

Optimal synthesis of planar mechanisms via an extensible-link approach

Jean-François Collard · Pierre Duysinx · Paul Fiset

Received: 9 May 2006 / Revised: 12 September 2006 / Accepted: 25 February 2010 / Published online: 28 March 2010
© Springer-Verlag 2010

Abstract This paper presents a novel approach to optimize the design of planar mechanisms with revolute joints for function generation or path synthesis. The proposed method is based on the use of an extensible-link mechanism model whose strain energy is minimized to find the optimal rigid design. This enables us to get rid of assembly constraints and the use of natural coordinates makes the objective function simpler. Two optimization strategies are developed and then discussed. The first one relies on alternate optimizations of design parameters and point coordinates. The second one uses multiple partial syntheses as starting point for a full synthesis process. The question of finding the global optimum is also addressed and developed. A simple algorithm is proposed to find several local optima among which the designer may choose the best one taking other criteria into account (e.g. stiffness, collision, size, . . .). Three applications are presented to illustrate the strategies while mentioning their limits.

Keywords Optimal synthesis · Natural coordinates · Closed-loop mechanisms

Part of this paper has been presented at the Sixth WCSMO, Rio, June 2005.

J.-F. Collard (✉) · P. Fiset
Centre for Research in Mechatronics,
Université catholique de Louvain (UCL),
Louvain-la-Neuve, Belgium
e-mail: jf.collard@uclouvain.be

P. Duysinx
Department of Mechanics and Aerospace,
Université de Liège (ULg),
Liège, Belgium

1 Introduction

The optimization of complex multibody systems is currently very topical considering the increasing development of computer resources. This statement remains particularly true for closed-loop mechanisms whose assembly constraints need to be satisfied before any mechanical analysis. They thus require a special attention when developing the optimization process strategy. A few solutions have been proposed to deal with these systems. For example, in Collard et al. (2005a), the authors suggested to properly penalize the objective function using the conditioning of the assembly constraints' Jacobian matrix. This extends the parameter space and leads to an enlarged scope of the objective function but the problem may remain difficult to optimize. The problem has also been extensively studied by Minnaar et al. (2001) who proposed to minimize the residual of assembly constraint equations rather than equate them to zero. Another well-known approach in path synthesis is to virtually deform the mechanism subject to a perfect following of the desired path as in Vallejo et al. (1995), Jiménez et al. (1997), Alba et al. (2000) and Avilés et al. (2000). From this point of view, the path-following objective becomes an optimization constraint while the deformation energy is the actual objective to be minimized. Therefore, the mechanism assembles at best possible each time the optimization process computes the objective function. This approach is adopted in this work and only planar mechanisms with revolute joints are considered.

In optimal design synthesis, a second issue consists in the choice of the formalism to describe the geometry of the mechanism. Among the different possibilities, one can mention the common use of relative coordinates in real form (Mariappan and Krishnamurty 1996; Cabrera et al. 2002; Hansen 2002; Laribi et al. 2004) or in complex form

(Sancibrian et al. 2004). This formalism has the advantage of limiting the number of assembly constraints but introduces trigonometric functions involving angular variables: it enhances the non-linearity of the problem and makes the optimization more complex. The use of natural—or point—coordinates is also widespread (Jiménez et al. 1997; Alba et al. 2000; Avilés et al. 2000; Lio et al. 2000; Jensen and Hansen 2003). In comparison with relative coordinates, natural coordinates involve additional algebraic constraints. However, these equations only contain linear and/or distance functions. This coordinate system is thus well suited to the use of gradient-based optimization techniques such as least squares methods.

The proposed method tries to combine these two features: deformable mechanisms and natural coordinates. The first one enables to solve the problem of non-assembly while the second one greatly simplifies the type of objective function. The optimization strategy is based on the minimization of the deformation energy over the followed path, and a subsequent update of the mean lengths as suggested in Hansen (2002). This minimization-update sequence is repeated until convergence when the total deformation energy no longer varies or is sufficiently small. The method has a rather slow convergence rate but produces a monotonic decrease, and may sometimes fail to find the global optimum. To improve the convergence rate, the technique is combined with a full synthesis approach as proposed by Alba et al. (2000). A similar combination of two techniques has also been described by Avilés et al. (2000) from a finite elements point of view. Nevertheless, in our case, the main difference results from the dependence between the two techniques: the first one is used here as starting point for the full synthesis, while in Avilés et al. (2000) the latter relies fully on the first technique (Minnaar et al. 2001).

