Optimization of Mechanisms by using a Minimum Distance Function

I. Fernández de Bustos, G. Urkullu, V. García Marina, R. Ansola

Abstract

The study examines the application of a general minimum distance error function to the dimensional kinematic synthesis of bidimensional mechanisms. The minimum distance approach makes it possible to solve the problem maintaining the same generality as that of the minimum deformation energy method while solving the problems that occasionally appear in the former method involving low stiffness mechanisms. It is a general method that can deal both with unprescribed and prescribed timing problems, and is applicable for path generation problems, function generation, solid guidance, and any combination of the aforementioned requirements as introduced in the usual precision point scheme. The method exhibits good convergence and computational efficiency. The minimum distance error function is solved with a sequential quadratic programming (SQP) approach. In the study, the synthesis problem is also optimized by using SQP, and the function can be easily adapted to other methods such as genetic algorithms.

In the study, the minimum distance approach is initially presented. Subsequently, an efficient SQP method is developed by using analytic derivatives for solving. The next point addresses the application of the concept for the synthesis of mechanisms by using an SQP approach with approximate derivatives. This delivers a situation where the optimization is performed on an error function that itself consists of an inner optimization function. A few examples are presented and are also compared with the minimum deformation energy method. Finally, a few conclusions and future studies are discussed.

Keywords

Mechanism Synthesis; SQP; Optimization

Introduction

The synthesis of mechanisms is a matter of broad interest, as it can be derived from the high volume of related publications that appeared in the last few years. In the process of development of a mechanism, we first approach the type synthesis that deals with the selection of the types of elements that are to be employed to construct the mechanism. Second, it is necessary to tackle topology synthesis (also termed as number synthesis) that is related to the amount and distribution of the selected elements. Finally, dimensional synthesis is used to obtain the dimensions of the elements, and thus the mechanism finally fulfils the requirements. The study is related to this final step, we provide a brief review of the methods available for the dimensional synthesis of mechanisms. The dimensional synthesis of mechanisms potentially corresponds to the step that attracts maximum interest in the scientific community. This is because most of the type and topology synthesis is typically performed in an intuitive way since the number of factors involved is extremely high including costs and availability of elements, weight, and expertise. The aforementioned considerations are extremely difficult to quantify.

The definition of a dimensional synthesis problem can also be significantly different. The most usual paradigm is that of the precision points where we define a set of precision points and the mechanism is required to fulfil a set of requirements for each precision point. This is a significantly flexible approach that is typically employed in numerical methods. Alternative options are typically used when the target of the optimization is of homogeneous nature as observed in path generation, solid guidance, or function generation. In this case, it is possible to define the objective of the optimization in the form of analytic functions or tables representing the functions. This last type of option is less flexible in the sense of the problem to be solved although it allows the use of a wider set of methods. Thus, in addition to the typical case for analytic methods, there are also a few specific cases of numerical methods that require the aforementioned type of formulations. While using precision points, another point of interest is if the input element position is known for each precision point. In this case, we refer to prescribed timing. If it is unknown, it is termed as unprescribed timing. The second problem is more difficult to solve, as the prescribed timing allows the solution of the position problem and subsequently computes the degree of verification of the precision point.

In order to solve the dimensional synthesis of mechanisms, three families of methods are typically used as follows: graphical methods, analytic methods, and numerical methods. Graphical methods were most frequently used in earlier studies of dimensional synthesis of mechanisms. They are applicable to simple mechanisms, because the graphical constructions increase in complexity and are cumbersome when applied to complex mechanisms. A few examples of the aforementioned methods are found in references [1]-[3]. A few examples of analytic methods are found in [4]-[9]. They typically rely on closure equations, and thus it is necessary to formulate the aforementioned equations. This typically implies that the algorithms should be developed on a per topology basis. In any case, the major disadvantages of analytic methods are that they are typically restricted to a specific type of requirements (typically path generation, function generation, or solid rigid guidance) and that the number of parameters to be determined should be equal to or exceed that of the precision points in the case of problems defined by precision points. This last condition is derived from the fact that analytic methods are not suitable to obtain an approximate solution albeit an exact one. The last group of methods is numerical. As previously mentioned, the aforementioned methods typically apply the precision point definition of the problem. Additionally, they typically require a transformation to a precision point approach in the case of problems in which the definition is in another form. This is typically not a major concern, and thus numerical methods typically correspond to the most flexible and allow path or function generation or solid guidance as well as any type of requirements or combinations. There are two important families of numerical methods based on the use or no use of heuristics. Non-heuristic methods are typically based on classical optimization techniques where the function to be optimized is analysed, and the most frequently used approach is sequential quadratic programming that requires the derivatives of the function that can be obtained in either in an approximate or an analytic manner. Conversely, heuristic methods typically rely on the simulation of natural phenomena such as the evolution of species (genetic algorithms and evolutionary algorithms) and biological behaviour of nervous systems (neural networks). Applications of the aforementioned methods are found in references [10]-[22] among others. Non-heuristic methods have been examined extensively since heuristic methods require intensive computational effort, and thus they were developed later. A few applications of the aforementioned methods can be found in references [23]-[28]. Most of the numerical methods attempt to solve the prescribed timing problem to simplify the problem. Furthermore, several real problems can be formulated in this manner. The disadvantage of the approach is its loss of generality. A few examples of methods suitable for non-prescribed timing problems can be found in [24],[25],[26],[10]. Most of the aforementioned methods are based on the minimal deformation energy function defined by Avilés et al in [24]. The function exhibits a significant advantage in the sense of versatility although it exhibits the limitation wherein it favours mechanisms with extremely low stiffness in topologically complex mechanisms, and this can lead to inadequate solutions. In the present study, a new approach is presented (as opposed to computing an energetic function) wherein the configuration of the mechanism that delivers the lower distance with respect to the set of requirements that appear in each precision point is computed. The function can be considered as a formal generalization of the study presented in [10] where the energetic function was modified to approximate the distance, and this allowed for the use of genetic algorithms for the synthesis of mechanisms with unprescribed timing and based on precision points. A preliminary approach for the optimization of the function was presented in [29]. In the study, derivatives are obtained in an analytic way, and the application of the function to the optimization of mechanisms with an SQP method is presented.

Minimum distance approach

We consider the problem shown in figure 1 as an introduction to the problem. The coupling node in the fourbar is required in the considered precision point to reach the indicated target point. The minimum deformation energy approach ([24],[30]) models the mechanism as constructed by linear elastic elements. With respect to the given precision point, we compute the configuration of the mechanism that verifies the restrictions with the lower deformation energy. This corresponds to figure 1.b. This defines the minimum deformation energy problem that should be solved for each precision point. The summation of the obtained energies for each precision point is the error function to be optimized. The minimum distance approach defines a similar concept although the mechanism is constructed on rigid elements that cannot be deformed. In the aforementioned conditions, in the general case, it is not possible to precisely verify the requirements in a precision point. Thus, we compute the configuration of the mechanism that delivers the lower distance to the requirements. The minimum distance position problem involves obtaining the assembled configuration that delivers the lower distance from the mechanism to the requirements. With respect to the example, the solution is presented in figure 1.c.

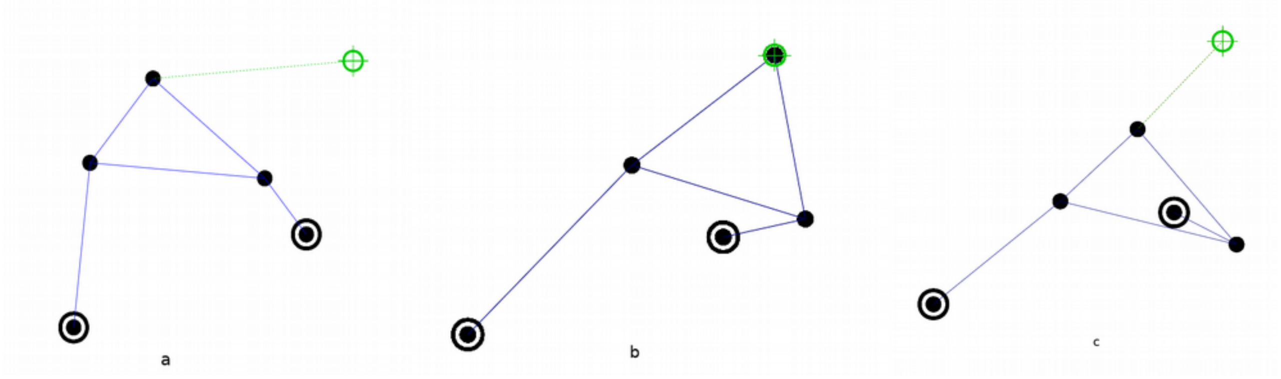

Figure 1 :a) Mechanism with a precision point with a node to point requirement. a) solution for the minimum deformation energy problem. c) solution for the minimum distance problem

As in the case of the minimum deformation energy approach, the error function for the synthesis can be constructed as the summation of the distance to the requirements for each precision point. This can also be weighted to consider the different relevancies of the precision points.

Evidently, there are situations where multiple requirements are introduced in a single precision point. In the aforementioned cases, we can use a weighted squared summation of the distances to each of those requirements as used in the minimum deformation energy approach. There are also situations in which the nature of requirements is different. The aforementioned restrictions are termed as after node to point although also we can introduce restrictions where the distance can be angular or node to line or similar. In any case, the aforementioned approach can be applied. With respect to the current document, we reduce the requirements to those of node to point and node to line where the target is fixed. Thus, the synthesis point s is expressed as follows:

$$D_s(\boldsymbol{l_u}, \boldsymbol{x_s}) = \sum_{o=0}^{N_r} q_{so} d_{so}^2(\boldsymbol{l_u}, \boldsymbol{x_s}) \tag{1}$$

where:

q_{so} : weight factor for requirement o in the synthesis point s

$d_{so}^2(\boldsymbol{x_s})$: distance from the point declared in the requirement to the required target point.

$\boldsymbol{l_u}$: vector containing the dimensions of the mechanism.

$\boldsymbol{x_s}$: coordinates of the nodes of the mechanism.

