Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2012, Article ID 151590, 20 pages
doi:10.1155/2012/151590

Research Article

Hybrid Optimization Approach for the Design of Mechanisms Using a New Error Estimator

A. Sedano,[1] R. Sancibrian,[2] A. de Juan,[2] F. Viadero,[2] and F. Egaña[3]

[1] *Calculation Department, MTOI, C/Maria Viscarret 1, Ártica (Berrioplano), 31013 Navarra, Spain*
[2] *Department of Structual and Mechanical Engineering, University of Cantabria, Avenida de los Castros s/n, 39005 Santander, Spain*
[3] *Mechatronics Department, Tekniker, Avenida Otaola 20, 20600 Eibar, Spain*

Correspondence should be addressed to R. Sancibrian, sancibrr@unican.es

Received 24 February 2012; Revised 24 April 2012; Accepted 14 May 2012

Academic Editor: Yi-Chung Hu

A hybrid optimization approach for the design of linkages is presented. The method is applied to the dimensional synthesis of mechanism and combines the merits of both stochastic and deterministic optimization. The stochastic optimization approach is based on a real-valued evolutionary algorithm (EA) and is used for extensive exploration of the design variable space when searching for the best linkage. The deterministic approach uses a local optimization technique to improve the efficiency by reducing the high CPU time that EA techniques require in this kind of applications. To that end, the deterministic approach is implemented in the evolutionary algorithm in two stages. The first stage is the fitness evaluation where the deterministic approach is used to obtain an effective new error estimator. In the second stage the deterministic approach refines the solution provided by the evolutionary part of the algorithm. The new error estimator enables the evaluation of the different individuals in each generation, avoiding the removal of well-adapted linkages that other methods would not detect. The efficiency, robustness, and accuracy of the proposed method are tested for the design of a mechanism in two examples.

1. Introduction

The mechanical design of modern machines is often very complex and needs very sophisticated tools to meet technological requirements. The design of linkages is no exception, and modern applications in this field have increasingly demanding requirements. The design of linkages consists in obtaining the best mechanism to fulfil a specific motion characteristic demanded by a specific task. In many engineering design fields there are three common requirements known as function generation, path generation, or rigid-body guidance [1, 2]. Dimensional synthesis deals with the determination of the kinematic parameters of the mechanism necessary to satisfy the required motion characteristics.

Different techniques have been used for the synthesis of mechanisms including graphical and analytical techniques [1–3]. Graphical and analytical methods developed in the literature are relatively restricted because they find the exact solution for a reduced number of prescribed poses and variables. However, during the last decades numerical methods have enabled an increase in the complexity of the problems by using numerical optimization techniques [4–8].

Despite the work done in dimensional synthesis over recent decades, the design of mechanisms is still a task where the intuition and experience of the engineers play an important role. One of the main reasons for this is the large number of variables involved in a strongly nonlinear problem. Under these circumstances the design variable space contains too many local minima and only some of them can be identified as local solutions. These local solutions provide an error below a limit established by the designer and can be considered acceptable. However, only the global minimum leads to the solution that provides the greatest accuracy and this should be the main objective in the design of mechanisms.

The application of local optimization techniques to the synthesis of mechanisms took place mainly during the 80s and 90s. Although other techniques have become more important in recent years, they remain important so far. Some local search methods have been described in references [4–6]. The main disadvantage of these methods is their dependence on the initial point, or initial guess, although they also require a differentiable objective function. Several research works have been done to achieve exact differentiation, which improve the accuracy and efficiency of these methods. For example, in [5] exact gradient is determined to optimize a six-bar and eight-bar mechanism. In [6] a general synthesis procedure is obtained by using exact differentiation and it is applied to different kinds of problems. However, the dependence on the initial point cannot be avoided and therein lies the weakest point of local search methods.

Global search methods avoid the dependence on the initial point, but there is a sharp increase in the computational time necessary to achieve convergence. Genetic algorithms (GAs) [7, 8], evolutionary algorithms (EAs) [9], and Particle Swarm (PS) are some of the most frequently used optimization techniques in the literature. All these techniques mimic the behaviour of processes found in nature and are based on biological processes.

Genetic and evolutionary algorithms apply the principles of evolution found in nature to the problem of finding an optimal solution. Holland [10] was the first to introduce the GA and DeJong [11] verified the usage. In GA the genes are usually encoded using a binary language whereas in EA the decision variables and objective function are used directly. As coding is not necessary, EAs are less complex and easier to implement for solving complicated optimization problems. Cabrera et al. [8] used GAs applied to a four-bar linkage in a path generation problem. Some years later Cabrera et al. [9] used EAs to solve more complex problems in the design of mechanisms. In this case a multiobjective problem is formulated including mechanical advantage in the objective function as a design requirement. In [12] a genetic algorithm is used for the Pareto optimum synthesis of a four-bar linkage considering the minimization of two objective functions simultaneously.

Hybrid algorithms with application to the synthesis of linkages have been studied in recent years. Lin [13] developed an evolutionary algorithm by combining differential evolution and the real-valued genetic algorithm. Khorshidi [14] developed a hybrid approach where a local search is employed to accelerate the convergence of the algorithm. However, these methods are limited to the four-bar mechanism and their application is restricted to path generation problems.

The objective function is based on the synthesis error estimation. The most widely used error estimator in the literature is Mean Square Distance (MSD). The MSD is used to measure the difference between the desired parameters and the generated ones. However, this formulation has proven to be ineffective when seeking the optimal solution [15]. In many cases the MSD misleads the direction of the design and good linkages generated by the algorithm can be underestimated. Therefore, error estimation is of the utmost importance for deterministic and stochastic optimization. In EA the error estimator must be evaluated for each individual in each generation and for this reason the lack of accuracy could lead to poor efficiency in the optimization process. To avoid these problems Ullah and Kota [15] proposed the use of Fourier Descriptors that evaluate only the shape differences between the generated and desired paths. However, the proposed formulation is limited to closed paths in path generation problems. An energy-based error function is used in [16] where the finite element method is used to assess the synthesis error. This formulation reduces the drawbacks of MSD, but problems with the relative distance between desired and generated parameters remain.

The aim of this work is to propose a new hybrid algorithm that combines an evolutionary technique with a local search optimization approach. Some of the fundamentals in mechanism synthesis studied in this paper have been extensively discussed in the literature. However, the originality of this work lies in two aspects: the first one is the introduction of a new error estimator which accurately compares the function generated by the candidate mechanism with desired function. The second one is a novel approach based on the combination of deterministic and stochastic optimization techniques in the so-called hybrid methods. The flowchart for the optimization process is presented in the paper together with the results and conclusions.

2. Objective Function and Deterministic Optimization Approach

In optimal synthesis of linkages the optimization problem is defined as follows:

$$\begin{aligned} &\text{minimize } F[\mathbf{q}(\mathbf{w}), \mathbf{w}] \\ &\text{subject to } \mathbf{\Phi}[\mathbf{q}(\mathbf{w}), \mathbf{w}] = 0, \\ &\qquad\qquad \mathbf{g}[\mathbf{q}(\mathbf{w}), \mathbf{w}] \leq 0, \end{aligned} \tag{2.1}$$

where the objective function $F[\mathbf{q}(\mathbf{w}), \mathbf{w}]$ formulates the technological requirements of the mechanism to be designed. The equality constraints $\mathbf{\Phi}[\mathbf{q}(\mathbf{w}), \mathbf{w}]$ formulate the kinematic restrictions during the motion, and the inequality constraints $\mathbf{g}[\mathbf{q}(\mathbf{w}), \mathbf{w}]$ establish the limitations in the geometrical dimensions. Vector $\mathbf{q}(\mathbf{w})$ is the vector of dependent coordinates and $\mathbf{w}$ is the n-dimensional vector of design variables.