Furthermore, this improvement allowed us to highlight an important issue in mechanism optimization: reaching different local optima starting from different initial parameters. The choice of the optimal mechanism among these local optima relies on other design constraints which are difficult to compute and not taken into account in the original problem. Moreover, it is also shown that the use of a standard genetic algorithm does not necessarily permit to reach the global optimum. Since it is interesting for the designer to keep and compare some of the best mechanisms (i.e. local optima), an exploration strategy of the design space is proposed to “unearth” most of the possible optima.

Various kinds of requirements may be encountered in dimensional mechanisms synthesis: path or function generation, body guidance, or mixed problems. Most applications concern path synthesis problems (Mariappan and Krishnamurty 1996; Minnaar et al. 2001; Zhou and Cheung 2001; Cabrera et al. 2002; Hansen 2002; Marín and González 2003; Laribi et al. 2004; Sancibrian et al.

2004; Shiakolas et al. 2005) and the four-bar mechanism will constitute a simple running example in the following. More realistic applications of function-generation synthesis will also be given based on the Ackerman steering linkage problem: a four-bar and then a six-bar synthesis. These last design syntheses were suggested by Simionescu et al. (2000), Pramanik 2002, Simionescu and Beale (2002). The proposed approach can easily be extended to problems of body guidance or mixed problems.

The paper is organized as follows. In Section 2, the general optimization problem is modeled and formulated: the objective function is built and the sensitivity analysis is performed. In Section 3, the optimization strategy is developed applying two different methods to the path synthesis of a four-bar mechanism as illustrative example. Section 4 then presents a more realistic application of function generation synthesis for the four-bar Ackerman steering linkage. Section 5 deals with the question of finding the global optimum followed by a new method to explore the design space. Before some conclusions and prospects in Section 7, a more complete application to optimize a six-bar steering linkage is presented so as to illustrate the concepts of Section 6.

2 Problem formulation

The problem formulation is divided into two subsections: first, the cost function is evolved and then the corresponding sensitivity analysis is carried out.

2.1 Objective function

Let us consider the well-known planar example of a four-bar mechanism which must follow a desired path (see dotted line in Fig. 1a). In order to make the mechanism exactly follow the given path, the four-bar is modeled with extensible links which replace the rigid bars and triangle by five springs with stiffnesses k_j and natural lengths l_j , $j = 1 \dots 5$ (see Fig. 1b).

The desired path is discretized into N points, leading to N different configurations of the mechanism. When it moves, the various points $P_0 \dots P_4$ composing the mechanism have different behaviors: P_0 and P_4 stay fixed to the ground, P_3 exactly follows the N points composing the path and P_1 and P_2 are free so as to reach equilibrium. All these points can thus be arranged into three groups:

- the *static* points P_0, P_4 ;
- the *tracking* point P_3 ;
- the *floating* points P_1, P_2 .

Their absolute coordinates are saved respectively in the following vectors: \mathbf{s} , \mathbf{t} and \mathbf{f} . As the tracking point and

