to find some of them which dominate those found by DEMD.

7. Conclusions and future work

The synthesis of a four-bar mechanism based on a kinematic analysis and the optimization of its design by using the multi-objective modified bacterial foraging optimization algorithm were presented. The design problem was defined by taking into account two objective functions and structural constraints, where at least one of them was a dynamic constraint.

MOMBFOA adopted Pareto dominance and the feasibility rules as selection criteria to bias the approach to the feasible region of the search space. Furthermore, the dynamic constraint was specially treated so as to promote its satisfaction by evaluating it 900 times so as to get a good approximation of its violation. An external file was used to store feasible non-dominated solutions found during the search and crowding distance was utilized for diversity maintenance. Finally, the reproduction of bacteria was limited to a very small number of clones per generation and an attractor operator was added to the chemotactic cycle so as to let bacteria to explore the neighborhood of isolated feasible non-dominated solutions.

Four experiments were carried out to test the behavior and performance of MOMBFOA in the solution of the real-world mechanical design problem. MOMBFOA was able to generate feasible solutions (including the dynamic constraint) early in the optimization process, but at the same time infeasible solutions were maintained and this balance caused, as it was later observed, better final results. The results obtained by MOMBFOA were compared with respect to those obtained by three state-of-the-art algorithms for multi-objective optimization (NSGA-II, SPEA2 and MOPSO) and one evolutionary algorithm designed to solve multi-objective mechanical design problems (DEMD). The visual comparison of the accumulated sub-optimal Pareto fronts obtained by each algorithm showed that the generated by MOMBFOA, even with a lower number of non-dominated solutions, was better with respect to those generated by DEMD, NSGA-II, SPEA2, and MOPSO. On the other hand, the comparison based on three performance metrics (C-metric, Hypervolume and Generational Distance) showed that MOMBFOA and DEMD were clearly better than NSGA-II, SPEA2, and MOPSO. Moreover, MOMBFOA and DEMD were equally good based on the C-metric and Generational Distance, while DEMD was slightly better than MOMBFOA for the Hypervolume metric due to its larger sub-optimal Pareto front. Finally, a review of representative solutions located in the most populated region of the objective space, showed that the solution obtained by MOMBFOA presented the best performance in a simulation-based analysis, where both angles related with the input force generated by the four-bar mechanism had the most competitive values.

From a theoretical point of view, the complexity of MOMBFOA is similar to that of SPEA2, $O(G(N + A)^2)$ (Jensen 2003), where G is the number of generations of the algorithm, N is the population size and A is the archive size. The similarities lie in the usage and update of the external archive where feasible non-dominated solutions are stored. DEMD also has a similar complexity due to its archive usage, while NSGA-II has a complexity of $O(GMN^2)$ (Jensen 2003).

The results presented in this paper validated the model proposed for the four-bar mechanism. Furthermore, the overall behavior of MOMBFOA (smaller sub-optimal Pareto front but with better solutions) showed to be suitable for this type of mechanical design problems.

Part of the future work relies on the incorporation of the design engineer preferences into MOMBFOA. Moreover, MOMBFOA will be used to solve other real-world mechanical design problems, specially those related with concurrent design, where both, the mechanical design and the control of the mechanism, are optimized at the same time. Finally, MOMBFOA will be analyzed to determine the sensitivity to its parameter values, mainly with respect to that related to the stepsize used in the chemotactic loop (R).

8. Acknowledgements

The first author acknowledges support from the Mexican Council for Science and Technology (CONACyT) through project No. 79809. The second author acknowledges support from the COFAA and EDI programs of the Instituto Politécnico Nacional through project No. SIP-20111027 and support from SNI-CONACyT. The third author acknowledges support from CONACyT through a scholarship to pursue graduate studies at LANIA.

References

Bremermann, H. (1974), "Chemotaxis and Optimization," *J. Franklin Inst.*, 297, 397–404.

Coello Coello, C.A., Van Veldhuizen, D.A., and Lamont, G.B., *Evolutionary Algorithms for Solving Multi-Objective Problems*, Kluwer Academic Publishers (2002).