To illustrate the formulation the scheme of a four-bar mechanism in a path generation synthesis problem is shown in Figure 1. The proposed method can be applied effortlessly to any type of planar mechanism; however, the example in Figure 1 enables the formulation to be easily understood. The equality constraints are formulated as follows:

$$\mathbf{\Phi}[\mathbf{q}(\mathbf{w}), \mathbf{w}]_i = \left\{ \begin{array}{c} L_1 \cos\theta_1 + L_2 \cos(\theta_{20} + \theta_{2i}) + L_3 \cos\theta_3 + L_3 \cos\theta_3 \\ L_1 \sin\theta_1 + L_2 \sin(\theta_{20} + \theta_{2i}) + L_3 \sin\theta_3 + L_3 \sin\theta_3 \\ x_g - x_0 - L_1 \cos\theta_1 - L_2 \cos(\theta_{20} + \theta_{2i}) - L_5 \cos(\theta_5 + \alpha) \\ y_g - y_0 - L_1 \sin\theta_1 - L_2 \sin(\theta_{20} + \theta_{2i}) - L_5 \sin(\theta_5 + \alpha) \end{array} \right\} = \mathbf{0}, \tag{2.2}$$

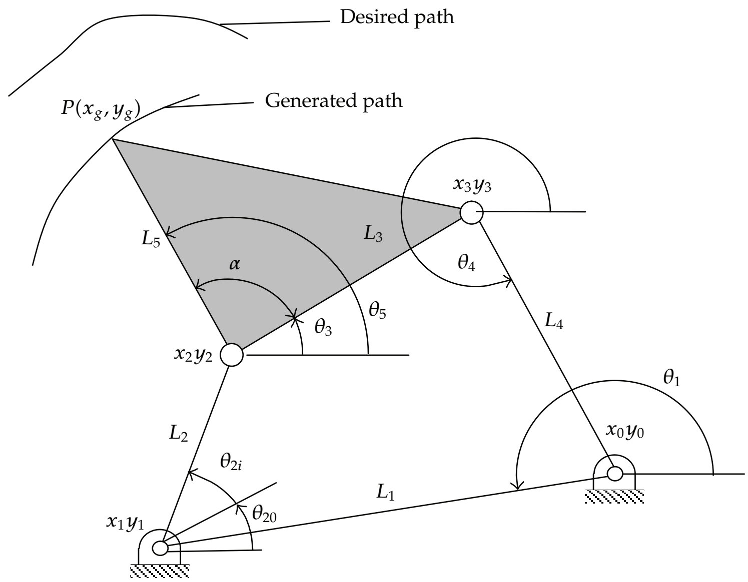


Figure 1: Scheme of the four-bar linkage.

where the vector of design variables contains the geometrical dimensions of the link. That is

$$\mathbf{w}^T = \begin{bmatrix} x_0 & y_0 & \theta_1 & \theta_{20} & L_1 & L_2 & L_3 & L_4 & L_5 & \alpha \end{bmatrix}, \tag{2.3}$$

and dependent variables are defined as

$$\mathbf{q}^T = \begin{bmatrix} \theta_3 & \theta_4 & x_g & y_g \end{bmatrix}. \tag{2.4}$$

An important aspect in dimensional synthesis of linkages is the formulation of the goal or objective function. The objective function is capable of expressing the difference between the desired and generated paths (see Figure 1), providing an estimation of the error between the two curves irrespective of location, orientation, and size. The minimization of this function obliges the design variables to be changed and leads to the optimal dimensions of the linkage which can be expressed as $\mathbf{w}^*$. The generated and desired paths can be either open or closed curves. Figure 2 shows the two paths for the case of two closed curves. In this work the definition of both curves is assumed to be specified by a number of points named precision points. The precision points are selected by the designer by using vector notation and Cartesian coordinates as follows:

$$\begin{aligned} \mathbf{d}^{iT} &= \begin{bmatrix} x_d^i & y_d^i \end{bmatrix} \\ \mathbf{g}^{iT} &= \begin{bmatrix} x_g^i & y_g^i \end{bmatrix} \end{aligned} \quad i = 1, 2, \ldots, p, \tag{2.5}$$

where subscript d stands for desired points, g stands for the generated ones and p is the number of precision points. Most of the works in dimensional synthesis propose the Mean Square Distance to assess the error between the two curves. That is

$$F = \frac{1}{2}\sum_{i=1}^{p}\left[\left(\mathbf{g}^i - \mathbf{d}^i\right)^T\left(\mathbf{g}^i - \mathbf{d}^i\right)\right] = \frac{1}{2}\left[(\mathbf{g} - \mathbf{d})^T(\mathbf{g} - \mathbf{d})\right]. \tag{2.6}$$

The deterministic approach used in this paper is based on a local search procedure which uses first-order differentiation to obtain the search direction. In the synthesis problem the generated precision points depend on the vector of design variables and (2.6) should be rewritten as follows:

$$F[\mathbf{q}(\mathbf{w}), \mathbf{w}] = \frac{1}{2}[\mathbf{g}\{\mathbf{q}(\mathbf{w}), \mathbf{w}\} - \mathbf{d}]^T[\mathbf{g}\{\mathbf{q}(\mathbf{w}), \mathbf{w}\} - \mathbf{d}] \tag{2.7}$$

which is the objective function that must be minimized subject to the equality and inequality constraints to obtain the optimal dimensions of the mechanism. Differentiating (2.7) with respect to the design variables and equating it to zero provides

$$\nabla F[\mathbf{q}(\mathbf{w}), \mathbf{w}] = \mathbf{J}^T[\mathbf{g}\{\mathbf{q}(\mathbf{w}), \mathbf{w}\} - \mathbf{d}] = \mathbf{0}, \tag{2.8}$$

where $\mathbf{J}$ is the Jacobian that can be expressed as,

$$\mathbf{J} = \frac{\partial \mathbf{g}[\mathbf{q}(\mathbf{w}), \mathbf{w}]}{\partial \mathbf{w}} = \frac{\partial \mathbf{g}[\mathbf{q}(\mathbf{w}), \mathbf{w}]}{\partial \mathbf{q}(\mathbf{w})}\frac{\partial \mathbf{q}(\mathbf{w})}{\partial \mathbf{w}}. \tag{2.9}$$

With the aim of greater clarity hereafter the dependence on the variables is omitted. The term between brackets in (2.8) can be expanded using Taylor series expansion as

$$\nabla \mathbf{F} \approx \mathbf{J}^T\mathbf{g} - \mathbf{J}^T\mathbf{d} + \mathbf{J}^T\mathbf{J}\Delta\mathbf{w} = \mathbf{0}. \tag{2.10}$$

From (2.10) a recursive formula can be obtained as follows:

$$\mathbf{w}_{j+1} = \mathbf{w}_j - \alpha\ \ \mathbf{J}^T\mathbf{J}\left[\mathbf{J}^T\mathbf{g} - \mathbf{J}^T\mathbf{d}\right]. \tag{2.11}$$

In this formula the stepsize α has been included in order to control the distance along the search direction. In [6] the determination of the stepsize and the exact Jacobian is described.