For example, with respect to the case of a node to point requirement:

$$d_{pso}^2(\boldsymbol{l_u}, \boldsymbol{x_s}) = (x_{so} - x_{iso})^2 + (y_{so} - y_{iso})^2 \tag{2}$$

where,

x_{iso} the x coordinate of the point that the node should reach

y_{iso} the y coordinate of the point that the node should reach

x_{so} the x coordinate of the node required to reach (x_{iso}, y_{iso})

y_{so} the x coordinate of the node required to reach (x_{iso}, y_{iso})

Evidently x_{so} and y_{so} are part of $\boldsymbol{x_s}$. If $\boldsymbol{x_s}$ defines a possible assembled configuration of the mechanism, the dimensions of the mechanism as defined by $\boldsymbol{x_s}$ must match those of the undeformed elements of the mechanism. For example, in order to obtain a mechanism defined by truss elements, the following restrictions should be satisfied as follows:

$$r_k(\boldsymbol{l_u}, \boldsymbol{x_s}) = l_k(\boldsymbol{x_s}) - l_{uk} = 0, k = 1,.., N_t \tag{3}$$

being:

l_{uk} : original (undeformed) length of the k-th truss

$l_k(\boldsymbol{x_s})$:length of the k-th truss as defined by the $\boldsymbol{x_s}$ coordinates of the nodes

This is termed as the length approach. An evident alternative (that is easier to implement) is derived as follows:

$$r'_k(\boldsymbol{l_u}, \boldsymbol{x_s}) = l_k^2(\boldsymbol{x_s}) - l_{uk}^2 = 0, k = 1,.., N_t \tag{4}$$

Which (as shown later) leads to different derivatives albeit similar performance. This is termed as the squared length approach. Given this, the problem of minimum distance is defined as follows:

minimize: $D_s(\boldsymbol{l_u}, \boldsymbol{x_s})$

with respect to: $\boldsymbol{x_s}$

subject to:

$$r_k(\boldsymbol{l_u}, \boldsymbol{x_s}) = 0, k = 1,.., N_t \tag{5}$$

or subject to:

$$r'_k(\boldsymbol{l_u}, \boldsymbol{x_s}) = 0, k = 1,.., N_t \tag{6}$$

When the minimum distance problem is defined, we approach the synthesis problem. In order to formulate the same, we consider that the best mechanism is that which delivers the lower value in all the synthesis points. It is possible to approach the problem via Pareto considerations although this not of significant practical use. Thus, a weighted approach is considered again. The synthesis problem error function that is used is defined as follows:

$$D(\boldsymbol{l_u})=\sum_{s=1}^{N_s} Q_s min_{x_s}(D_s(\boldsymbol{l_u},\boldsymbol{x_s})) \qquad [7]$$

where,

N_s : number of synthesis points to be considered.

Q_s : weight factor for the synthesis point s .

As shown in [31], we use $\boldsymbol{l_u}$ as vector of variables to be optimized or a set $\boldsymbol{x_0}$ of initial coordinates. The final option allows the inclusion of the assembly configuration in the optimization although it leads to more complicated developments. In this stage, the optimization computes derivatives via finite differences, and thus the use of coordinates is selected. Thus, the optimization problem involves minimizing $D(\boldsymbol{x_0})$ with respect to $\boldsymbol{x_0}$.

Minimizing the minimum distance function

The main problem while attempting to approach the synthesis of mechanisms is the computational cost. Thus, it is necessary to reduce it to the maximum possible extent. The minimum distance function appears in the inner loop of the process, and thus it is important to focus on it. In order to reduce the cost to the maximum possible extent, we resort to a Sequential Quadratic Programming (SQP) approach. This is also important because we can use the result in the previous precision point as a starting vector for solving the next, and thus a reliable starting vector is exhibited. This is extremely important while using SQP. In order to introduce the dimensional restrictions, a Lagrange multiplier approach is applied. In order to further reduce the computational cost, both first and second analytical derivatives of the aforementioned restrictions are also obtained. This also contributes to increasing precision in the solution of the problem. The required derivatives include those of the error function and those of the restrictions. With respect to the derivatives of the error function, we obtain them through the addition of the derivatives of the different targets of the optimization. This leads to an assembly algorithm that is similar to that of the finite element techniques. Within the limits considered in the study, the problem is warranted as free of saddle points or undesired extrema. Thus, a simple Karush-Kuhn-Tucker solver is considered based on the null subspace approach. The specific solution in the null subspace method is selected as the minimal least squares solution to improve numerical behaviour. The final symmetric system of equations is solved by using the method presented in [32]. It is important to note that the introduction of angular requirements can easily lead to situations with saddle points in other circumstances.

Requirements

With respect to the case of node to point requirements, the following expression is obtained:

$$\frac{\partial D_s(\boldsymbol{l_u},\boldsymbol{x_s})}{\partial \boldsymbol{x_s}}=\sum_{o=0}^{N_r}\frac{\partial q_{so} d_{pso}^2(\boldsymbol{l_u},\boldsymbol{x_s})}{\partial \boldsymbol{x_s}} \qquad [8]$$

Where, the following expression holds:

$$\frac{\partial q_{so} d_{pso}^2(\boldsymbol{x_s})}{\partial \boldsymbol{x_s}}=q_{so}\begin{bmatrix} 0 \\ \dots \\ -2(x_{tso}-x_{so}) \\ -2(y_{tso}-y_{so}) \\ \dots \\ 0 \end{bmatrix} \qquad [9]$$

The non-null elements appear in the rows that correspond to the node to verify the requirement. With respect to the second derivative, the following expression is obtained:

$$\frac{\partial^2 D_s(\boldsymbol{l_u},\boldsymbol{x_s})}{\partial \boldsymbol{x_s}^2}=\sum_{o=0}^{N_r}\frac{\partial^2 q_{so} d_{pso}^2(\boldsymbol{l_u},\boldsymbol{x_s})}{\partial \boldsymbol{x_s}^2} \qquad [10]$$

Additionally, the following expression holds:

$$\frac{\partial^2 q_{so} d_{pso}^2(\boldsymbol{x_s})}{\partial \boldsymbol{x_s}^2} = q_{so} \begin{bmatrix} 0 & \ldots & 0 & 0 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & \ldots & 2 & 0 & \ldots & 0 \\ 0 & \ldots & 0 & 2 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & \ldots & 0 & 0 & \ldots & 0 \end{bmatrix} \quad [11]$$

Evidently, the crossed derivatives are null for this type of requirement, and thus a diagonal matrix is reached. Additionally, for each component, the non-null elements appear in the rows corresponding to the node to verify the requirement. It is also easy to observe that for a problem with only node to point requirements, the Hessian before restrictions is always positive definite.

The node to line requirement can be defined in several ways. In order use the advantage of floating point properties in the study, the use of angles is avoided. This avoids trigonometric functions in the code. This is important due to their lack of efficiency, and the loss of precision (due to non-strict implementation of IEEE754) in a few implementations. Thus, the line requirement is defined by the tangent of the angle with respect to the horizontal and the signed minimum distance of the line to the origin of coordinates. Thus, the tangent varies from $-\infty$ to $+\infty$, and the distance is positive when the line that delivers the distance in y is positive (see figure 2).

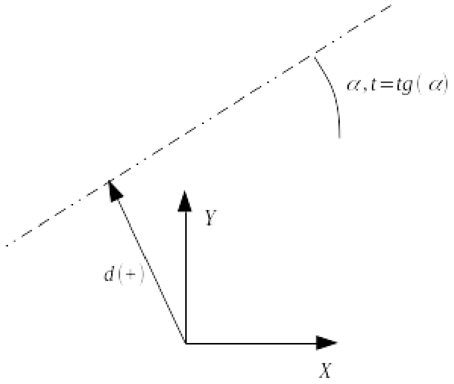

Figure 2: Line definition for the node to line requirement

In the conditions, the distance to line is computed as follows:

$$d_{lso}^2(\boldsymbol{l_u}, \boldsymbol{x_s}) = \left(\frac{y_{so} - t_{iso} x_{so}}{\sqrt{1 + t_{iso}^2}} - d_{iso} \right)^2 \quad [12]$$

where t_{iso} and d_{iso} denote the parameters that define the line that the node is required to reach. In this case, it is important to consider that eq. [12] can lead to numerical problems in cases where t_{iso} is of significant magnitude. In order to avoid the aforementioned problems, the implementation includes the alternatives shown in eq. [13] as follows:

$$d_{lso}^2(\boldsymbol{l_u}, \boldsymbol{x_s}) = \left\{ \begin{array}{c} t_{iso} < -1; \left(\frac{-y_{so}/t_{iso} + x_{so}}{\sqrt{1/t_{iso}^2 + 1}} - d_{iso} \right)^2 \\ -1 \leq t_{iso} \leq 1; \left(\frac{y_{so} - t_{iso} x_{so}}{\sqrt{1 + t_{iso}^2}} - d_{iso} \right)^2 \\ t_{iso} > 1; \left(\frac{y_{so}/t_{iso} - x_{so}}{\sqrt{1/t_{iso}^2 + 1}} - d_{iso} \right)^2 \end{array} \right\} \quad [13]$$

With respect to the derivatives, the following expression is obtained:

$$\frac{\partial q_{so} d_{lso}^2(\boldsymbol{x_s})}{\partial \boldsymbol{x_s}} = q_{so} \begin{bmatrix} 0 \\ \dots \\ \dfrac{2t_{iso} d_{iso} \sqrt{1+t_{iso}^2} - 2t_{iso}(y_{so} - x_{so} t_{iso})}{1+t_{iso}^2} \\ \dfrac{2(y_{iso} - x_{iso} t_{iso}) - 2d_{iso} \sqrt{1+t_{iso}^2}}{1+t_{iso}^2} \\ \dots \\ 0 \end{bmatrix} \quad [14]$$

Furthermore, the equation is suitable for $-1 \le t_{iso} \le 1$. In the case of $t_{iso} < -1$, it is advisable to use the following expression:

$$\frac{\partial q_{so} d_{lso}^2(\boldsymbol{x_s})}{\partial \boldsymbol{x_s}} = q_{so} \begin{bmatrix} 0 \\ \dots \\ \dfrac{-2 d_{iso} \sqrt{1/t_{iso}^2 + 1} - 2(y_{so}/t_{iso} - x_{so})}{1/t_{iso}^2 + 1} \\ \dfrac{2/t_{iso}(y_{iso}/t_{iso} - x_{iso}) + 2/t_{iso} d_{iso} \sqrt{1/t_{iso}^2 + 1}}{1/t_{iso}^2 + 1} \\ \dots \\ 0 \end{bmatrix} \quad [15]$$