Daneshyari, M., and Yen, G.G. (2012), "Cultural-based particle swarm for dynamic optimisation problems," *International Journal of Systems Science*, 43(7), 1284–1304.

Deb, K. (2000), "An Efficient Constraint Handling Method for Genetic Algorithms," *Computer Methods in Applied Mechanics and Engineering*, 186(2/4), 311–338.

Deb, K., *Multi-Objetive Optimization using Evolutionary Algorithms*, John Wiley Sons, LTD (2002).

Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002), "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA–II," *IEEE Transactions on Evolutionary Computation*, 6(2), 182–197.

Derrac, J., García, S., Molina, D., and Herrera, F. (2011), "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm and Evolutionary Computation*, 1(1), 3–18.

Dorigo, M., Maniezzo, V., and Colorni, A. (1996), "The Ant System: Optimization by a Colony of Cooperating Agents," *IEEE Transactions of Systems, Man and Cybernetics-Part B*, 26(1), 29–41.

Eiben, A., and Smith, J.E., *Introduction to Evolutionary Computing*, Natural Computing Series, Springer Verlag (2003).

Engelbrecht, A.P., *Computational Intelligence. An Introduction*, 2nd ed., John Wiley & Sons (2007).

Fogel, L.J., *Intelligence Through Simulated Evolution. Forty years of Evolutionary Programming*, New York: John Wiley & Sons (1999).

Jensen, M.T. (2003), "Reducing the Run-Time Complexity of Multiobjective EAs: The NSGA-II and Other Algorithms," *IEEE Transactions on Evolutionary Computation*, 7(5), 503–515.

Kennedy, J., and Eberhart, R.C., *Swarm Intelligence*, UK: Morgan Kaufmann (2001).

Knowles, J.D., and Corne, D.W. (2000), "Approximating the Nondominated Front Using the Pareto Archived Evolution Strategy," *Evolutionary Computation*, 8(2), 149–172.

Koza, J.R., Keane, M.A., Streeter, M.J., Mydlowec, W., Yu, J., and Lanza, G., *Genetic Programming IV: Routine Human-Competitive Machine Intelligence*, Hingham, MA, USA: Kluwer Academic Publishers (2003).

Kukkonen, S., and Lampinen, J. (2006), "Constrained Real-Parameter Optimization with Generalized Differential Evolution," in *2006 IEEE Congress on Evolutionary Computation (CEC'2006)*, July, Vancouver, BC, Canada: IEEE, pp. 911–918.

Kumar-Singh, H., Isaacs, A., Thanh-Nguyen, T., Ray, T., and Yao, X. (2009), "Performance of Infeasibility Driven Evolutionary Algorithm (IDEA) on Constrained Dynamic Single Objective Optimization Problems," in *IEEE 2009 Congress on Evolutionary Computation (CEC'2009)*, May, Trondheim, Norway: IEEE Service Center, pp. 3127–3134.

Mermetas, V. (2004), "Optimal design of manipulator with four-bar mechanism," *Mechanism and Machine Theory*, 1(39), 545–554.

Mezura-Montes, E., and Hernández-Ocaña, B. (2009), "Modified Bacterial Foraging Optimization for Engineering Design," in *Proceedings of the Artificial Neural Networks in Enginnering Conference (ANNIE'2009)*, eds. C.H. Dagli and et al., November, Vol. 19 of *Intelligent Engineering Systems Through Artificial Neural Networks*, St. Louis, MO, USA: ASME Press, pp. 357–364.

Mezura-Montes, E., Portilla-Flores, E.A., Coello, C.A.C., Álvarez Gallegos, J., and Cruz-Villar, C.A. (2008), "An Evolutionary Approach to Solve a Novel Mechatronic Multiobjective Optimization Problem," in *Advances in Metaheuristic Methods for Hard Optimization* eds. P. Siarry and Z. Michalewicz, ISBN 978-3-540-72959-4, Berlin: Springer, pp. 329–351.