Differentiation of equality constraints given in (2.1) using the chain rule yields

$$\frac{\partial \boldsymbol{\Phi}}{\partial \mathbf{q}}\frac{\partial \mathbf{q}}{\partial \mathbf{w}} + \frac{\partial \boldsymbol{\Phi}}{\partial \mathbf{w}} = \mathbf{0}. \tag{2.12}$$

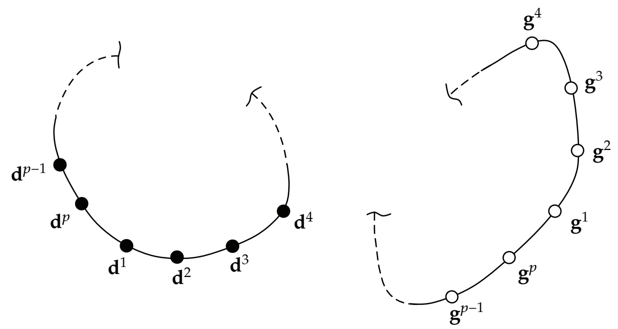


Figure 2: Desired ($\mathbf{d}$) and generated ($\mathbf{g}$) closed paths.

Thus, (2.9) can be rewritten as

$$\mathbf{J} = -\frac{\partial \mathbf{g}}{\partial \mathbf{q}} \left(\frac{\partial \mathbf{\Phi}}{\partial \mathbf{q}} \right)^{-1} \frac{\partial \mathbf{\Phi}}{\partial \mathbf{w}}. \tag{2.13}$$

All terms in (2.13) can be easily obtained from the objective function and constraints, and they enable the exact Jacobian to be determined for use in the deterministic optimization method.

If there are inequality constraints in the optimization problem, they can be converted to equality constraints through the addition of so-called slack variables. That is

$$g_i(\mathbf{w}) + v_i^2 = 0. \tag{2.14}$$

In this way, each inequality constraint adds a new variable that must be included in the formulation.

3. New Error Estimator for EA Algorithms

In EA, lack of accuracy in the error estimation could lead to overestimation of the error and removal of good individuals from the optimization process. On the other hand, underestimation of the error could lead to selecting individuals who are not better adapted than others in fulfilling the goal.

Equation (2.7) is used in many works as the objective function [4–9]. It has been widely used in deterministic approaches, but it is also used in probabilistic optimization. However, the function itself is an estimator of the error, not a representation of the actual error. This function depends on the relative position of the two curves and under certain circumstances the approximation may not be good enough. For instance, in the case shown in Figure 2 the error given by (2.7) can be increased or decreased if the generated curve is translated closer to the desired one or away from it, respectively. Moreover, rotation and scaling can be added to the transformation in order to reduce the error. For practical applications in engineering the translation of the curve only entails the translation of the linkage even as the rotation only needs to change the mechanism orientation. The lack of accuracy of (2.7) can be reduced by selecting the appropriate initial guess linkage in local optimization. However, in EA this option is not available and it should be solved using other strategies.

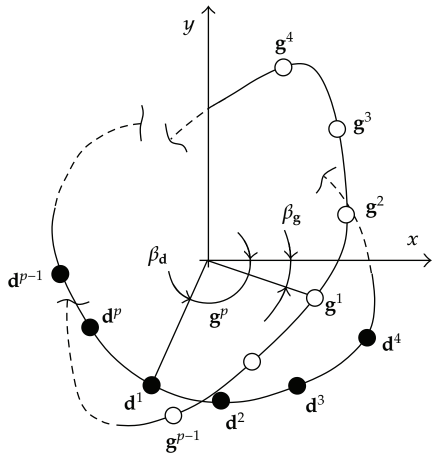


Figure 3: Translation of the desired (**d**) and generated (**g**) paths.

Thus, one can say that the error between two curves is minimum if they are compared by the translation, rotation, and scaling. Therefore, (2.7) could underestimate the error unless some transformations are introduced. The first transformation consists of the translation of the generated curve towards the desired one. To do that, the geometric centroids of both curves are determined by using the precision points as follows:

$$\mathbf{d}_c = \frac{1}{p}\sum_{i=1}^{p}\mathbf{d}^i,$$
$$\mathbf{g}_c = \frac{1}{p}\sum_{i=1}^{p}\mathbf{g}^i, \tag{3.1}$$

where $\mathbf{d}_c$ and $\mathbf{g}_c$ are the coordinates of the geometric centroids for desired and generated curves, respectively. The new coordinates of the precision points for the two paths are obtained by translating the geometric centroids to the origin of the reference frame. That is

$$\mathbf{d}_0^i = \mathbf{d}^i - \mathbf{d}_c,$$
$$\mathbf{g}_0^i = \mathbf{g}^i - \mathbf{g}_c. \tag{3.2}$$

Figure 3 shows the translation of both curves. Thus, the error estimation can be reformulated in the following way:

$$E_0 = \frac{1}{2}\sum_{i=1}^{p}\left[\left(\mathbf{g}_0^i - \mathbf{d}_0^i\right)^T\left(\mathbf{g}_0^i - \mathbf{d}_0^i\right)\right]. \tag{3.3}$$

Obviously, (3.3) reduces the error and is more accurate than (2.7). However, it should be pointed out that it still depends on the order chosen for numbering the precision points. In other words, (3.3) provides a comparison of the precision points with the same superscript, which depends on the arbitrary choice previously made by the designer. Therefore, it could be possible to reduce the error when the order of numbering is changed. Therefore, removing the effect of the numbering requires the formulation of p error estimators. For the case of two closed curves, as shown in Figure 3, the following matrix can be written:

$$\begin{bmatrix} \mathbf{g}_0^1-\mathbf{d}_0^1 & \mathbf{g}^2-\mathbf{d}_0^1 & \cdots & \mathbf{g}^{p-1}-\mathbf{d}_0^1 & \mathbf{g}_0^p-\mathbf{d}_0^1 \\ \mathbf{g}_0^2-\mathbf{d}_0^2 & \mathbf{g}_0^3-\mathbf{d}_0^2 & \cdots & \mathbf{g}_0^p-\mathbf{d}_0^2 & \mathbf{g}_0^1-\mathbf{d}_0^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{g}_0^{p-1}-\mathbf{d}_0^{p-1} & \mathbf{g}_0^p-\mathbf{d}_0^{p-1} & \cdots & \mathbf{g}_0^{p-3}-\mathbf{d}_0^{p-1} & \mathbf{g}_0^{p-2}-\mathbf{d}_0^{p-1} \\ \mathbf{g}_0^p-\mathbf{d}_0^p & \mathbf{g}_0^1-\mathbf{d}_0^p & \cdots & \mathbf{g}_0^{p-2}-\mathbf{d}_0^p & \mathbf{g}_0^{p-1}-\mathbf{d}_0^p \end{bmatrix}_{p\times p}. \tag{3.4}$$