If $t_{iso} > 1$, then the following expression is obtained:

$$\frac{\partial q_{so} d_{lso}^2(\boldsymbol{x_s})}{\partial \boldsymbol{x_s}} = q_{so} \begin{bmatrix} 0 \\ \dots \\ \dfrac{2 d_{iso} \sqrt{1/t_{iso}^2 + 1} - 2(y_{so}/t_{iso} - x_{so})}{1/t_{iso}^2 + 1} \\ \dfrac{2/t_{iso}(y_{iso}/t_{iso} - x_{iso}) - 2/t_{iso} d_{iso} \sqrt{1/t_{iso}^2 + 1}}{1/t_{iso}^2 + 1} \\ \dots \\ 0 \end{bmatrix} \quad [16]$$

The second derivative is as follows:

$$\frac{\partial^2 q_{so} d_{lso}^2(\boldsymbol{x_s})}{\partial \boldsymbol{x_s}^2} = q_{so} \begin{bmatrix} 0 & \dots & 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & \dfrac{2t_{iso}^2}{1+t_{iso}^2} & \dfrac{-2t_{iso}}{1+t_{iso}^2} & \dots & 0 \\ 0 & \dots & \dfrac{-2t_{iso}}{1+t_{iso}^2} & \dfrac{2}{1+t_{iso}^2} & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 0 & \dots & 0 \end{bmatrix} \quad [17]$$

This is not a problem if $|t_{iso}| \le 1$, while in other case it is advisable to use the following expression:

$$\frac{\partial^2 q_{so} d_{lso}^2(\boldsymbol{x_s})}{\partial \boldsymbol{x_s}^2} = q_{so} \begin{bmatrix} 0 & \dots & 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & \frac{2}{1/t_{iso}^2+1} & \frac{-2/t_{iso}}{1/t_{iso}^2+1} & \dots & 0 \\ 0 & \dots & \frac{-2/t_{iso}}{1/t_{iso}^2+1} & \frac{2/t_{iso}^2}{1/t_{iso}^2+1} & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 0 & \dots & 0 \end{bmatrix} \quad [18]$$

In all the aforementioned cases, there is the possibility of division by $\pm\infty$. Although this could be a problem in significantly old systems, all IEEE754 compatible systems and most of the modern systems can manage the situation by returning 0 unless the number to be divided also corresponds to infinity, and this is never the case in the presented expressions.

Restrictions

The derivatives of the dimensional restrictions are addressed. Given the purposes if simplicity, only truss length restrictions are examined. There are two possible approaches as previously discussed. With respect to the length approach, we consider a truss k that connects nodes i and j as follows:

$$\frac{\partial r_k(\boldsymbol{l_u}, \boldsymbol{x_s})}{\partial \boldsymbol{x_s}} = \frac{\partial (l_k(\boldsymbol{x_s}) - l_{uk})}{\partial \boldsymbol{x_s}} = \frac{\partial l_k(\boldsymbol{x_s})}{\partial \boldsymbol{x_s}} = \frac{1}{l_k(\boldsymbol{x_s})} \begin{pmatrix} 0 \\ \dots \\ x_{ski} - x_{skj} \\ y_{ski} - y_{skj} \\ \dots \\ x_{skj} - x_{ski} \\ y_{skj} - y_{ski} \\ \dots \\ 0 \end{pmatrix} = \frac{1}{l_k(\boldsymbol{x_s})} \boldsymbol{\Delta x_{sk}} \quad [19]$$

and:

$$\frac{\partial^2 r_k(\boldsymbol{l_u}, \boldsymbol{x_s})}{\partial \boldsymbol{x_s}^2} = -\frac{1}{l_k^3(\boldsymbol{x_s})} \boldsymbol{\Delta x_{sk}^T \Delta x_{sk}} + \frac{1}{l_k(\boldsymbol{x_s})} \begin{bmatrix} 0 & \dots & 0 & 0 & \dots & 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 1 & 0 & \dots & -1 & 0 & \dots & 0 \\ 0 & \dots & 0 & 1 & \dots & 0 & -1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & -1 & 0 & \dots & 1 & 0 & \dots & 0 \\ 0 & \dots & 0 & -1 & \dots & 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 0 & \dots & 0 & 0 & \dots & 0 \end{bmatrix} \quad [20]$$

With respect to the alternative squared length approach:

$$\frac{\partial r'_k(\boldsymbol{l_u}, \boldsymbol{x_s})}{\partial \boldsymbol{x_s}} = \frac{\partial (l_k^2(\boldsymbol{x_s}) - L_k^2)}{\partial \boldsymbol{x_s}} = \frac{\partial l_k^2(\boldsymbol{x_s})}{\partial \boldsymbol{x_s}} = 2 \boldsymbol{\Delta x_{sk}} \quad [21]$$

and:

$$\frac{\partial^2 r'_k(\boldsymbol{l_u},\boldsymbol{x_s})}{\partial \boldsymbol{x_s}^2}=2\begin{bmatrix} 0 & \ldots & 0 & 0 & \ldots & 0 & 0 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & \ldots & 1 & 0 & \ldots & -1 & 0 & \ldots & 0 \\ 0 & \ldots & 0 & 1 & \ldots & 0 & -1 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & \ldots & -1 & 0 & \ldots & 1 & 0 & \ldots & 0 \\ 0 & \ldots & 0 & -1 & \ldots & 0 & 1 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & \ldots & 0 & 0 & \ldots & 0 & 0 & \ldots & 0 \end{bmatrix} \qquad [22]$$

The length approach delivers more complicated derivatives although it is also necessary to consider that it can also lead to a better performance in terms of convergence.

Convergence of the minimum distance function

Prior to approaching synthesis, it is important to verify that the optimization of the minimum distance function accurately converges. Thus, a few experiments are performed. This also serves to compare the behaviour of both formulations. It is also interesting to note how the function behaves in comparison to the minimum deformation algorithm. The first example (figure 3) is a fourbar with a single point to line requirement. In this case, this is a situation where the minimum achievable distance is zero, and thus the minimum distance and minimum deformation energy methods typically yield the same result.

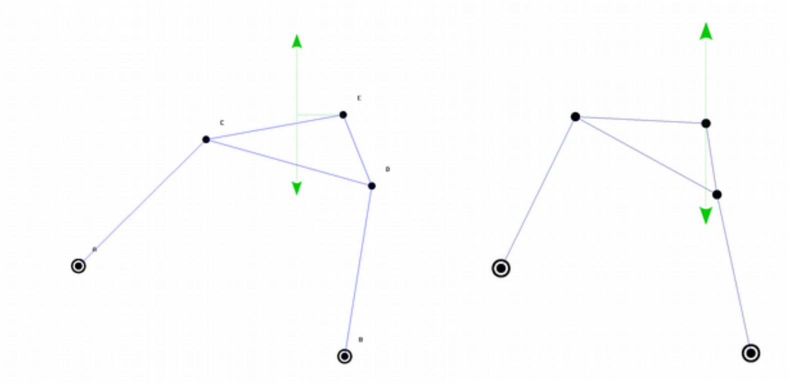

Figure 3: Fourbar in a starting position and requirement (left). Solution obtained with minimum distance and minimum deformation energy (right)

The problem is solved with the minimum distance algorithm (both with length restrictions and squared length restrictions) and the minimum deformation energy algorithm. In all the cases, the algorithms converge to the same solution, and the computational cost is excessively low such that it cannot be measured with precision (it could be derived from the average of several executions although it is better to compare it in more complex problems). The amount of iterations to convergence is also similar. Figure 4 shows the convergence graph obtained for the minimum distance problem by using squared lengths for introducing restrictions. The other approach for introducing restrictions leads to similar results. This is also observed for the minimum deformation energy algorithm although numerical values in this case are not comparable due to the differences in the nature of the error functions.

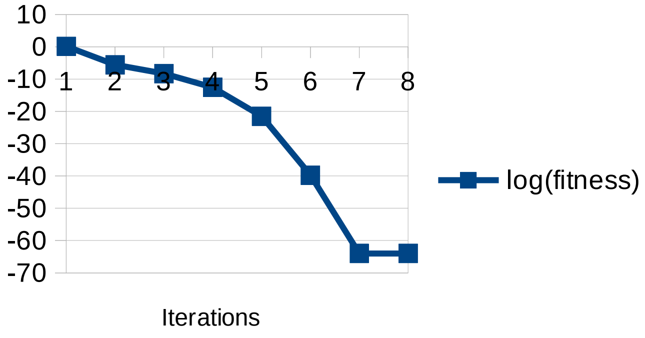

Figure 4: Typical convergence graph. Values with log(fitness)<-64 are rounded to -64

The next problem addresses a problem without an exact solution, and this corresponds to a typical situation while dealing with mechanism synthesis. It is also a fourbar mechanism although a node to point requirement is presently introduced.

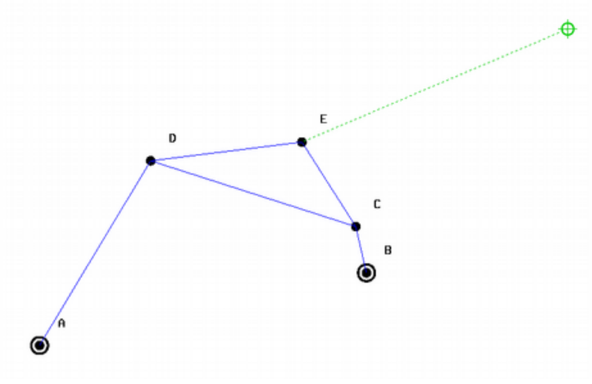

Figure 5: Problem description. Fourbar with node to point requirements

Evidently, the differences among the minimum distance and minimum deformation energy problem are significantly visible. First, a comparison of both restriction formulations is performed. The use of direct length restrictions leads to the result shown in figure 6 (left).