Mundo, D., and Yan, H. (2007), "Kinematic optimization of ball-screw transmission mechanism," *Mechanism and Machine Theory*, 1(42), 34–47.

Nariman-Zadeh, N., Felezi, M., Jamali, A., and Ganji, M. (2009), "Pareto optimal synthesis of four-bar for path generation," *Mechanism and Machine Theory*, 1(44), 180–191.

Nguyen, T.T., and Yao, X. (2009), "Benchmarking and Solving Dynamic Constrained Problems," in *IEEE 2009 Congress on Evolutionary Computation (CEC'2009)*, May, Trondheim, Norway: IEEE Service Center, pp. 690–697.

Norton, R., *Diseño de Maquinaria, una introducción a la síntesis y al análisis de mecanismos y máquinas*, In Spanish, McGraw-Hill (1997).

Oyama, A., Shimoyama, K., and Fujii, K. (2007), "New constraint-handling method for multi-objective and multi-constraint evolutionary optimization," *Transactions of the Japan Society for Aeronautical and Space Sciences*, 50(167), 56–62.

Pardo, M., and Sánchez, (2006), "Equipo Completo de transmisión unilateral para pedaleo basculante con sistema de cambio de velocidades," Technical report, Patent request 200401700, Spanish Patent and Brand Office.

Passino, K. (2002), "Biomimicry of bacterial foraging for distributed optimization and control," *IEEE Control Systems Magazine*, 22(3), 52–67.

Portilla-Flores, E.A., Mezura-Montes, E., Álvarez Gallegos, J., Coello, C.A.C., Cruz-Villar, C.A., and Villareal-Cervantes, M.G. (2011), "Parametric Reconfiguration Improvement in Non-Iterative Concurrent Mechatronic Design Using an Evolutionary-Based Approach," *Engineering Optimization of Artificial Intelligence*, 24(5), 757–771.

Portilla-Flores, E.A., Mezura-Montes, E., Álvarez Gallegos, J., Coello-Coello, C.A., and Cruz-Villar, C. (2007), "Integration of structure and control using an evolutionaryapproach: an application to the optimal concurrent design of a CVT," *International Journal for Numerical Methods in Engineering*, 8(71), 883–901.

Price, K., Storn, R., and Lampinen, J., *Differential Evolution: A Practical Approach to Global Optimization*, Natural Computing Series, Springer-Verlag (2005).

Radovan, R., and Stevan, R. (2009), "On the optimum synthesis of a four-bar linkage using differential evolution and method of variable controlled deviations," *Mechanism and Machine Theory*, 1(44), 235–246.

Ray, T., Singh, H.K., Isaacs, A., and Smith, W. (2009), "Infeasibility Driven Evolutionary Algorithm for Constrained Optimization," (Vol. 198, ed. E. Mezura-Montes, Springer-Verlag, Studies in Computational Intelligence Series, ISBN:978-3-642-00618-0, pp. 145–165.

Reyes-Sierra, M., and Coello Coello, C.A. (2006), "Multi-Objective Particle Swarm Optimizers: A Survey of the State-of-the-Art," *International Journal of Computational Intelligence Research*, 2(3), 287–308.

Saridakis, K., and Dentsoras, A. (2009), "Integration of genetic optimisation and neuro-fuzzy approximation in parametric engineering design," *International Journal of Systems Science*, 40(2), 131–145.

Schwefel, H.P. (ed.) *Evolution and Optimization Seeking*, New York: Wiley (1995).

Shigley, J., and Uicker, J., *Theory of Machines and Mechanisms*, McGraw-Hill (1995).

Wang, H., Yang, S., Ip, W., and Wang, D. (2012), "A memetic particle swarm optimisation algorithm for dynamic multi-modal optimisation problems," *International Journal of Systems Science*, 43(7), 1268–1283.

Yang, S., Ong, Y.S., and Jin, Y. (eds.) *Evolutionary Computation in Dynamic and Uncertain Environments Evolutionary Computation in Dynamic and Uncertain Environments*, Springer

(2007).