Each column of (3.4) gives the terms of the error estimator for each possible combination. Thus, each error estimator can be formulated as the summatory function defined by

$$F_j = \sum_{i=1}^{p-j+1}\left(\mathbf{g}_0^{i+j-1}-\mathbf{d}_0^i\right)^2 + \sum_{i=p-j+2}^{p}\left(\mathbf{g}_0^{i+j-p-1}-\mathbf{d}_0^i\right)^2; \quad j=1,2,\ldots,p, \tag{3.5}$$

where subscript j stands for the estimator index. Therefore, F_j is a single-valued function providing the estimation of the error. A vector can be formulated with all the values given by (3.5)

$$\mathbf{F}^T = \left[F_1F_2\cdots F_p\right]. \tag{3.6}$$

Only one of the terms in this vector provides the minimum error and will be selected to form the objective function. That is

$$F_m = \min(\mathbf{F}). \tag{3.7}$$

The matrix given by (3.4) and the summatory given by (3.5) are only valid for the comparison of closed-closed curves. However, it is possible to have two other situations: open-open or open-closed paths. The former case is shown in Figure 4, while the latter is shown in Figure 5. In both cases the number of precision points may be different for the desired and generated curves. Thus, the precision points are redefined as follows:

$$\begin{aligned} \mathbf{d}^{iT} &= \left[x_d^i \;\; y_d^i\right]; \;\; i=1,2,\ldots,p \\ \mathbf{g}^{rT} &= \left[x_g^r \;\; y_g^r\right]; \;\; r=1,2,\ldots,c \end{aligned} \quad c \geq p, \tag{3.8}$$

where c is the number of precision points for the generated curve. Similarly to the closed-closed case, the centroid of the precision points is determined and the curves are translated

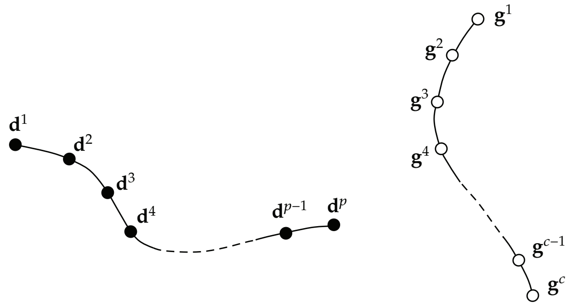


Figure 4: Desired (**d**) and generated (**g**) open paths.

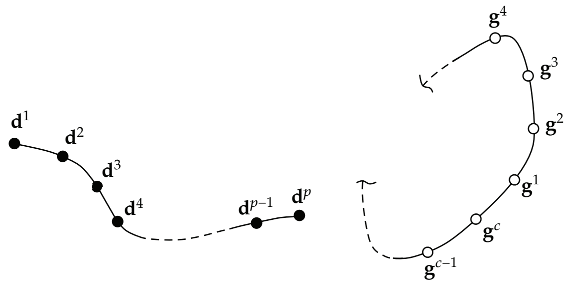


Figure 5: Desired open path (**d**) and generated closed path (**g**).

to the origin of the reference frame. The possible combinations that allow the estimation of the error are given by the following matrix:

$$\mathbf{E} = \begin{bmatrix} \mathbf{g}_0^1 - \mathbf{d}_0^1 & \mathbf{g}_0^2 - \mathbf{d}_0^1 & \cdots & \mathbf{g}_0^{c-p} - \mathbf{d}_0^1 & \mathbf{g}_0^{c-p+1} - \mathbf{d}_0^1 \\ \mathbf{g}_0^2 - \mathbf{d}_0^2 & \mathbf{g}_0^3 - \mathbf{d}_0^2 & \cdots & \mathbf{g}_0^{c-p+1} - \mathbf{d}_0^2 & \mathbf{g}_0^{c-p+2} - \mathbf{d}_0^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{g}_0^{p-1} - \mathbf{d}_0^{p-1} & \mathbf{g}_0^{p} - \mathbf{d}_0^{p-1} & \cdots & \mathbf{g}_0^{c-2} - \mathbf{d}_0^{p-1} & \mathbf{g}_0^{c-1} - \mathbf{d}_0^{p-1} \\ \mathbf{g}_0^{p} - \mathbf{d}_0^{p} & \mathbf{g}_0^{p+1} - \mathbf{d}_0^{p} & \cdots & \mathbf{g}_0^{c-1} - \mathbf{d}_0^{p} & \mathbf{g}_0^{c} - \mathbf{d}_0^{p} \end{bmatrix}_{p\times(c-p+1)}. \tag{3.9}$$

The sum of the squared elements of each column in (3.9) leads to

$$F_j = \sum_{i=1}^{p} \left(\mathbf{g}_0^{i+j-1} - \mathbf{d}_0^i\right)^2; \quad j = 1, 2, \ldots, c - p + 1. \tag{3.10}$$

Equation (3.10) provides the different error estimators and the minimum value given by this formula is selected as the objective function.

When the desired path is an open curve and the generated path is a closed curve (see Figure 5), the aforementioned process can be used. However, the error estimator should be adapted to this situation. That is

$$\mathbf{E} = \begin{bmatrix} \mathbf{g}_0^1 - \mathbf{d}_0^1 & \mathbf{g}_0^2 - \mathbf{d}_0^1 & \cdots & \mathbf{g}_0^{c-1} - \mathbf{d}_0^1 & \mathbf{g}_0^c - \mathbf{d}_0^1 \\ \mathbf{g}_0^2 - \mathbf{d}_0^2 & \mathbf{g}_0^3 - \mathbf{d}_0^2 & \cdots & \mathbf{g}_0^c - \mathbf{d}_0^2 & \mathbf{g}_0^1 - \mathbf{d}_0^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{g}_0^{p-1} - \mathbf{d}_0^{p-1} & \mathbf{g}_0^p - \mathbf{d}_0^{p-1} & \cdots & \mathbf{g}_0^{p-3} - \mathbf{d}_0^{p-1} & \mathbf{g}_0^{p-2} - \mathbf{d}_0^{p-1} \\ \mathbf{g}_0^p - \mathbf{d}_0^p & \mathbf{g}_0^{p+1} - \mathbf{d}_0^p & \cdots & \mathbf{g}_0^{p-2} - \mathbf{d}_0^p & \mathbf{g}_0^{p-1} - \mathbf{d}_0^p \end{bmatrix}_{p\times c}. \tag{3.11}$$

Thus, the error estimators can be formulated as follows:

$$\begin{gathered} F_j = \sum_{i=1}^{p} \left(\mathbf{g}_0^{i+j-1} - \mathbf{d}_0^i\right)^2, \quad 1 \le j \le c-p+1, \\ F_j = \sum_{i=1}^{c-j+1} \left(\mathbf{g}_0^{i+j-1} - \mathbf{d}_0^i\right)^2 + \sum_{i=c-j+2}^{p} \left(\mathbf{g}_0^{i+j-c-1} - \mathbf{d}_0^i\right)^2, \quad c-p+1 \le j \le c. \end{gathered} \tag{3.12}$$

Equations (3.5), (3.10), and (3.12) provide a better comparison because they remove the effect of the translation and avoid the influence of the numbering. However, the error estimation can be enhanced by rotation and scaling. Indeed, if the generated curve is rotated and scaled with respect to the desired one, the difference between the two curves could be reduced. To do this, two new parameters must be introduced. The first one is a reference angle which provides the orientation of each curve. In Figure 2 the orientation angles are given by β_d and β_g. In practical design of mechanisms the modification of β_g implies the rotation of the whole linkage in the plane, which is allowed for most of the cases. The second parameter is the scaling factor s. This parameter allows the generated curve to be expanded or contracted to reduce the difference with respect to the desired path. For the case of closed-closed curves the introduction of the rotation and scaling factor in the formulation modifies equations as follows:

$$F_m(\beta_g, s) = \sum_{i=1}^{p-j+1} \left[s\mathbf{A}\mathbf{g}_0^{i+j-1} - \mathbf{d}_0^i\right]^2 + \sum_{i=p-j+2}^{p} \left(s\mathbf{A}\mathbf{g}_0^{i+j-p-1} - \mathbf{d}_0^i\right)^2; \quad j = 1, 2, \ldots, p, \tag{3.13}$$

where

$$\mathbf{A}(\beta_g) = \mathbf{A} = \begin{bmatrix} \cos\beta_g & -\sin\beta_g \\ \sin\beta_g & \cos\beta_g \end{bmatrix}, \tag{3.14}$$

is the rotation matrix and provides the rotation of the generated precision points.