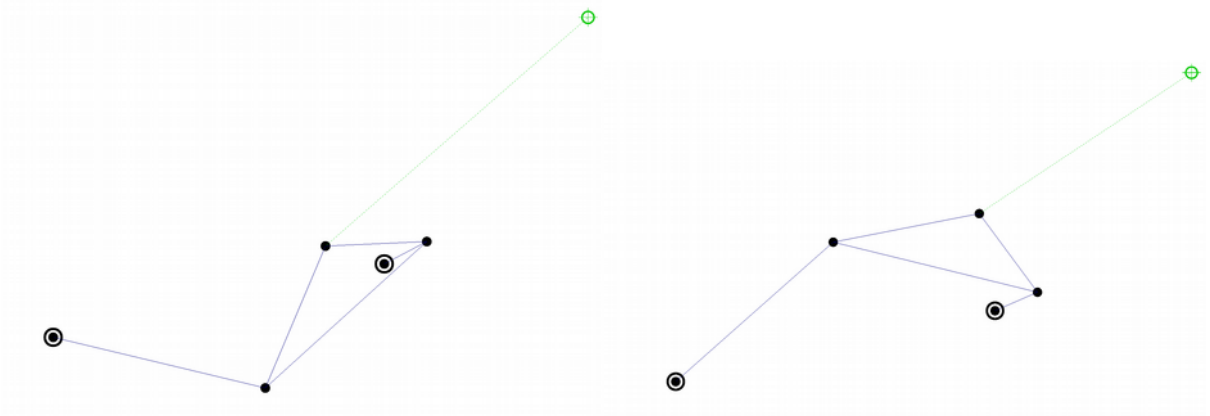

Figure 6: Result obtained with minimal distance function. Left: solution obtained with simple length restrictions. Right: solution obtained with restrictions formulated as lengths are raised to the power of two

The result obtained with the formulation of restrictions based on lengths raised to the power of two lead to a surprisingly better result as shown in figure 6 on the right. As shown in the performed experiments, the difference in the result is significantly common in problems with multiple local minima and where the initial configuration is far from the aforementioned minima. In the aforementioned cases, different values obtained for the Hessian of the restriction can lead to different results. This does not imply that the formulation of lengths raised to the power of two is better. Occasionally, a formulation leads to a better minimum value although both lead to the same final solution in most situations. With respect to the convergence, we observe the evolutions of the aptitude and restriction violation as shown in figure 7.

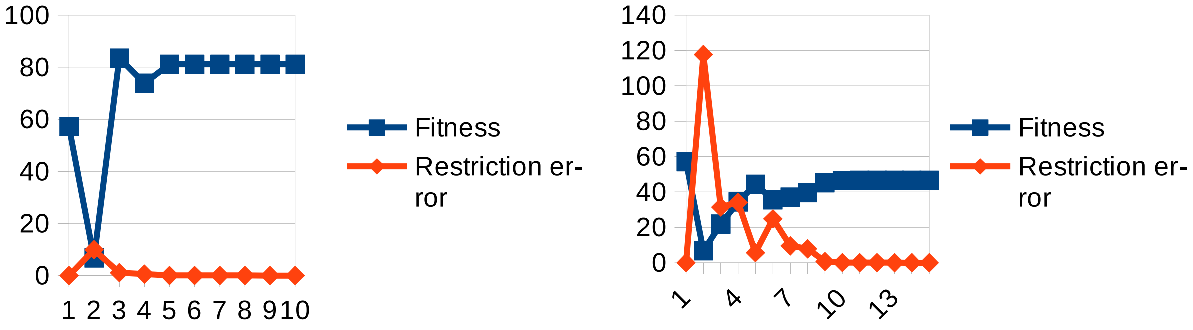

Figure 7:Evolution of fitness and restriction error. Left: Restrictions based in lengths. Right: Restrictions based in lengths raised to the power of two

As shown in figures 7 left and right, the first iteration is equal in both formulations. This is because the initial configuration is undeformed, and thus only the gradient of the restrictions is considered in the iteration. In the second iteration, the difference in the Hessian matrix leads to different search spaces for both algorithms, thereby leading to different results. It is significantly interesting that the first formulation leads to a result that is worse than the initial vector although this is not uncommon while optimizing functions with multiple local optima. The minimal deformation energy algorithm leads to a configuration and evolution as shown in figure 8.

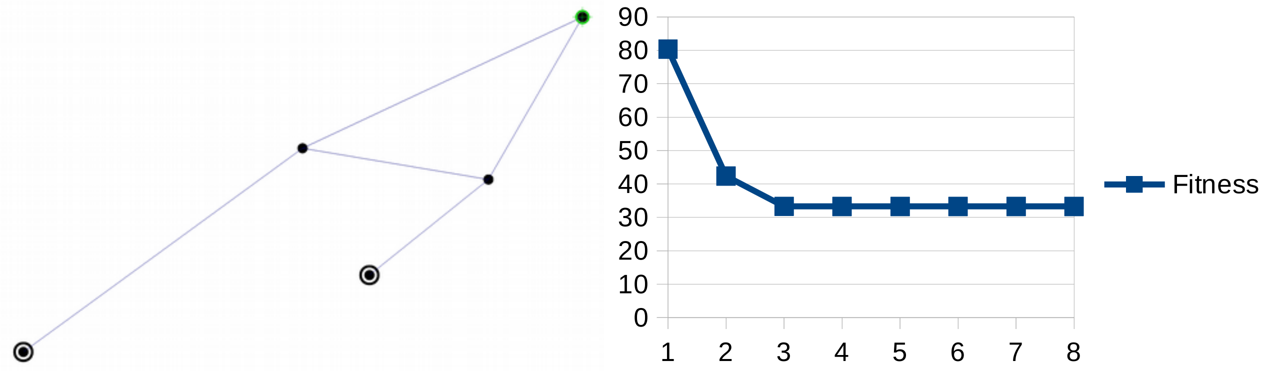

Figure 8: Solution obtained with minimal deformation energy function (left) and optimization evolution (right)

Evidently, it is not possible to compare the fitness obtained with the minimal deformation energy function with that obtained with the minimal distance function although we can compare the number of iterations that are slightly lower (although similar) and a computational cost as that as noted in the previous example that is excessively low such that it is examined in more complex examples. The most important problem of the deformation energy formulation is that it can favour low stiffness mechanisms that are typically useless.

The following example is a complex one. It is based on the double butterfly mechanism. The mechanism kinematics is significantly complex to solve and especially when compared to those of a fourbar. The problem is shown in the figure below:

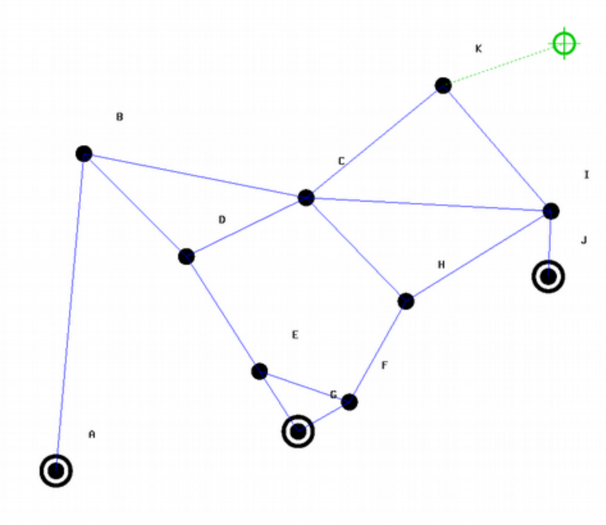

Figure 9: Double butterfly mechanism with a precision point

The minimum distance problem solution with both formulations for restrictions is shown in figure 10:

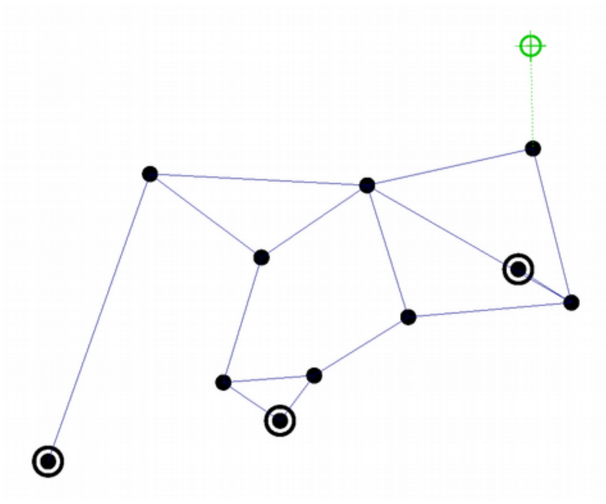

Figure 10: Solution obtained for the minimum distance problem with both simple and alternative length restrictions.

The evolution of both algorithms is shown in figure 11:

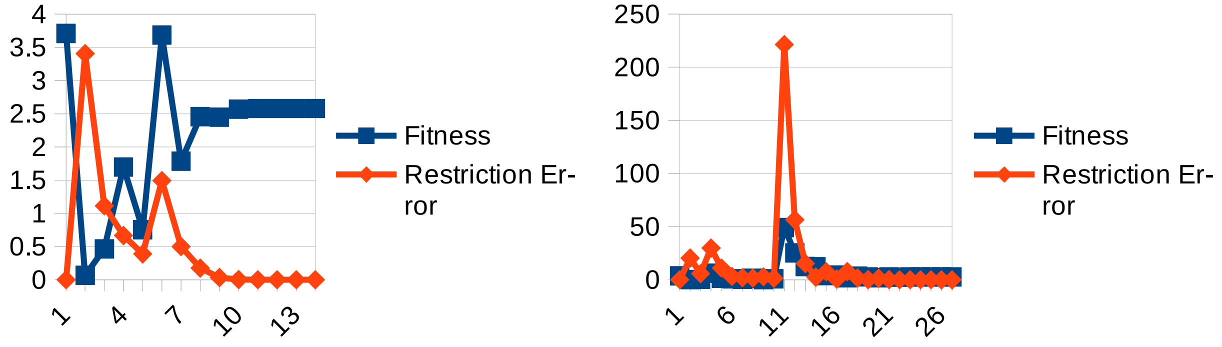

Figure 11: Evolution of the Fitness and Restriction Error. Left: Simple length restrictions. Right: Alternative formulation

As shown in this case, the results are opposite to those of the fourbar. In this case, the simple length restriction formulation leads the optimal best result while the lengths raised to the power of two yield a slightly worst result. In the experiments performed, there is no specific benefit on by using a formulation or the other. In terms of computational cost, both algorithms approximately correspond to 1 ms for each of the 10 iterations. This is confirmed in several tests with different configurations of the same problem.

The result obtained with minimum deformation energy are shown in the figure below along with the convergence graph.