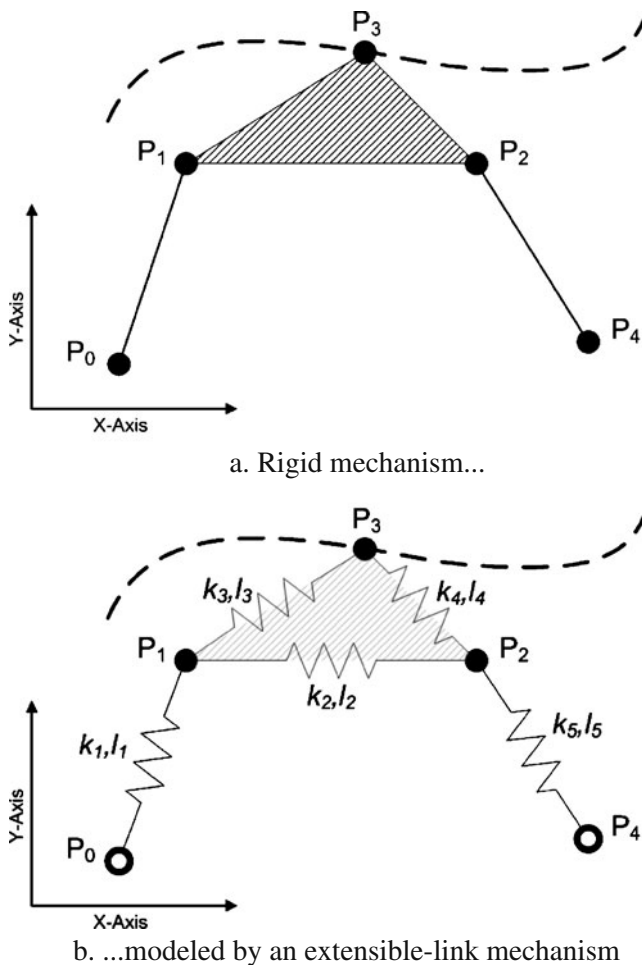


Fig. 1 Model adaptation of four-bar mechanism for path synthesis

the floating points may have different coordinates for each configuration, \mathbf{t} and \mathbf{f} are referenced by the index i : \mathbf{t}_i and \mathbf{f}_i , $i = 1 \dots N$.

Grouping the natural lengths l_j in the column vector \mathbf{l} and the stiffness parameters k_j on the diagonal of the stiffness matrix \mathbf{K} , we define the total strain energy as a scalar cost function:

$$E(\mathbf{s}, \mathbf{t}_1, \dots, \mathbf{t}_N, \mathbf{f}_1, \dots, \mathbf{f}_N, \mathbf{l}, \mathbf{K}) = \frac{1}{2} \sum_{i=1}^N (\mathbf{d}_i - \mathbf{l})^T \mathbf{K} (\mathbf{d}_i - \mathbf{l}), \tag{1}$$

where index i stands for the i th configuration of the mechanism and \mathbf{d}_i is a column vector containing the five distances d_j^i between each couple of linked points. At the outset, the only known parameters are the $2N * 2$ coordinates of the floating points. The stiffness parameters k_j may be chosen by the user. They play the role of weights in the sum of all the contributions to the total energy. Making a bar stiffer increases its relative importance in the cost function but this

possibility is not used in the following. After these considerations, the optimization problem is stated as follows:

$$\min_{\mathbf{s}, \mathbf{f}_1, \dots, \mathbf{f}_N, \mathbf{l}} \frac{1}{2} \sum_{i=1}^N [\mathbf{d}(\mathbf{s}, \mathbf{t}_i, \mathbf{f}_i) - \mathbf{l}]^T \mathbf{K} [\mathbf{d}(\mathbf{s}, \mathbf{t}_i, \mathbf{f}_i) - \mathbf{l}], \tag{2}$$

where the actual design parameters are \mathbf{s} and \mathbf{l} . This constitutes an obvious non-linear least squares optimization problem. Defined in that way, the problem remains difficult to solve. Therefore, two proposals are made below to improve the homogeneity of the problem and to avoid multiple triangle configurations.

First, let us note that the actual design parameters, the static point coordinates \mathbf{s} and the natural length \mathbf{l} , appear in different terms of the cost function. This makes the cost function differently sensitive to the two of them. We propose to transform all static point coordinates into natural lengths of two springs (see Fig. 2). In this way, a new floating point is inserted into \mathbf{f} , vector \mathbf{s} is appended to vector \mathbf{l} and two new stiffness parameters are added to the diagonal of matrix \mathbf{K} . Note that the corresponding functions $d(\mathbf{s}, \mathbf{t}_i, \mathbf{f}_i)$ actually become the two coordinate values which are not always positive: this introduces so-called *oriented* springs according to the sign of their natural lengths. But this has no consequence on the energy formulation (1). For example, replacing static point (x_0, y_0) of Fig. 2 would create two additional contributions to the total energy: $\frac{1}{2}k_x(x - x_0)^2$ and $\frac{1}{2}k_y(y - y_0)^2$. Thanks to this transformation, all the design parameters may be grouped into the same vector \mathbf{l} .