Yen, G.G. (2009), "An adaptive penalty function for handling constraint in multi-objective evolutionary optimization," (Vol. 198, ed. E. Mezura-Montes, Springer-Verlag, Studies in Computational Intelligence Series, ISBN:978-3-642-00618-0, pp. 121–143.

Zitzler, E., and Thiele, L. (1999), "Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach," *IEEE Transactions on Evolutionary Computation*, 3(4), 257–271.

Zitzler, E., Laumanns, M., and Thiele, L. (2002), "SPEA2: Improving the Strength Pareto Evolutionary Algorithm," in *Proceedings of the Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems Conference (EUROGEN'2001)*, eds. K. Giannakoglou, D. Tsahalis, J. Periaux, P. Papailou and T. Fogarty, Athens, Greece, pp. 95–100.

Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C.M., and da Fonseca, V.G. (2003), "Performance Assessment of Multiobjective Optimizers: An Analysis and Review," *IEEE Transactions on Evolutionary Computation*, 7(2), 117–132.

Four-bar mechanism configuration	θ_3	θ_4
open	$+\sqrt{}$	$-\sqrt{}$
crossed	$-\sqrt{}$	$+\sqrt{}$

Table 1. Choosing the sign of the radical according with the type of mechanism

```
Begin
    Create a random initial swarm of bacteria θ^i(j,k,l)
    ∀i, i = 1,...,S_b
    Evaluate f(θ^i(j,k,l)) ∀i, i = 1,...,S_b
    For l=1 to N_ed Do
        For k=1 to N_re Do
            For j=1 to N_c Do
                For i=1 to S_b Do
                    Update f(θ^i(j,k,l)) to emulate the swarming process
                    Perform the chemotaxis process
                    (tumble-swim) with Equations 60 and 61
                    for bacteria θ^i(j,k,l) controlled by N_s
                End For
            End For
            Perform the reproduction process by sorting
            all bacteria in the swarm based on f(θ^i(j+1,k,l)),
            deleting the S_r worst bacteria and duplicating the remaining 1 − S_r
        End For
        Perform the elimination-dispersal process by eliminating
        each bacteria θ^i(j,k,l) ∀i, i = 1,...,N_b with probability 0 ≤ P_ed ≤ 1
    End For
End
```

Table 2. BFOA pseudocode. Input parameters are number of bacteria S_b, chemotactic loop limit N_c, swim loop limit N_s, reproduction loop limit N_{re}, number of bacteria for reproduction S_r (usually $S_r = S_b/2$), elimination-dispersal loop limit N_{ed}, stepsizes C_i (they depend of the dimensionality of the problem) and probability of elimination dispersal p_{ed}.

```
Begin
    Create a random initial swarm of bacteria θ^i(j,0)
    ∀i, i = 1,...,S_b
    Evaluate f(θ^i(j,0)), g_m(θ^i(j,0)), ∀i, i = 1,...,S_b, ∀m, m = 1,...,M
    For G=1 to GMAX Do
        For i=1 to S_b Do
            For j=1 to N_c Do
                Perform the chemotaxis process
                (tumble-swim with Equations 60 and 62
                and attractor operator with Equation 64
                for bacteria θ^i(j,G) by considering the constraint-handling technique
            End For
        End For
        Perform the reproduction process by sorting
        all bacteria in the swarm based on the constraint-handling technique,
        deleting the S_r worst bacteria and duplicating the remaining 1 − S_r
        Perform the elimination-dispersal process by eliminating
        the worst bacterium θ^w(j,G) in the current swarm
        by considering the constraint-handling technique
    End For
End
```