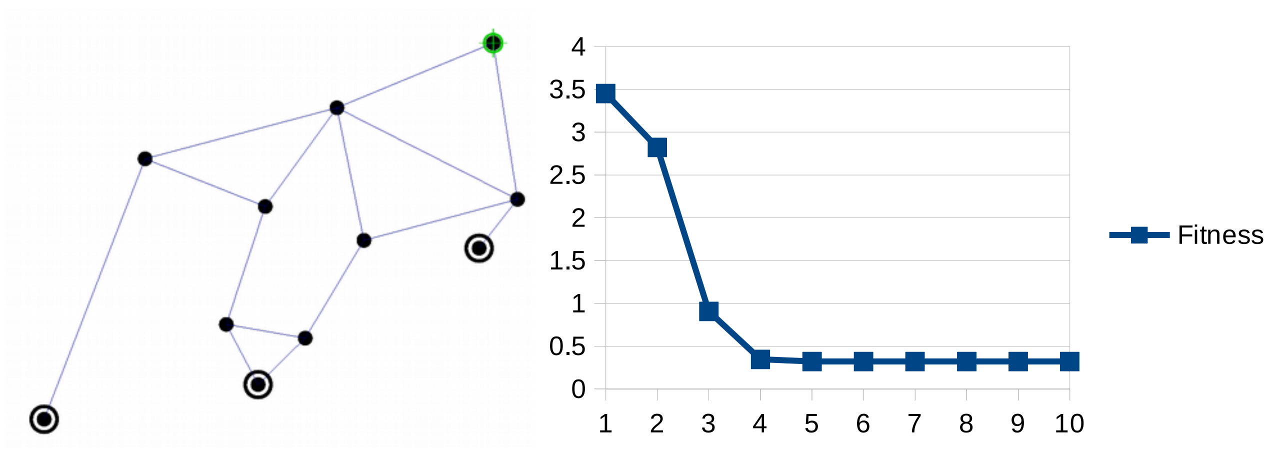

Figure 12: Minimum deformation energy solution and evolution.

The minimum deformation problem is typically solved with a slightly lower number of iterations. This is logical given that restrictions are not used in the optimization, and this favours convergence. The main problem is that the obtained solution is not an assembly configuration although it can be used to obtain an estimation of the error (the deformation energy).

The next problem is based on the same double butterfly linkage although the requirement in this case is significantly far from the initial position.

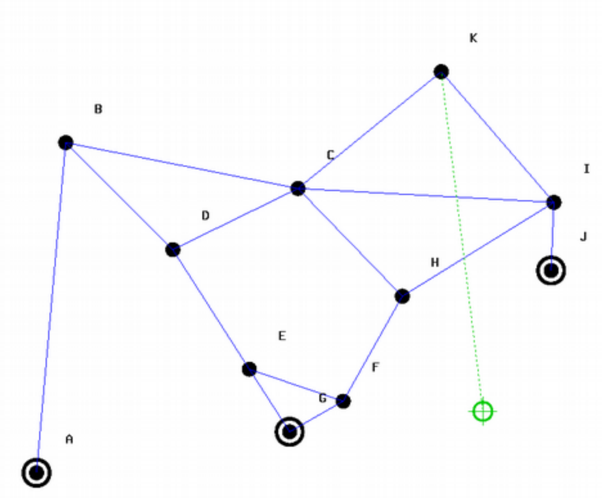

Figure 13: Double Butterfly with a requirement far from the initial configuration

The results obtained with the minimal distance function are shown in figure 14:

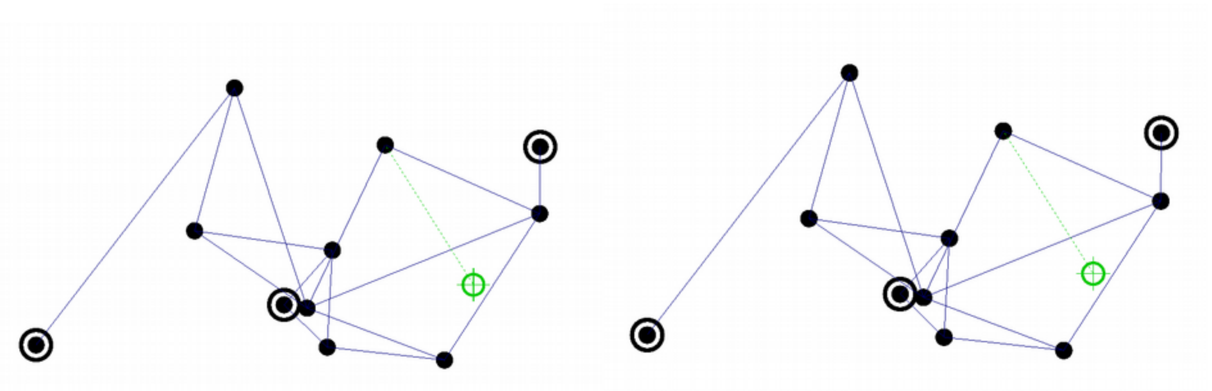

Figure 14: Solutions. Left: by using simple length restriction formulation. Right: by using lengths raised to the power of two

In this case, the results are equal. This is not uncommon. The convergence graphs are shown in the figure below:

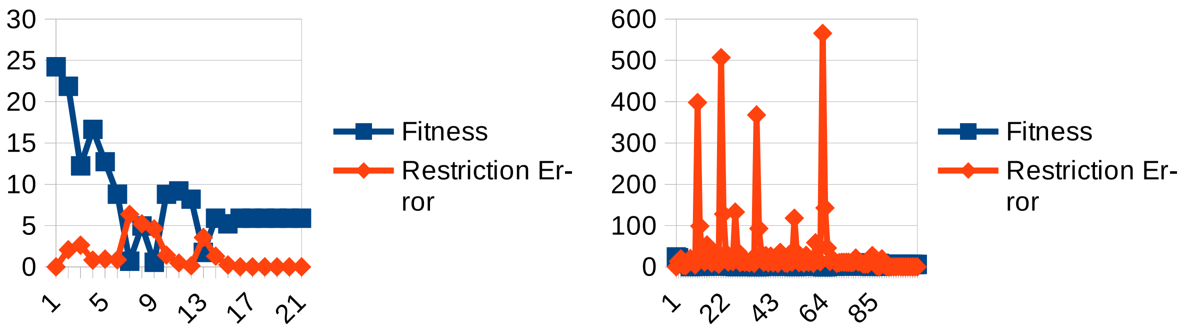

Figure 15: Convergence graphs. Left: Restrictions formulated as simple lengths. Right: Restrictions formulated as lengths raised to the power of two.

Although both algorithms lead to the same result, the second algorithm involves a long computation period. Although this difference is significantly rare (in the first tests, the effect is unnoticed), it should be considered since it increases computational cost from 1 ms to 10 ms. It only occurs with the aforementioned difference in requirements far from the initial solution, and thus it is not a significant problem for synthesis purposes. It appears that the simplification of the Hessian in the second restriction formulation can lead to small penalties in convergence.

The minimal distance result and its convergence graph is shown in figure 16:

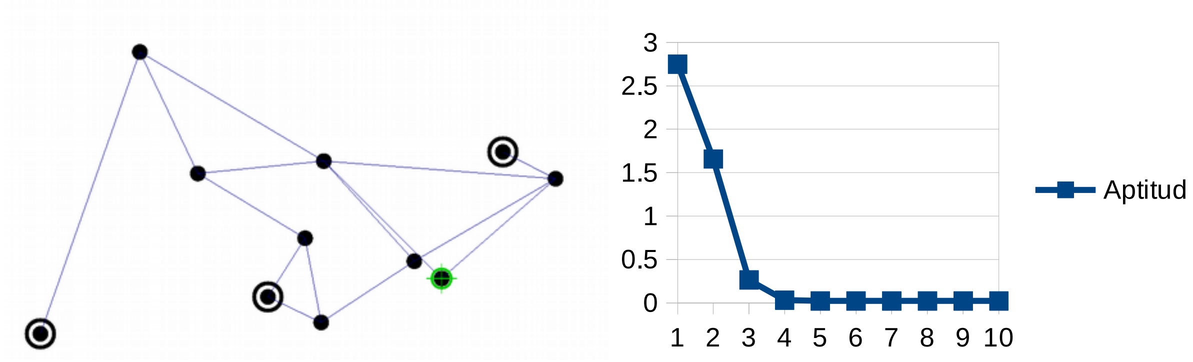

Figure 16: Solution obtained by using minimum deformation energy and convergence graph.

Here it is possible to observe the biggest disadvantage of the minimum deformation energy method. The algorithm can easily change the assembly configuration of the mechanism, thereby leading to an impossible solution. In the presented solution, the single element defined by nodes CHIK folds. This type of folding is overcome with specific formulations as those presented in [33] although all the assembly changes cannot be solved in this manner. The minimum distance method significantly avoids assembly changes albeit not all. This is because there is a significant change in configuration of the initial iteration when the minimum energy approach forces the nodes to their requirements and especially if the requirement is far from the initial solution. This is not always a disadvantage because it can occasionally lead to a configuration that is reachable without a change in the assembly configuration, and the minimum distance algorithm is unable to reach the same because it converges to another local solution. In any case, this mostly corresponds to a disadvantage than an advantage.

Minimizing the synthesis function

In order to apply the minimum distance function to the synthesis of mechanisms, it should be minimized. This can be performed in two significantly different ways. The first method is useful whenever a mechanism that nearly verifies the requirements or a similar set is known. This occurs in the packaging industry where the packaging mechanism of a machine should be adapted to another one with similar requirements. The other problem occurs when clues on the dimensions of the mechanism to be used are not available. In the first case, numerical optimization techniques including gradient descent or SQP are of the maximum interest. In the second case, heuristics are typically more useful. However, in the latter case, the use of a numerical technique to further improve the results is significantly interesting. In the study, an SQP method is used for two reasons. The first reason is that a distance based function was already proven as successful along with genetic algorithms (although it is significantly different than the approach presented here). The second is that a numerical method is of interest in both situations, and thus it is interesting to implement it before the heuristics. The SQP method matches the one used to optimize the minimum distance function. However, in this case, a finite difference approach is used to obtain the derivatives. The use of finite differences as opposed to a BFGS or similar quasi-Newton approach is because exact derivatives can be obtained and the use of a finite difference approach yields results that resemble the exact derivatives approach, and thus it serves to prove the utility of the method (although at a higher computational cost). Given the results obtained in the previous examples, the restrictions are modelled by using the length formulation. It is important to note that since a set of initial coordinates are used for the optimization, the system is typically be undefined, and thus it is necessary to use a method that can deal with this problem.

Examples of mechanism synthesis

As previously discussed, the obtained results are heavily dependent on the starting proposed solution. Given the significant number of local extrema of the function, a small change in the coordinates of the nodes or in the optimization algorithm can change the result. If the initial configuration is not adequate, the result always corresponds to a better mechanism. However, it is occasionally necessary to change the starting solution several times to obtain an adequate final result. As previously mentioned, the algorithm can improve a mechanism with a behaviour similar to that required. This does not mean that a good result far from the starting point is occasionally achieved. In the case in which a good initial guess is not available, the use of a heuristic method is potentially useful to retrieve a good estimate. The aforementioned starting vector can subsequently be improved with this algorithm. In any case, the use of heuristics is beyond the scope of this study. The computer used to perform the calculations for the mechanism synthesis is a XeonE5645@2.4 GHz. The code is programmed in a single thread configuration, and thus there the advantage of the multiprocessor characteristics of the computer are absent except for the advantage of the other cores that perform the operative system duties. The first example is based on a simple fourbar. The path to be described is not associated with that traced by the mechanism in the initial configuration as shown in in figure 17. The fixed nodes coordinates are not part of the optimization, and thus their coordinates are restricted by using linear restrictions.