The second proposal relates to the three springs composing the triangle P_1, P_2, P_3 . Fixing the points P_1 and P_2 , two stable positions remain for P_3 : above or below the $P_1 - P_2$ line. To remove any ambiguity, the use of *oriented* springs (see above) is proposed to unequivocally locate P_3 with respect to P_1 and P_2 . Thus, the two springs $P_1 - P_3$ and $P_2 - P_3$ are replaced by two *orthogonal oriented* springs as shown in Fig. 3. In this example, the contributions of springs 3 and 4 are replaced by: $\frac{1}{2}k_a(a - a_0)^2$ and $\frac{1}{2}k_b(b - b_0)^2$, with $a = \frac{\vec{P_1 P_2} \cdot \vec{P_1 P_3}}{\|P_1 P_2\|}$ and $b = \frac{\vec{P_1 P_2} \times \vec{P_1 P_3}}{\|P_1 P_2\|} \cdot \hat{z}$.

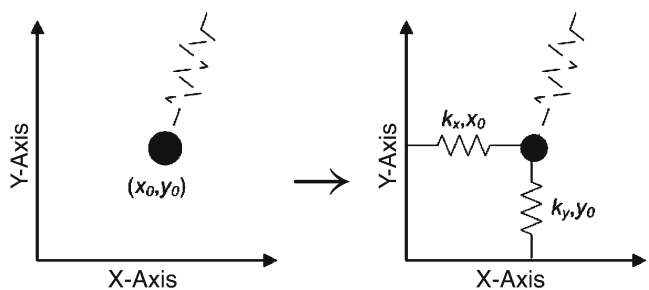


Fig. 2 New model of static points

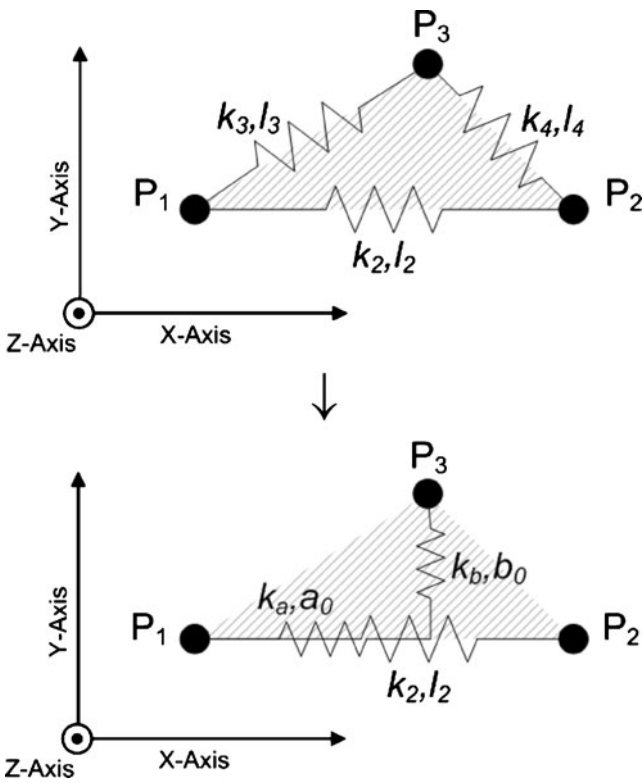


Fig. 3 New model of triangle element

Finally, taking both proposals into account, the new cost function (1) is:

$$E(t_1, \dots, t_N, f_1, \dots, f_N, \tilde{\mathbf{I}}, \tilde{\mathbf{K}}) = \frac{1}{2} \sum_{i=1}^N (\tilde{\mathbf{d}}_i - \tilde{\mathbf{I}})^T \tilde{\mathbf{K}} (\tilde{\mathbf{d}}_i - \tilde{\mathbf{I}}), \tag{3}$$

leading to the following rearranged optimization problem:

$$\min_{f_1, \dots, f_N, \tilde{\mathbf{I}}} \frac{1}{2} \sum_{i=1}^N [\tilde{\mathbf{d}}(t_i, f_i) - \tilde{\mathbf{I}}]^T \tilde{\mathbf{K}} [\tilde{\mathbf{d}}(t_i, f_i) - \tilde{\mathbf{I}}], \tag{4}$$

where the tilde symbol stands for the two modifications described above. A corresponding configuration of the four-bar model is illustrated in Fig. 4. Note the size of the various vectors for this simple four-bar example: nine components in $\tilde{\mathbf{I}}$, eight in \mathbf{f}_i and two in \mathbf{t}_i . Therefore, for the N configurations, $9 + 8 * N$ optimization variables have to be taken into account for problem (4), which may be a lot.