Table 3. MBFOA pseudocode. Input parameters are number of bacteria S_b, chemotactic loop limit N_c, number of bacteria for reproduction S_r (usually $S_r = S_b/2$), scaling factor β, percentage of initial stepsize R and number of generations $GMAX$.

```
Input data: archive (A)
a=size(A) // store archive size
For All (i ∈ A ) do
  A[i]_dist=0 //initialize crowding distance for all bacteria to 0
End For
For All objective function Φ_j do
  A=sort(A,Φ_j) //Sort A in descending order according to Φ_j
  A[1]_dist = A[a]_dist = ∞ // ∞ value to bacteria in the extremes of the front for objective j
    For i=2 to a-1 do // for the remaining bacteria
      A[i]_dist = A[i]_dist + ‖ (Φ_j^{A[i-1]} - Φ_j^{A[i+1]}) / (Φ_j^{max} - Φ_j^{min}) ‖ //compute distance value for objective j
    End For
End For
```

Table 4. Crowding distance pseudocode.

```
Begin
    Archive = ∅
    Create a random initial swarm of bacteria θ^i(j,0)
    ∀i, i = 1,...,S_b
    Evaluate Φ_k(θ^i(j,0)),
    g_m(θ^i(j,0)), ∀i, i = 1,...,S_b, ∀k, k = 1,...,K, ∀m, m = 1,...,M
    For i=1 to S_b do
        If θ^i(j,0) is feasible
            Add θ^i(j,0) to the Archive by using Pareto dominance
        End If
    End For
    Compute the crowding distance for all solutions in the Archive
    For G=1 to GMAX Do
        For i=1 to S_b Do
            For j=1 to N_c Do
                Perform the chemotaxis process
                (tumble-swim with Equations 60 and 62
                and attractor operator with Equation 66 (or 64 if Archive=∅))
                for bacteria θ^i(j,G) by considering the criteria defined in Section 5.5.1
            End For
        End For
        Perform the reproduction process by duplicating the best
        bacterium in the swarm based on the criteria defined in Section 5.5.1
        and deleting the second-to-last worst bacterium
        Perform the elimination-dispersal process by eliminating
        the worst bacterium θ^w(j,G) in the current swarm
        by considering the criteria defined in Section 5.5.1
        Compute the crowding distance for all solutions in the Archive
    End For
End
```

Table 5. MOMBFOA pseudocode. Input parameters are number of bacteria S_b, chemotactic loop limit N_c, scaling factor β, percentage of initial stepsize R and number of generations $GMAX$.

Algorithm/Parameters	**Best**	**Mean**	**Median**	**St. Dev.**	**Worst**
MOMBFOA. Sb=100, GMAX=250, Nc=20, B=1.7, R=0.0018	0.2233	0.2026	0.2085	0.02419	0.176
MOMBFOA. Sb=200, GMAX=125, Nc=20, B=1.2, R=0.0018	0.1836	0.0938	0.0696	0.06	0.0284
MOMBFOA. Sb=200, GMAX=100, Nc=25, B=1.7, R=0.0018	0.2803	0.2574	0.2488	0.02	0.2431
MOMBFOA. Sb=200, GMAX=250, Nc=10, B=1.7, R=0.0018	0.1591	0.0937	0.1222	0.0832	0
MOMBFOA. Sb=250, GMAX=100, Nc=20, B=1.7, R=0.0018	0.2763	0.2699	0.2678	0.0056	0.2657
MOMBFOA. Sb=200, GMAX=125, Nc=20, B=1.7, R=0.01	0.2598	0.2373	0.2404	0.02419	0.2117
MOMBFOA, Sb=200, GMAX=125, Nc=20, B=1.7, R=0.0025	0.2471	0.2373	0.2439	0.01423	0.221
DEMD. NP=200, GMAX=2500	0.2936	0.2796	0.29	0.021	0.2554
DEMD. NP=50, GMAX=10000	0.3259	0.3086	0.3127	0.0196	0.2872
SPEA-I. Popsize=200, GMAX=2500, C=0.5, M=0.5	0.1356	0.1034	0.0922	0.02827	0.0825
SPEA-I. Popsize=200, GMAX=2500, C=1.0, M=0.5	0.1132	0.1066	0.1056	0.006	0.1012
SPEA-I. Popsize=200, GMAX=2500, C=0.5, M=1.0	0.0948	0.0881	0.0879	0.0066	0.0816
SPEA-I. Popsize=100, GMAX=5000, C=1.0, M=1.0	0.07252	0.0457	0.0562	0.0333	0.00831
NSGA-II. Popsize=100, GMAX=5000, C=0.5, M=0.3	0.0974	0.0847	0.0842	0.01245	0.0725
NSGA-II. Popsize=100, GMAX=5000, C=0.8, M=0.5	0.1236	0.1089	0.1052	0.0131	0.0981
NSGA-II. Popsize=200, GMAX=2500, C=0.8, M=0.3	0.0215	0.0175	0.0192	0.005	0.0118
MOPSO. Popsize=200, gmax=2500, ϕ_1=2.0, ϕ_2=2.0	0.0018	0.0009	0.001	0.0009	0
MOPSO. Popsize=100, gmax=5000, ϕ_1=1.8, ϕ_2=2.5	0.0023	0.0052	0.0017	0.0006	0.0012
MOPSO. Popsize=100, gmax=5000, ϕ_1=1.5, ϕ_2=1.7	0.0102	0.00748	0.0089	0.0036	0.0034
MOPSO. Popsize=100, gmax=5000, ϕ_1=2.5, ϕ_2=1.5	0.0313	0.0216	0.0212	0.0095	0.0123