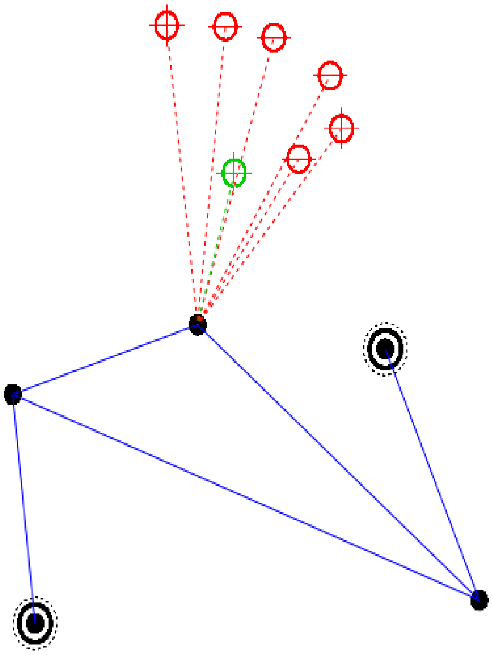

Figure 17: Synthesis of a fourbar, unprescribed timing

The problem corresponds to an unprescribed timing type because the input link position for each precision point is not defined. The problem includes 5 independent variables and 7 coordinates, and thus it does not possess an exact solution. It is important to state that the mechanism exhibits a degree of freedom, and thus only 5 variables really define the mechanism although there are 6 variables. This is important because it implies that the optimization always leads to an underdetermined system of equations. Thus, the solver should be able to deal with the same. The resulting linkage along with the convergence graph of the synthesis is shown in figure 18. The total time of the calculation is less than 3 s.

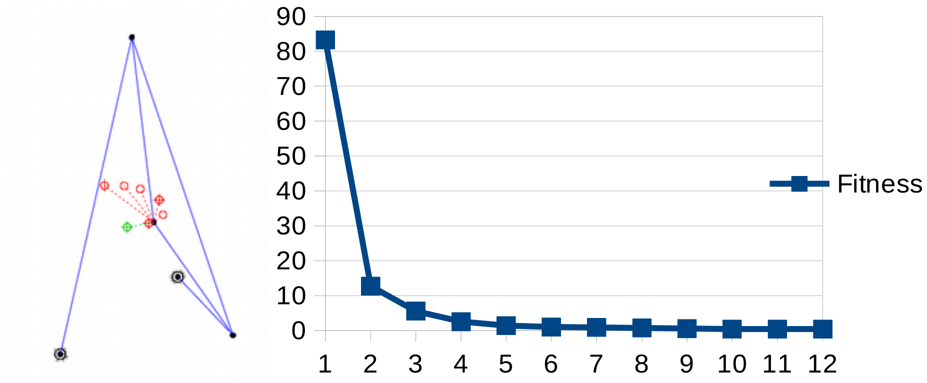

Figure 18: Minimum Distance result and convergence graph

The minimum distance positions of the mechanism to each of the precision points are shown in figure 19. It is interesting to investigate the relative performance of the algorithm when compared to the formulation of the minimal distance problem based on central differences. First, the obtained results need not coincide. Second, for the problem, the obtained result with central differences is obtained at a similar iteration count at a cost corresponding to a factor of 10 to that of a fourbar (approximately 30 s). More complex mechanisms further increase the factor.

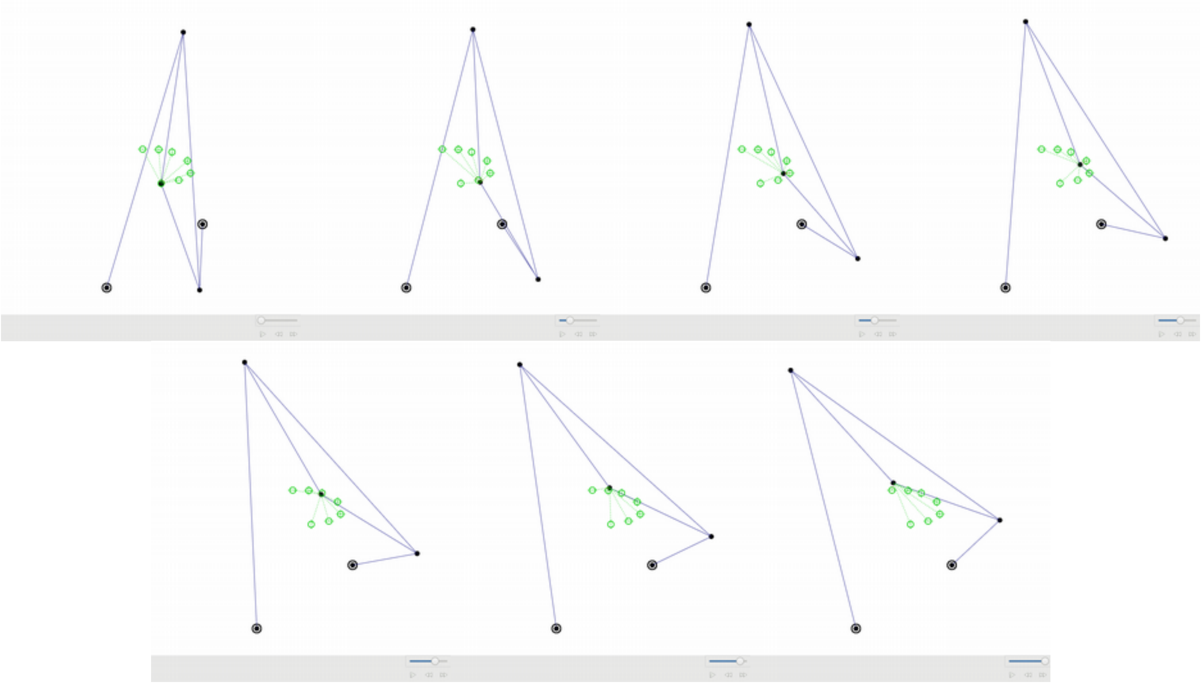

Figure 19: Minimum distance positions of the solution

The use of fourbars as a basis for examples of optimizations is widespread. This provides an idea of the complexity of the problem of kinematical synthesis. The next example is significantly more complex. It is a path generation performed by a double butterfly mechanism. The problem is described in figure 20. The difficulty of the mechanism is that it cannot be decomposed in simple loops. Furthermore, it again corresponds to a non-prescribed timing problem. In this case, the fixed node coordinates are allowed to change. The final obtained result is shown in figure 21. The minimum distance positions are shown in figure 22. The evolution of the fitness is shown in figure 23. The convergence is achieved in approximately 14 iterations. Initially, the convergence is fast. However, at the later stages, the central differences approach for the derivatives leads to a slowdown. The use of analytically formulated derivatives can avoid the aforementioned problem.