The error estimator given by (3.13) is now the objective function in a local optimization subproblem with two variables, β_g and s. This optimization subproblem attempts to find the best orientation and size of the generated curve (and also the linkage) in order to reduce the error with respect to the desired path. The objective functions for the open-open curves may be readily derived as

$$F_m(\beta_g, s) = \sum_{i=1}^{p}\left(s\mathbf{A}\mathbf{g}_0^{i+j-1} - \mathbf{d}_0^i\right)^2; \quad j = 1, 2, \ldots, c - p + 1, \tag{3.15}$$

and (3.12) for the open-closed curves becomes

$$\begin{aligned} F_m(\beta_g, s) &= \sum_{i=1}^{p}\left(s\mathbf{A}\mathbf{g}_0^{i+j-1} - \mathbf{d}_0^i\right)^2, \quad 1 \le j \le c - p + 1, \\ F_m(\beta_g, s) &= \sum_{i=1}^{c-j+1}\left(s\mathbf{A}\mathbf{g}_0^{i+j-1} - \mathbf{d}_0^i\right)^2 + \sum_{i=c-j+2}^{p}\left(s\mathbf{A}\mathbf{g}_0^{i+j-c-1} - \mathbf{d}_0^i\right)^2, \quad c - p + 1 \le j \le c. \end{aligned} \tag{3.16}$$

These expressions might suggest that the problem must be solved as an optimization with two variables. However, the authors' experience shows that better results are obtained when the problem is solved independently for each variable. In other words, the results obtained are very accurate when the rotation optimization problem is solved before the scaling problem.

In summary, the aforementioned transformations are the core of the comparison between the desired curve and the candidate, avoiding the influence of location, orientation, and size all at once. This provides an important contribution that improves the efficiency in the exploration of the search space when using evolutionary algorithms.

4. Hybrid Approach for the Synthesis of Mechanisms

The design space of linkages contains a large number of local minima. Deterministic approaches based on local optimization start from a random point converging to the nearest local minimum. Thus, the solution may be an unsatisfactory solution because the design space is not sufficiently explored. The strength of stochastic optimization approaches lies in searching the entire design space of the design variables in order to locate a region with the lowest values of the objective function. This region probably contains the global minimum. However, the cost of the computational time required to achieve the convergence by using EA could be very expensive when an accurate solution is demanded. Local search approaches need less time to achieve solutions, but the accuracy depends on the quality of the initial guess. To ensure convergence and enhance its ratio hybrid methods combine the benefits of both techniques. The main advantages expected from this approach are the generality and total independence of the initial guess. The evolutionary process for searching among the optima is briefly outlined below.

4.1. Evolutionary Strategy

It should be highlighted that the efficiency of an evolutionary algorithm is given by both the quality of the objective function and the structure of the chromosomes and their genes. In this work the objective function is formulated as was described in the previous section. The chromosomes are encoded using real-valued genes instead of a binary code because several works [9, 13] have demonstrated the advantages of this procedure in the design of linkages. Thus, each gene gives the real value of a design variable in the mechanism to be synthesized and all genes are grouped in a chromosome which in classical optimization is known as the vector of design variables. That is

$$\mathbf{w}_{r,g}^{T} = \left[w_{1,g} \;\; w_{2,g} \;\; \cdots \;\; w_{m,g} \right]; \quad r = 1, 2, \ldots, r_{\max}, \tag{4.1}$$

where m represents the dimensionality of $\mathbf{w}_{r,g}$, g is the generation subscript, and $r_{\max}$ is the number of individuals in each generation. The dimension of $\mathbf{w}$ is given by the type of mechanism to be synthesized and the kind of coordinates used in their definition. In this work natural coordinates are used for this purpose, as well as in the definition of the generated and desired paths. The starting and successive populations are randomly generated:

$$\mathbf{P}_g = \mathbf{w}_{r,g}; \quad r = 1, 2, \ldots, r_{\max}; \; g = 1, 2, \ldots, g_{\max}, \tag{4.2}$$

where $g_{\max}$ is the number of generations. In this work $r_{\max}$ does not change during the optimization process so the population neither increases nor decreases. After a generation has been created, the fitness of each individual is evaluated in order to sort them for the selection. The evaluation of the fitness depends on the type of curves involved in the problem, selecting (3.13), (3.15) or (3.16) according to the case. The algorithm uses an elitism strategy in order to preserve the best individuals for the next generation. To obtain the number of best individuals an elitism factor, ef, is used as follows:

$$nE = \text{Round } (\text{ef}\, r_{\max}), \tag{4.3}$$

where nE is the number of individuals whose genetic information is preserved for the following generation. After that, the tournament selection starts and the parents are chosen for reproduction. The first step in reproduction is to establish the number of offspring generated by the crossover, whose valued is given by the following formula:

$$nC = \text{Round } [\text{rf}(r_{\max} - nE)], \tag{4.4}$$

where nC is the number of offspring generated by the crossover operator and rf is the reproduction factor. Mutation is another operator used to change the genes randomly during the reproduction. The number of offspring affected by mutation is given by

$$nM = r_{\max} - nE - nC. \tag{4.5}$$

Thus, the number of parents is twice the number of offspring selected for crossover plus the number of individuals selected for mutation. To decide whether or not it should become a member for reproduction, the roulette wheel method [8] is used for the selection of parents from the complete population. The number of slots in the roulette is equal to the number of individuals and the size of the slots is equal to their expectation. Once the parents are selected, crossover is used to increase the diversity of the individuals in the complete population.

Crossover generates the offspring by taking genetic information from the two parents. The chromosomes of the descendents are obtained using the arithmetic mean of the same genes taken from each parent using a random coefficient with normal distribution. The mutation operator is controlled by two coefficients. The first one is the scale, sM, which controls the range of the variation allowed in the genes. The second one is the shrink coefficient, hM. This coefficient allows mutations with a wider interval of variation in the first generations but gradually reduces this interval in the following generations. In this way the algorithm provides exploration and exploitation of the global optimum and maintains a suitable balance during the optimization process. The authors have verified that the control of the shrink coefficient is fundamental to obtain the optimal solution when the range of the design variables is very different.

After the reproduction has finished for one generation, the new generation is evaluated by using the fitness value for every individual and the same process is repeated until the convergence of the evolutionary algorithm is achieved. Different convergence criteria may be used to stop the algorithm. The first one is based on the accuracy obtained for the best individual in the last generation, but a limit in the number of generations is also established to stop the process. Once the convergence is achieved, the fitness of the last generation is evaluated and a family of best individuals is obtained. The family of best individuals is selected from those linkages whose fitness value is below a threshold. This family of linkages is used as the initial guess for the deterministic approach to form the hybridization process which is described in the following subsection.