Table 6. Summary of preliminary results based on the Hypervolume metric for parameter selection for each compared algorithm.

Algorithm	**Best**	**Mean**	**Median**	**St. Dev.**	**Worst**	**Mann-Whitney test**
C(MOMBFOA,NSGA-II)	1	1	1	0	1	
C(NSGA-II-MOMBFOA)	0	0	0	0	0	$\surd$
C(DEMD-NSGA-II)	1	1	1	0	1	
C(NSGA-II-DEMD)	0	0	0	0	0	$\surd$
C(MOMBFOA-DEMD)	0.9259	0.3366	0.3537	0.2571	0	
C(DEMD-MOMBFOA)	1	0.3283	0.25	0.3266	0	=
C(SPEA2-DEMD)	0	0	0	0	0	
C(DEMD-SPEA2)	1	0.3413	0.1111	0.4101	0	$\surd$
C(SPEA2-NSGA-II)	1	0.1	0	0.1005	0	
C(NSGA-II-SPEA2)	1	0.1917	0	0.3263	0	=
C(SPEA2-MOMBFOA)	0.125	0.0033	0	0.0193	0	
C(MOMBFOA-SPEA2)	1	0.3072	0.0909	0.4035	0	$\surd$
C(MOMBFOA-MOPSO)	0.5455	0.4191	0.4978	0.1792	0.2141	
C(MOPSO-MOMBFOA)	0	0	0	0	0	$\surd$
C(DEMD-MOPSO)	0.5455	0.4237	0.4912	0.1661	0.2345	
C(MOPSO-DEMD)	0	0	0	0	0	$\surd$
C(SPEA2-MOPSO)	0.5455	0.3256	0.4312	0.2877	0	
C(MOPSO-SPEA2)	0.0714	0.0238	0	0.0412	0	$\surd$
C(NSGA-II-MOPSO)	0	0	0	0	0	
C(MOPSO-NSGA-II)	0.5455	0.3898	0.3895	0.1555	0.2345	$\surd$

Table 7. Summary of results for pairwise comparisons among the four algorithms based on the C-metric. "$\surd$" indicates a significant difference based on the Mann-Whitney test with 95% confidence. "=" means no significant difference based on the aforementioned test.