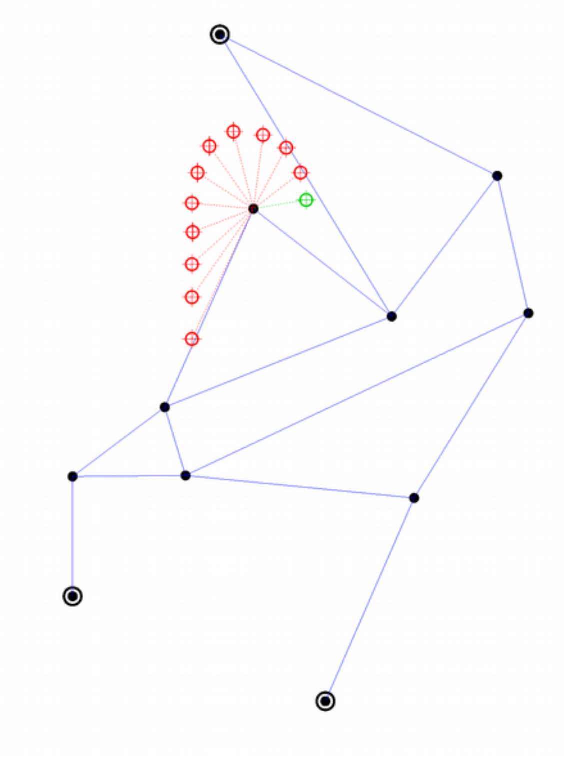

Figure 20: Path Synthesis for a double butterfly linkage

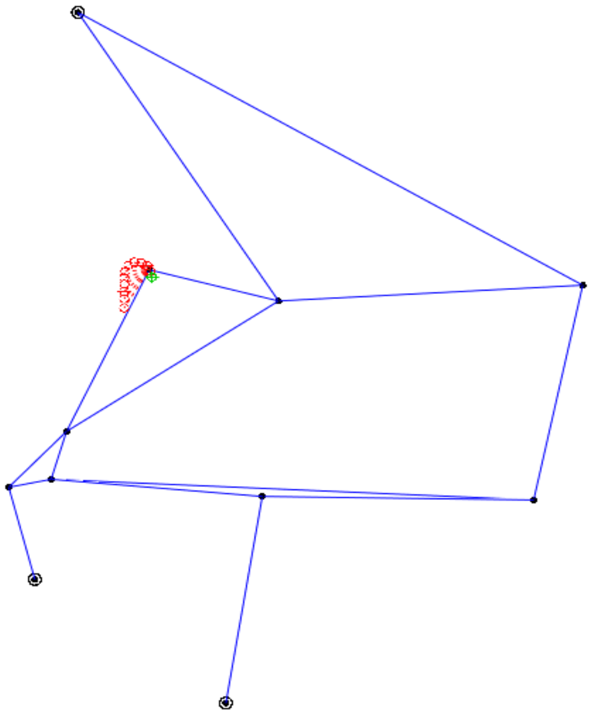

Figure 21: Obtained result

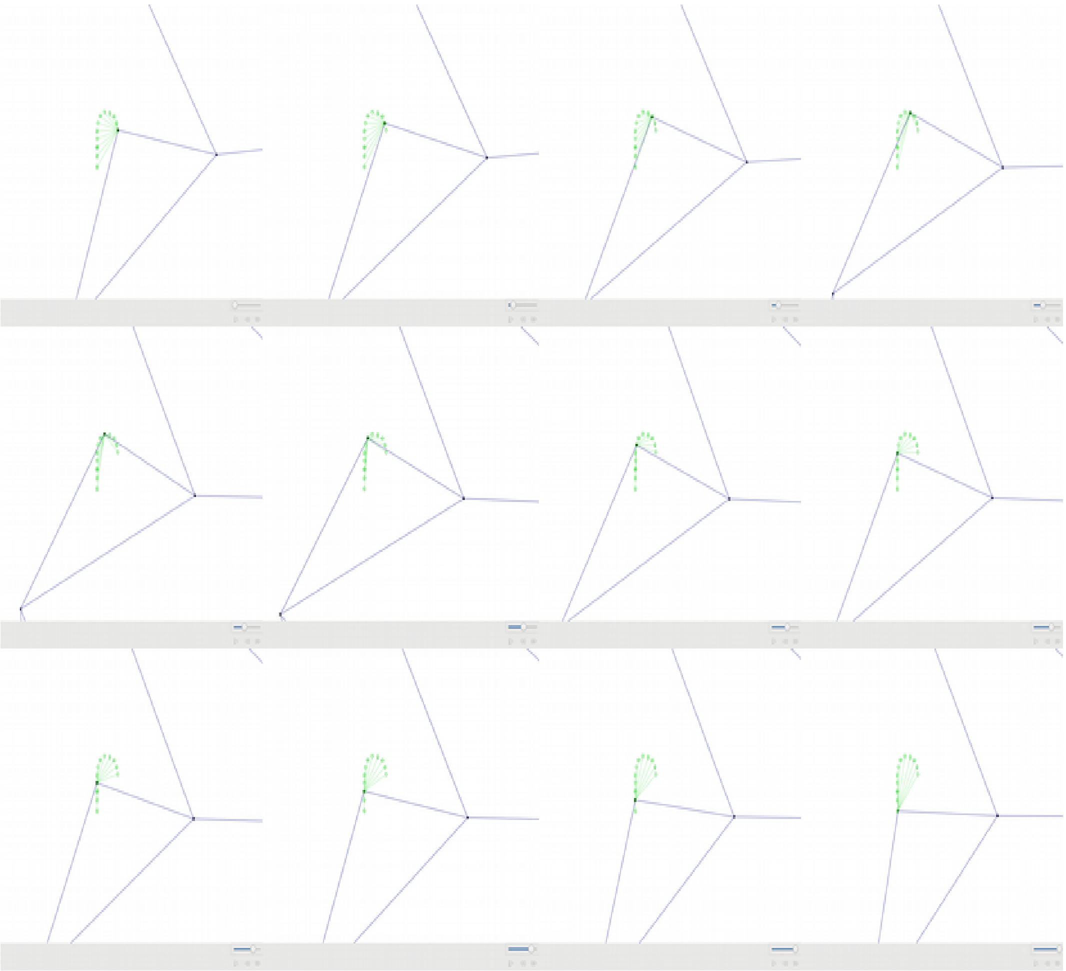

Figure 22: Minimum distance positions in the solution

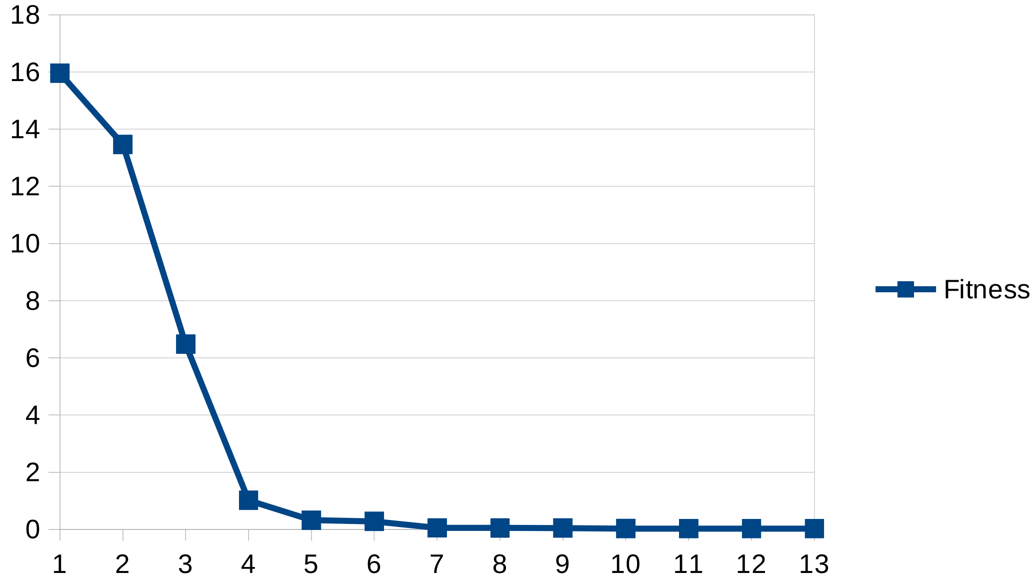

Figure 23: Fitness evolution

The final example corresponds to a function generation performed with a double butterfly. It also includes a small set of precision points (only 4), and thus it is further differentiated from the previous examples wherein several precision points are introduced. The upper fixed node (J) is allowed to change location while the other two node locations (nodes A and B) are not included in the optimization. Furthermore, the lengths of trusses AC and BD are fixed. This is necessary to ensure that the optimization process delivers zero lengths for the aforementioned trusses. The problem is described in figure 24.

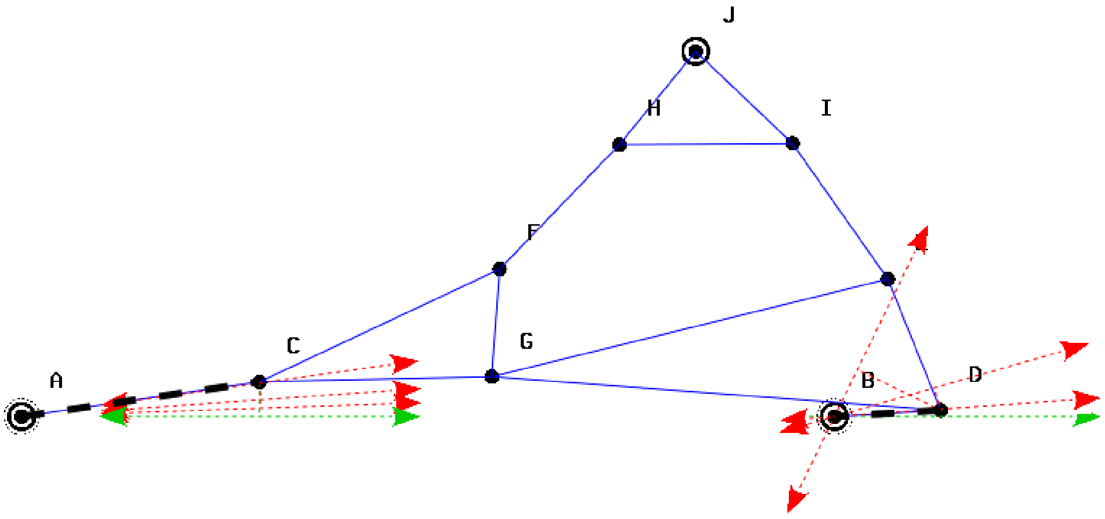

Figure 24: Function synthesis in a double butterfly mechanism

The resulting mechanism along with the convergence graph is shown in figure 25. Convergence was achieved in approximately 30 iterations with a total time of approximately 35 s. Typically, a decrease in the precision points increases the number of iterations required for convergence. This is potentially because the problem is less stiff, and thus it exhibits a smaller number of local extrema. This typically implies that the initial solution is also typically farther from the local optima that it reaches.

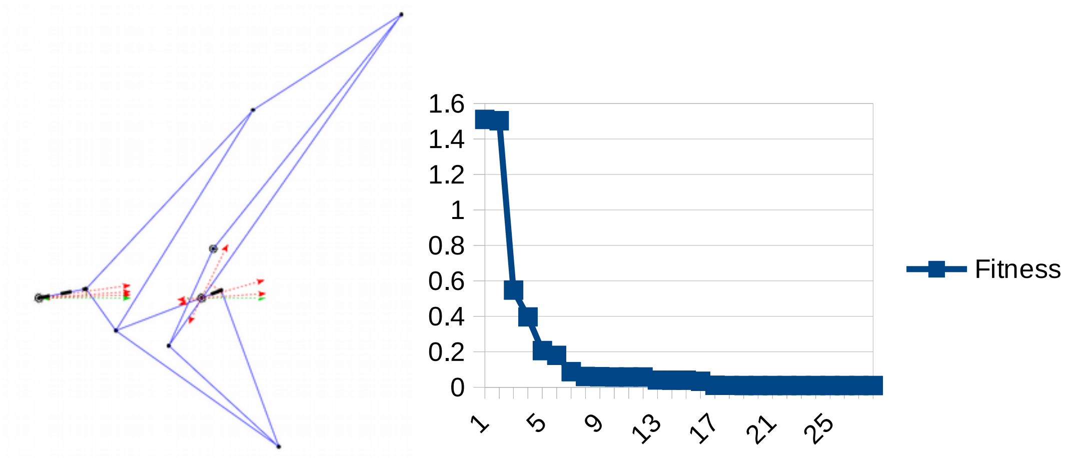

Figure 25: Obtained solution and convergence graph

Minimum distance configurations are shown in figure 26:

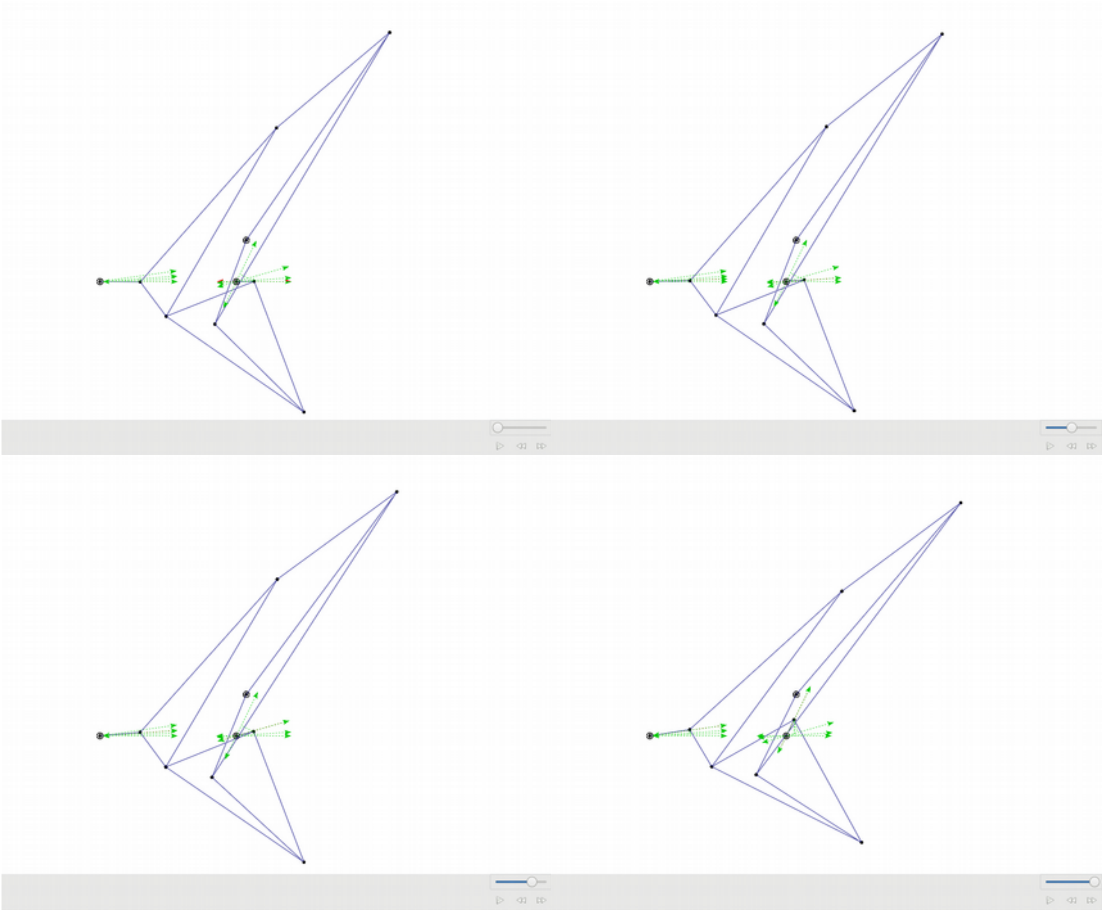

Figure 26: Minimum distance positions for each of the precision points

Conclusions and future work

In the study, a new error function for the optimization of mechanisms is presented. The main advantage of the error function is its flexibility. It allows unprescribed and prescribed timing problems. When compared with methods including minimum deformation energy, it exhibits the same advantages although it avoids the problem of low stiffness mechanisms. The minimization of the minimum distance error function with an SQP method is presented, and it is demonstrated that it accurately converges. The analytic derivatives of the requisites of node to point and node to line are obtained. This significantly reduces the computational cost of the solution of the minimum distance problem, and thus the synthesis is reached at a reasonable cost. The minimization of the synthesis error function is approached with the same SQP method although derivatives in this case are obtained in a central differences approach. The use of this type of numerical optimization improves an initial solution and is of interest in industrial areas where mechanisms with different although similar requisites are required such as packaging. The synthesis method indicated that it can deal with simple mechanisms, such as fourbars, and also with more complicated linkages such as the double butterfly. Future developments include the introduction of increased requirements such as node to node distance. The development of analytic derivatives should also improve the performance of the algorithm. The application of heuristics to the synthesis problem is of special interest such as genetic algorithms. The aforementioned approach is already demonstrated as successful when it is applied to an approximate minimum distance function, and thus it is potentially useful with respect to the exact minimum distance function examined in the study. This leads to a good starting point that subsequently be numerically improved with the presented SQP synthesis approach.

Acknowledgements

The authors acknowledge direct or indirect economic support provided by the Investigation Groups recognized by the Basque Government under section IT 947-16, and the Spanish Ministry of Economy and competitiveness through the project DPI2016-80372-R (AEI/FEDER, UE)

References

[1]A. Svoboda, Computing mechanisms and linkages, Dover Publ., New York, 1965.

[2] J.T. Kimbrell, Graphical synthesis of a four-bar mechanism, Mechanisms and Machine Theory, Vol 19, no1, 1984, pp 45-50.

[3]K.J. Waldron, Graphical solution of the branch and order problems of linkage synthesis for multiple separated positions, Journal of Engineering for Industry, 1977, pp

591-597.

[4]A. B. Kempe, On a general method of describing plane curves of the nth degree by linkwork, Proc. London Math. Soc. S1–7 (1876), 213–216.

[5] F. Freudenstein. “Approximate Synthesis of Four-Bar Linkages”. Trans. ASME 77, pages 853-861, 1955.

[6] C.W. Wampler, A.P. Morgan, A.J. Sommese, Complete solution of the nine-point path synthesis problem for four bar linkages, ASME Journal of Mechanical Design. Vol 112, 1990, pp. 59-68.

[7] T.S. Todorov, Synthesis of four-bar mechanisms by Freudenstein-Chebyshev, Mechanisms and Machine Theory, Vol 37, 2002, pp 1505-1512.

[8] P. Larochelle, Synthesis of planar mechanisms for pick and place tasks with guiding positions, Journal of Mechanisms and Robotics, vol.7, Paper No. 031009, 10 pages, 2015.

[9] Z. Li, J. Schicho, and H.-P. Schröcker. Kempe's universality theorem for rational space curves. Found. Comput. Math., 2017.

[10] Fernández-Bustos I, Aguirrebeitia J, Avilés R, Angulo C. (2005) Kinematical synthesis of 1-dof mechanisms by using finite elements and genetic algorithms. Finite Elem Anal Des 41(15):1441–1463 2006.

[11] Kunjur A, Krishnamurty S (1997) Genetic algorithms in mechanism synthesis. J Appl Mech Robot 4(2):18–24

[12] Cabrera J, Simon A, Prado M (2002) Optimal synthesis of mechanisms with genetic algorithms. Mech Mach Theory 37(10):1165–1177

[13] Sonpimple M, Bapat P, Modak J, Pimpalapure S (2010) A novel hybrid simulated annealing-direct search algorithm for optimum synthesis of function generation problem. J Theor Appl Inf Technol 11:16–24

[14] Tlale S, Modungwa D, Twala B (2012) Optimization approaches applied in dimensional synthesis of parallel mechanisms. Trans Control Mech Syst 1(2):57–64

[15] Lanni C, Saramago S, Ceccarelli M (2002) Optimal design of 3R manipulators by by using classical techniques and simulated annealing. J Braz Soc Mech Sci 24(4):293–301

[16] Oliva JC, Goodman ED (2009) Evolutionary search and convertible agents for the simultaneous type and dimensional synthesis of planar mechanisms. In: Proceedings of the 11th annual conference on genetic and evolutionary computation. ACM, pp 1577–1584

[17] M.A. Laribi, A. Milka, L. Romdhane, S. Zeghloul. A combined genetic algorithm-fuzzy logic method (GA-FL) in mechanism synthesis. Mechanism and Machine Theory, V39, N7, July 2004, pp 717-735

[18] J.R. McGarva, Rapid search and selection of path generating mechanisms from a library, Mechanisms and Machine Theory, vol 29, no2, 1994, pp 223-235.

[19] Zhang K (2008) Synthesis of a hybrid five-bar mechanism with particle swarm optimization algorithm. In: Sun F, Zhang J, Tan Y, Cao J, Yu W (eds) Advances in neural networks—ISNN 2008. ISNN 2008. Lecture notes in computer science, vol 5263. Springer, Berlin, Heidelberg, pp 873–882

[20] Alinia S, Daniali H, Ghadimi M, Kaliji H (2012) Optimal synthesis of planar mechanisms with PSO algorithms. World Appl Sci J 18(2):268–273

[21] Sedlaczek K, Eberhard P (2007) Augmented lagrangian particle swarm optimization in mechanism design. J Syst Des Dyn 1(3):410–421

[22] Smaili AA, Diab NA, Atallah NA (2005) Optimum synthesis of mechanisms by using tabu-gradient search algorithm. J Mech Des 127(5):917–923

[23] Suh CH, Radcliffe CW (1978) Kinematics and mechanisms design. Wiley, Hoboken

[24]R. Avilés, M.B. Ajuria, J. García de Jalón. A fairly general method for optimum synthesis of planar mechanisms. Mechanism and Machine Theory, 20 (4) (1985), pp. 321-328

[25] J.A. Alba, M. Doblaré, L. Gracia, A simple method for the synthesis of 2D and 3D mechanisms with kinematic constraints, Mechanisms and Machine Theory, vol 35, 2000, pp 645-674.

[26] R. Sancibrian, F. Viadero, P. García, A. Fernández, Gradient-based optimization of path synthesis problems in planar mechanisms, Mechanism and Machine Theory, Volume 39, Issue 8, 2004, Pages 839-856

[27] Ana De-Juan, Ramon Sancibrian, Pablo Garcia, Fernando Viadero, Miguel Iglesias, Alfonso Fernandez. Journal of Advanced Mechanical Design, Systems, and Manufacturing. Volume 6, Issue 4, 2012 Pages 472-483

[28] G. A. Hassaan, Optimal kinematic synthesis of planar mechanisms, Part I: Offset crank-slider mechanism, International Journal of Computer Techniques, vol.2, issue 2, pp.151-157, 2015.

[29] I. Fernández de Bustos, V. García Marina, G. Urkullu. "Solving the minimum distance problem for the synthesis of mechanisms" In New Trends in Mechanism and Machine Science: Theory and Industrial Applications. Mechanisms and Machine Science, 43. Springer, 2017. ISSN 2211-0984

[30] Avilés, R., Vallejo, J., Ajuria, G. Agirrebeitia, J. Second-order methods for the optimum synthesis of multibody systems. Struct Multidisc Optim (2000) 19: 192. https://doi.org/10.1007/s001580050102

[31] García-Marina, V., Fernández de Bustos, I., Urkullu, G., Abasolo, G. By using nodal coordinates as variables for the dimensional synthesis of mechanisms. Meccanica (2017). https://doi.org/10.1007/s11012-017-0799-6

[32] I. Fernández de Bustos, J. Aguirrebeitia, G. Ajuria, R. Ansola. An alternative full-pivoting algorithm for the factorization of indefinite symmetric matrices. Journal of Computational and Applied Mathematics 274 (2015) 44-57

[33] Fernández-Bustos I, Aguirrebeitia G, Ajuria R, Angulo C. (2005) A new finite element to represent prismatic joint constraints in mechanisms Finite Elem Anal Des 43(2006):36–50