2.2 Sensitivity analysis

In problem (4), two kinds of optimization variables are now considered: $\tilde{\mathbf{I}}$ and $\mathbf{f}_i, i = 1 \dots N$. The gradients of the

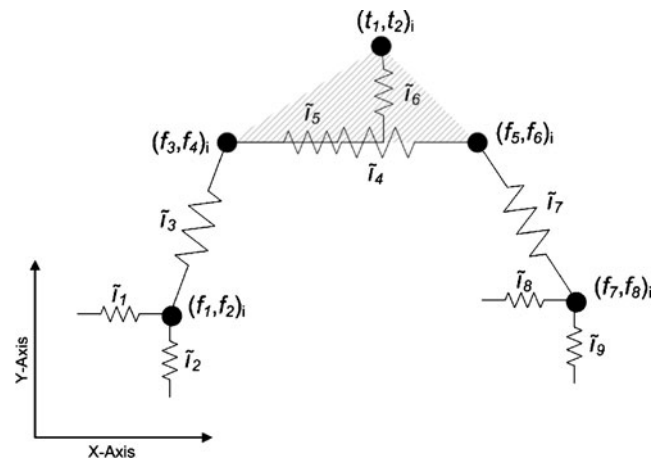


Fig. 4 One configuration of the four-bar model for path synthesis

total energy function with respect to these two vectors are given by:

$$\frac{\partial E}{\partial \mathbf{f}_i} = \frac{\partial \tilde{\mathbf{d}}^T(t_i, f_i)}{\partial \mathbf{f}_i} \tilde{\mathbf{K}} [\tilde{\mathbf{d}}(t_i, f_i) - \tilde{\mathbf{I}}] \tag{5}$$

$$\frac{\partial E}{\partial \tilde{\mathbf{I}}} = - \sum_{i=1}^N \tilde{\mathbf{K}} [\tilde{\mathbf{d}}(t_i, f_i) - \tilde{\mathbf{I}}] \tag{6}$$

Let us point out that (5) depends only on the i th configuration if the design parameters $\tilde{\mathbf{I}}$ are fixed. This means that the sensitivity of the total energy function with respect to the i th floating point coordinates vector is independent of the other configurations if the design parameters are known beforehand. This may greatly simplify the optimization problem and is the basis of the first optimization strategy described in the next section.

3 Optimization strategy

Two optimization strategies have been developed. The original idea relies on the alternation between multiple optimizations of the coordinates and the update of the design parameters. In the second strategy, this update part is replaced by an outer full optimization process.

3.1 Mean values update

This strategy is inspired by Hansen (2002) who proposed to minimize the deviation of each variable dimension over a cycle and to update the mean value after each cycle. The main difference here is the use of natural coordinates instead of relative coordinates which makes the objective function far more non-linear due to trigonometric functions. The corresponding algorithm flowchart is summarized in Fig. 5.