Algorithm	**Best**	**Mean**	**Median**	**St. Dev.**	**Worst**	**Mann-Whitney test**
DEMD	**0.3548**	**0.3052**	**0.3013**	0.0266	**0.2731**	$\surd$
NSGA-II	0.1478	0.1070	0.1077	0.0311	0.0478	$\surd$
SPEA2	0.2252	0.1068	0.1718	0.0896	0	$\surd$
MOPSO	0.0778	0.02593	0	0.0449	0	$\surd$
MOMBFOA	0.2890	0.2698	0.2750	**0.0164**	0.2465	$\surd$

Table 8. Summary of results for the five algorithms based on the Hypervolume metric. "$\surd$" indicates that the differences observed in the statistical values reported in this table are significant, based on the Mann-Whitney test with 95% confidence, with respect to the other three algorithms.

Algorithm	**Best**	**Mean**	**Median**	**St. Dev.**	**Worst**	**Mann-Whitney test**
DEMD	**0.000850**	**0.007123**	**0.006300**	0.005317	**0.014500**	$\surd$ (except with MOMBFOA)
NSGA-II	0.120000	0.035200	0.017150	0.054988	0.191000	$\surd$ (except with SPEA2)
SPEA2	0.039000	0.103078	0.068600	0.086284	0.313000	$\surd$ (except with NSGA-II)
MOPSO	0.029000	1.098743	0.0488	1.903072	3.296222	$\surd$
MOMBFOA	0.002200	0.010120	0.010200	**0.004933**	0.021100	$\surd$ (except with DEMD)

Table 9. Summary of results for the four algorithms based on the Generational Distance metric. "$\surd$" indicates that the differences observed in the statistical values reported in this table are significant, based on the Mann-Whitney test with 95% confidence, with respect to the other three algorithms.

Algorithm	x_1/m	x_2/m	x_3/m	x_4/m	x_5/rad
DEMD	0.4439	0.0505	0.4464	0.1406	-0.1295
NSGA-II	0.3213	0.0745	0.3220	0.2054	-0.3013
SPEA2	0.4041	0.1246	0.386	0.3033	-0.4164
MOPSO**	0.4741	0.1107	0.4704	0.2947	-0.3234
MOMBFOA	0.4825	0.0631	0.4825	0.1543	-0.1467

Table 10. Details of representative solutions by each compared algorithm.** MOPSO solution is sligthly infeasible.

Figure captions.

Figure 1: Traditional continuously variable transmission.

Figure 2: Continuously variable transmission proposed in this work.

Figure 3: Four-bar mechanism diagram.

Figure 4: Tightening points in a four-bar mechanism.

Figure 5: Maximum and minimum transmission angles.

Figure 6: Average number of solutions per generation on 10 independent runs which satisfy the dynamic constraint.

Figure 7: Average number of feasible solutions per generation on 10 independent runs.

Figure 8: Average number of feasible non-dominated solutions per generation in the external file on 10 independent runs.

Figure 9: Filtered Pareto sub-optimal fronts obtained from 10 independent runs for MOMB-FOA, DEMD, NSGA-II, MOPSO, and SPEA2.

Figure 10: Average hypervolume value versus number of evaluations per each compared algorithm.

Figure 11: Region where more feasible non-dominated solutions were generated by the compared algorithms.

Figure 12: Simulator output of the angular displacement of the rocker θ_4 and transmission angle μ for each feasible representative solution in Table 9.

Efrén Mezura-Montes is a full-time researcher at the National Laboratory for Advanced Informatics (LANIA) A.C. in Xalapa, Veracruz, MÉXICO. His research interests are the design, analysis and application of nature-inspired algorithms to solve complex optimization problems. He has published over 70 papers in peer-reviewed journals and conferences. He also has one edited book published and nine book chapters published by international publishing companies.

Edgar A. Portilla-Flores is a full time researcher at the Centro de Innovación y Desarrollo Tecnológico en Cómputo del Instituto Politécnico Nacional (CIDETEC-IPN) in México City, MÉXICO. His research interests are: integrated design of mechatronic systems, design of robotic grippers and mechanical systems of power transmission.

Betania Hernández-Ocaña graduated with honors from the MsC on Applied Computing at LANIA A.C., and is currently a PhD student at the Universidad Juárez Autónoma de Tabasco, MÉXICO. Her research interests are nature-inspired algorithms for optimization.