4.2. Hybrid Algorithm

Figure 6 shows the flowchart of the hybrid algorithm including the stochastic and deterministic optimization. On the left-hand side of Figure 6, the scheme of the evolutionary technique is shown. The right side in the same figure shows the deterministic part of the hybrid algorithm. The algorithm starts with the definition by the designer of the desired function based on the required motion for the linkage. The designer also establishes the EA parameters that will be used in the algorithm (e.g. the operators for selection, crossover, etc.). After that the optimization process starts with the generation of the first population. The fitness evaluation of this first generation requires the estimation of the error by using deterministic optimization to obtain the orientation angle, β_g, and scaling factor, s. If the fitness value is below a threshold, a family of linkages is selected to be optimized by the deterministic approach. This rarely occurs in the first generation and several generations are necessary to cross from the probabilistic approach to the deterministic optimization as is shown in Figure 6. The deterministic approach uses the best individuals selected from the evolutionary algorithm which are called Family 1. These individuals are optimized irrespective of their fitness values because local optimization could lead to obtaining better individuals among those with worse initial expectation. The deterministic approach optimizes each individual independently to obtain a second family called Family 2. The solution is selected as the best linkage of this second family.

5. Numerical Examples

In this section two examples are presented in order to demonstrate the capacity of the hybrid algorithm. In the first example a four-bar mechanism is selected to be synthesized to

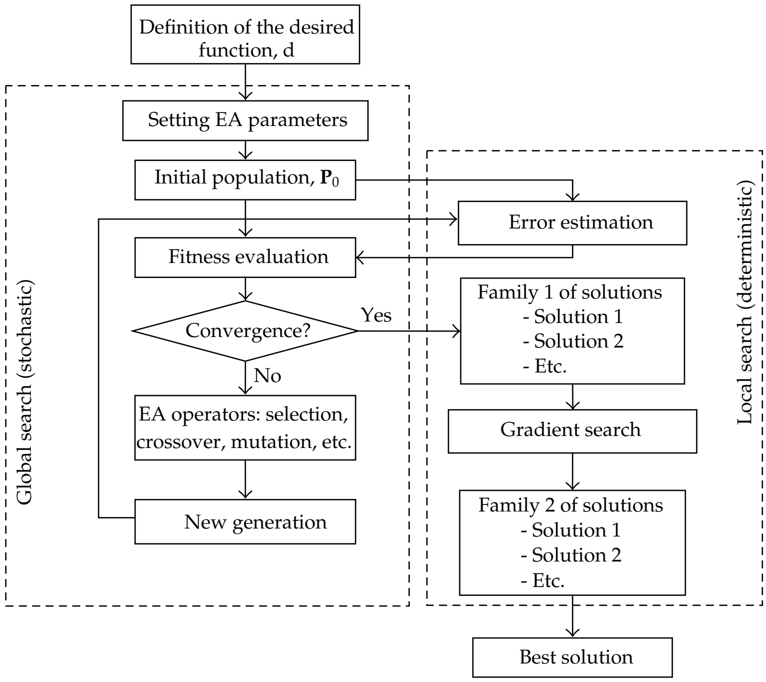


Figure 6: Flowchart of the hybrid algorithm.

generate a right angle path. The example does not correspond to any actual implementation in engineering design, but this type of path is a challenging objective and demonstrates the accuracy, robustness, and efficiency of the proposed approach. The second example is a practical application in the design of an actual machine. The results in these examples are divided into two stages. The first one is the result obtained by the evolutionary algorithm and the second one is the result obtained by the complete hybrid algorithm which includes the local optimization approach in the dimensional synthesis.

5.1. Four-Bar Linkage Generating a Right Angle Path

In this example the methodology is applied to the synthesis of a four-bar mechanism. The scheme of the mechanism is the same as that used in Section 2 (see Figure 1). Likewise the constraints and design variables are given by (2.2) and (2.3), respectively. The aim of the problem is that the coupler point, P, of the synthesized linkage describes a right angle path during the motion. The path is defined by 11 prescribed points whose coordinates are shown in the first two rows in Table 1. Table 2 shows the values of the operator factors used in the evolutionary algorithm. It is important to highlight the small size of the population and the maximum number of generations.

The best resulting mechanism and the path followed by the coupler point in the evolutionary part of the algorithm is shown in Figure 7(a), in addition to the desired precision points. The evolutionary algorithm takes 189.59 seconds to achieve the convergence with an error of 2.439 mm^2 using an Intel Core I5 PC. As can be observed in this figure, the generated path approximates well to the desired one; however, there is clearly a lack of accuracy. In

Table 1: Desired path and the path generated at the convergence with the proposed algorithm.

Paths		1	2	3	4	5	6	7	8	9	10	11
Desired (mm)	x_d	0	0	0	0	0	0	3.00	6.00	9.00	12.00	15.00
	y_d	15.00	12.00	9.00	6.00	3.00	0	0	0	0	0	0
EA (mm)	x_g	0.21	0.06	0.08	−0.24	−0.29	0.82	2.96	5.70	8.77	11.92	14.97
	y_g	15.33	12.20	8.75	5.27	2.76	1.14	0.03	−0.59	−0.68	−0.17	0.94
Hybrid (mm)	x_s	0.22	−0.16	−0.18	0.04	0.11	0.00	3.00	5.99	9.00	12.01	14.93
	y_s	14.95	12.01	8.99	6.00	2.98	0.24	0.05	−0.17	−0.26	−0.11	0.32

Table 2: Values of the different factors used in the evolutionary algorithm.

EA factors	$r_{\max}$	$g_{\max}$	ef	rf	sM	hM
Values	150	10	0.02	0.8	0.8	0.4

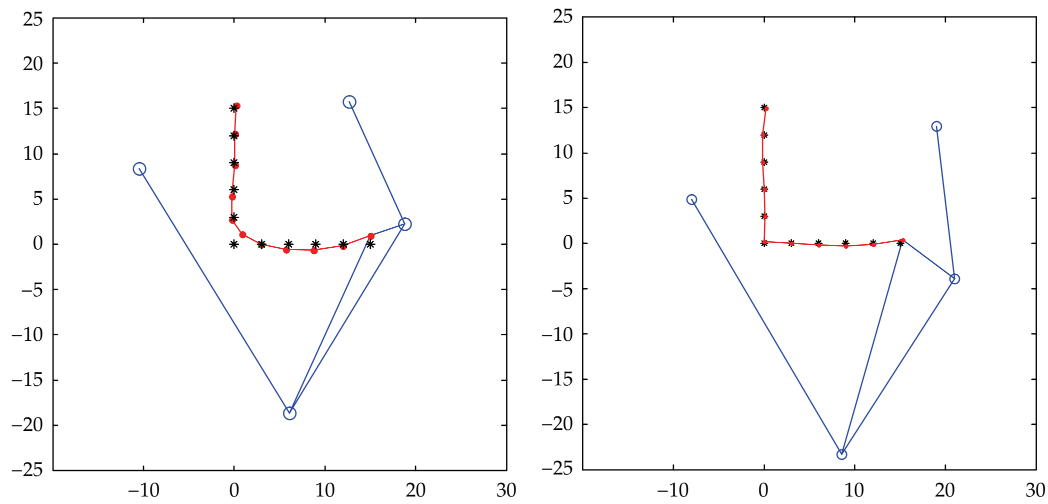
order to compare the result with the desired path, the third and fourth rows of Table 1 show the coordinates of the generated points. The solution for the hybrid algorithm is shown in Figure 7(b) where the path followed by the coupler point fits very well with the desired one. In the last two rows of Table 1 the coordinates of the generated path are shown and the last row of Table 3 shows the values for the design variables.

The error at convergence is $0.2025\,\text{mm}^2$ and the time necessary to achieve the convergence was 212 seconds, which is a very reasonable computational cost in this kind of problem.

Since it is stochastic, the results differ each time the algorithm runs. In order to evaluate the robustness, the algorithm was run 30 times and the sample mean error obtained at convergence was $0.309\,\text{mm}^2$ with a sample standard deviation of $0.211\,\text{mm}^2$. The sample mean CPU time to achieve the convergence was 215.05 seconds with a standard deviation of 19.031 seconds.

5.2. Application to a Mechanism for Injection Machine

In this example the methodology has been applied to the design of a mechanism for die-cast injection machine. Figure 8(a) shows the scheme of such a machine, where the system for the injection of zamak alloys is shown at the top. The mould is located below the injection system (not shown in the figure). The system for the displacement of the mould is shown on the left-hand side of the figure. Figure 8(b) shows the detail of the injection system where it is possible to see the linkage used for this purpose. The mechanism selected for this application is a combination of a four-bar linkage together with a slider-crank mechanism connected by the coupler link. The motion of the slider follows a straight line pushing the zamak alloy through the entrance to fill the mould. This motion must be controlled in order to fill the mould adequately. To obtain good quality in the manufacturing process a rapid, motion of the slider is necessary initially, then a slower motion, and finally a fast backward motion when the mould has been filled. This motion of the slider is coordinated with the input link which is driven by an electric motor with constant velocity (see Figure 8(b)). The precision points are set every 18 deg of the motor rotation, or in other words, 20 precision points are selected for a full rotation of the motor. The coordinates of the precision points are shown in Table 4 and the desired motion is dotted in Figure 9.



(a) (b)

Figure 7: (a) Solution with the evolutionary algorithm and (b) with the hybrid algorithm.

Table 3: Design variables.

Design variables	x_0 (mm)	x_0 (mm)	θ_1 (rad)	θ_{20} (rad)	L_1 (mm)	L_2 (mm)	L_3 (mm)	L_4 (mm)	L_5 (mm)	α (rad)
EA solution	−10.47	8.34	0.31	1.68	24.30	14.81	24.48	31.70	4.02	−0.68
Hybrid solution	−8.01	4.91	0.28	5.50	28.16	16.94	23.05	32.68	7.13	−1.63

The scheme of the mechanism to be synthesized is shown in Figure 10 together with the twelve design variables. Figure 9 compares the results for the evolutionary algorithm and the hybrid optimization approach and Table 4 gives the values of the coordinates generated in all cases. Finally, Table 5 shows the values of the design variables at convergence for the evolutionary algorithm and the hybrid algorithm.

Despite of the difficulty of the problem, the graphical results in Figure 9 show that the evolutionary algorithm provides good accuracy in general; however, in the central part of the curve the accuracy is lower. The hybrid algorithm enhances the accuracy in this zone and provides a very good solution.

The sample mean error obtained by the hybrid algorithm is 357.70 mm^2 with a standard deviation of 14.07 mm^2. The mean CPU time to achieve the convergence is 623.17 seconds with a standard deviation of 46.40 seconds.

6. Concluding Remarks

In this paper a hybrid optimization approach has been presented with application to the optimal dimensional synthesis of planar mechanisms. The objective function is selected using a new error estimator defined by means of the precision points. This error estimator enables the evaluation of the fitness of the function without influence of translation, rotation, and scaling effects. The error estimation is done using a local optimization procedure

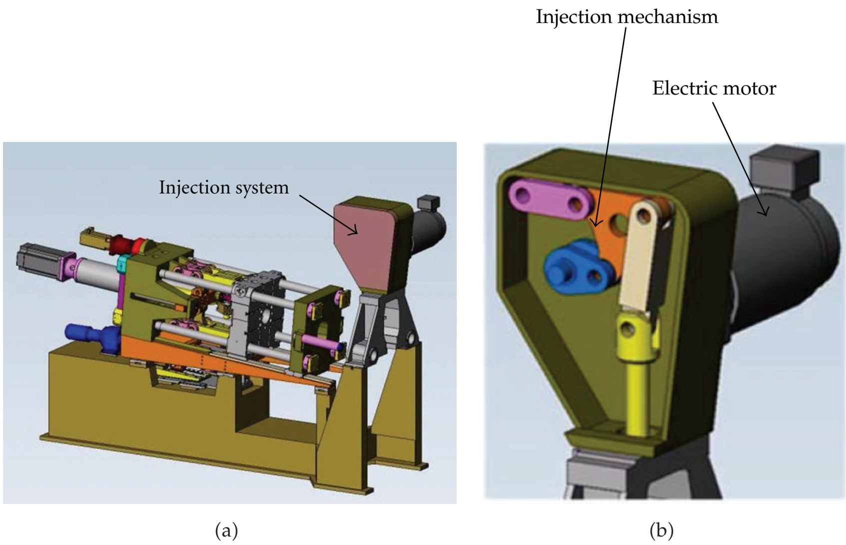


Figure 8: (a) Injection moulding machine and (b) detail of the injection system.

Table 4: Desired path and the path generated at convergence with the proposed algorithm.

	Desired y_d (mm)	EA solution y_g (mm)	Hybrid solution y_s (mm)
1	900	892.33	908.6
2	800	757.53	776.8
3	600	588.30	587.4
4	400	417.67	402.7
5	200	274.19	255.0
6	190	178.37	165.4
7	180	137.35	143.2
8	170	140.40	161.5
9	160	161.43	161.6
10	150	173.98	146.8
11	140	166.79	133.0
12	130	143.37	123.9
13	120	113.29	117.4
14	110	88.23	110.3
15	100	83.79	104.2
16	150	125.08	118.4
17	200	250.88	219.1
18	500	491.63	496.4
19	800	777.75	785.5
20	900	937.63	908.4

providing a very efficient hybrid algorithm. The hybrid algorithm combines the advantages of both stochastic and deterministic approaches to improve the robustness and accuracy. Two examples have been presented in the paper to demonstrate the capacity of the method. The examples show that the proposed method not only achieves the convergence but also demonstrates how the accuracy is improved by the combination of the two procedures.

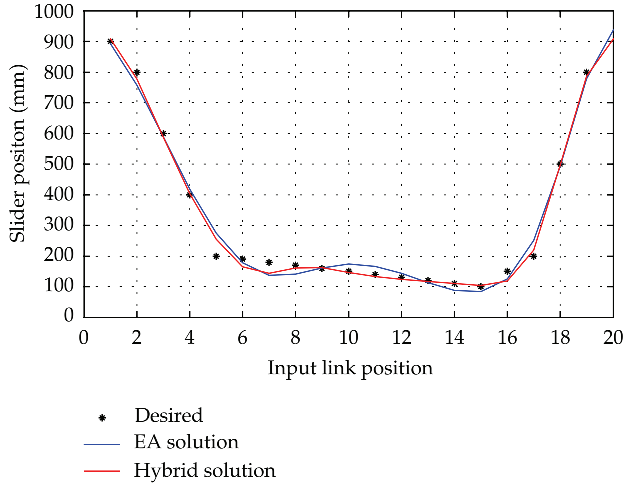


Figure 9: Desired motion, EA, and hybrid solution.

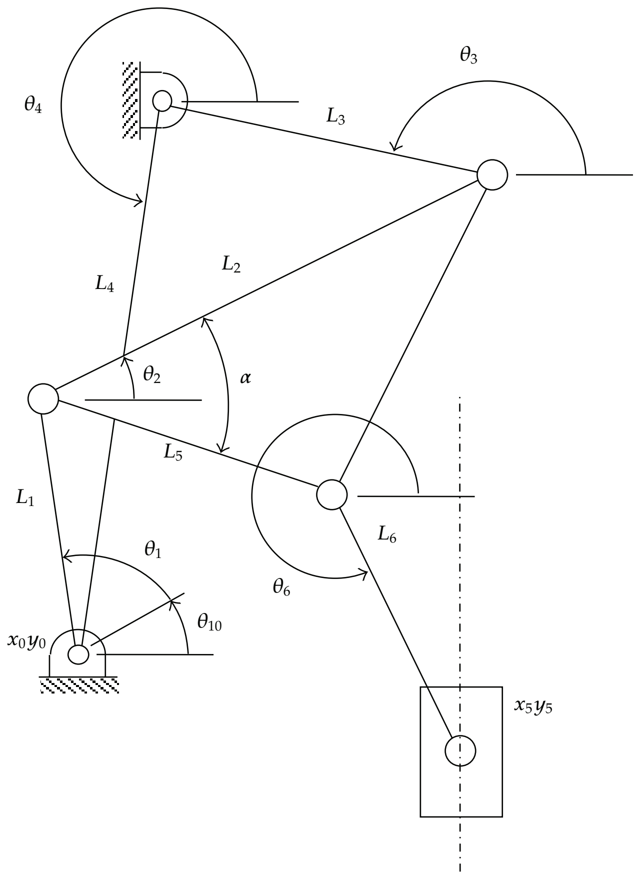


Figure 10: Scheme of the mechanism for injection.

Table 5: Design variables.

Design variables	x_0 (mm)	y_0 (mm)	θ_{10} (rad)	θ_4 (rad)	L_1 (mm)	L_2 (mm)	L_3 (mm)	L_4 (mm)	L_5 (mm)	L_6 (mm)	α (rad)	x_5 (mm)
EA solution	810.92	311.51	2.745	−1.23	517.03	992.82	991.66	1227.4	1161.7	722.41	−0.286	100
Hybrid solution	537.39	393.31	2.785	−1.4	416.52	603.75	554.85	724.41	611.08	372.27	0.2437	100

To do this, the examples depict the solution for the case of the evolutionary algorithm working alone, and then the solution improved by the hybrid algorithm. This shows how the evolutionary algorithm provides an approximation to the solution and then the local optimization improves the accuracy. In both examples the solution provides good designs and the generated curves fit very well with the desired ones. In summary, the hybrid algorithm is a valuable tool for the design of mechanisms when highly demanding requirements are imposed. Thus, the conclusion we draw is that the appropriate combination of stochastic and deterministic algorithms has an enormous potential in the more effective solution of optimization problems in the design of mechanisms. This work will be further developed for the solution of other mechanism design problems by adapting the algorithm. Furthermore, another future task in this field aims to improve the efficiency of the hybrid optimizer by using the most recent developments in metaheuristic approaches such as Particle Swarm Optimization and Differential Evolution.

Acknowledgment

This paper has been developed in the framework of the Project DPI2010-18316 funded by the Spanish Ministry of Economy and Competitiveness.

References

[1] F. Freudenstein, “An analytical approach to the design of four-link mechanisms,” *Transactions of the ASME*, vol. 76, pp. 483–492, 1954.

[2] G. N. Sandor, *A general complex number method for plane kinematics synthesis with applications [Ph.D. thesis]*, Columbia University, New York, NY, USA, 1959.

[3] A. G. Erdman, “Three and four precision point kinematic synthesis of planar linkages,” *Mechanism and Machine Theory*, vol. 16, no. 3, pp. 227–245, 1981.

[4] S. Krishnamurty and D. A. Turcic, “Optimal synthesis of mechanisms using nonlinear goal programming techniques,” *Mechanism and Machine Theory*, vol. 27, no. 5, pp. 599–612, 1992.

[5] J. Mariappan and S. Krishnamurty, “A generalized exact gradient method for mechanism synthesis,” *Mechanism and Machine Theory*, vol. 31, no. 4, pp. 413–421, 1996.

[6] R. Sancibrian, P. García, F. Viadero, and A. Fernández, “A general procedure based on exact gradient determination in dimensional synthesis of planar mechanisms,” *Mechanism and Machine Theory*, vol. 41, no. 2, pp. 212–229, 2006.

[7] A. Kunjur and S. Krishnamurty, “Genetic algorithms in mechanical synthesis,” *Journal of Applied Mechanism and Robotics*, vol. 4, no. 2, pp. 18–24, 1997.

[8] J. A. Cabrera, A. Simon, and M. Prado, “Optimal synthesis of mechanisms with genetic algorithms,” *Mechanism and Machine Theory*, vol. 37, no. 10, pp. 1165–1177, 2002.

[9] J. A. Cabrera, F. Nadal, J. P. Muñoz, and A. Simon, “Multiobjective constrained optimal synthesis of planar mechanisms using a new evolutionary algorithm,” *Mechanism and Machine Theory*, vol. 42, no. 7, pp. 791–806, 2007.

[10] J. H. Holland, *Adaptation in Natural and Artificial Systems*, The University of Michigan Press, Ann Arbor, Mich, USA, 1975, An introductory analysis with applications to biology, control, and artificial

intelligence.

[11] K. A. DeJong, *An analysis of the behaviour of a class of genetic adaptive system [Ph.D. thesis]*, University of Michigan, Ann Arbor, Mich, USA, 1975.

[12] N. Nariman-Zadeh, M. Felezi, A. Jamali, and M. Ganji, “Pareto optimal synthesis of four-bar mechanisms for path generation,” *Mechanism and Machine Theory*, vol. 44, no. 1, pp. 180–191, 2009.

[13] W. Y. Lin, “A GA-DE hybrid evolutionary algorithm for path synthesis of four-bar linkage,” *Mechanism and Machine Theory*, vol. 45, no. 8, pp. 1096–1107, 2010.

[14] M. Khorshidi, M. Soheilypour, M. Peyro, A. Atai, and M. S. Panahi, “Optimal design of four-bar mechanisms using a hybrid multi-objective GA with adaptive local search,” *Mechanism and Machine Theory*, vol. 46, no. 10, pp. 1453–1465, 2011.

[15] I. Ullah and S. Kota, “Optimal synthesis of mechanisms for path generation using fourier descriptors and global search methods,” *Journal of Mechanical Design*, vol. 119, no. 4, pp. 504–510, 1997.

[16] I. Fernández-Bustos, J. Aguirrebeitia, R. Avilés, and C. Angulo, “Kinematical synthesis of 1-dof mechanisms using finite elements and genetic algorithms,” *Finite Elements in Analysis and Design*, vol. 41, no. 15, pp. 1441–1463, 2005.