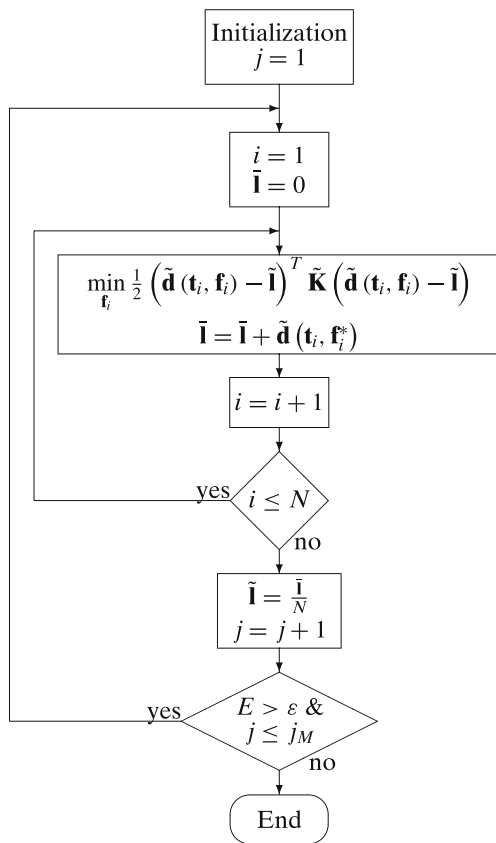


Fig. 5 “Mean update” algorithm flowchart

Starting from given values of the design parameters $\tilde{\mathbf{I}}$, the algorithm begins minimizing the total energy with respect to the \mathbf{f}_i . This is equivalent to solving N partial optimization problems because the \mathbf{f}_i are independent and $\tilde{\mathbf{I}}$ is constant:

$$\begin{aligned} & \min_{\mathbf{f}_1, \dots, \mathbf{f}_N} \frac{1}{2} \sum_{i=1}^N [\tilde{\mathbf{d}}(t_i, \mathbf{f}_i) - \tilde{\mathbf{I}}]^T \tilde{\mathbf{K}} [\tilde{\mathbf{d}}(t_i, \mathbf{f}_i) - \tilde{\mathbf{I}}] \\ \Leftrightarrow & \sum_{i=1}^N \min_{\mathbf{f}_i} \frac{1}{2} [\tilde{\mathbf{d}}(t_i, \mathbf{f}_i) - \tilde{\mathbf{I}}]^T \tilde{\mathbf{K}} [\tilde{\mathbf{d}}(t_i, \mathbf{f}_i) - \tilde{\mathbf{I}}] \end{aligned} \quad (7)$$

After one optimization cycle, the optimum distances $\tilde{\mathbf{d}}_i$ allow us to compute a new vector of design parameters by taking their mean values:

$$\tilde{\mathbf{I}} = \frac{1}{N} \sum_{i=1}^N \tilde{\mathbf{d}}(t_i, \mathbf{f}_i) \quad (8)$$

This is achieved in the second step of the algorithm (see Fig. 5). Assuming that the \mathbf{f}_i are frozen as if the bars were uncoupled (see Aviles et al. 2000), (8) is equivalent to a necessary condition for a local minimizer of function (3): finding the roots of (6). It can also be seen as a Newton step

since the cost function is quadratic in $\tilde{\mathbf{I}}$. At the end, the algorithm stops either when the total strain energy E decreases below a given floor value ε , or if the update counter j reaches a given maximum value j_M . In this last case, the algorithm does not converge to an optimum mechanism. Note that a classic Levenberg–Marquardt algorithm is used for the partial least squares optimization problems which only involve a few variables (e.g. eight point coordinates in the case of the four-bar).

As simple example, Fig. 6 presents a four-bar path synthesis excerpted from Hansen (2002). The initial, desired and resulting paths are shown in Fig. 6a. The evolution of the sum of the average deviations over a cycle is plotted in

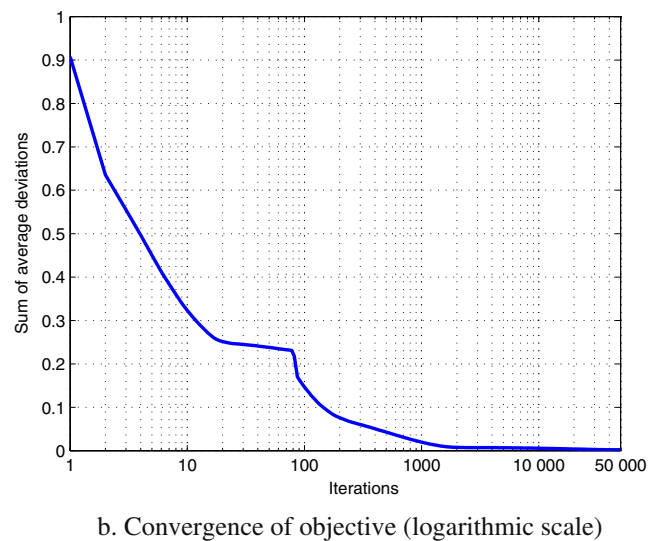
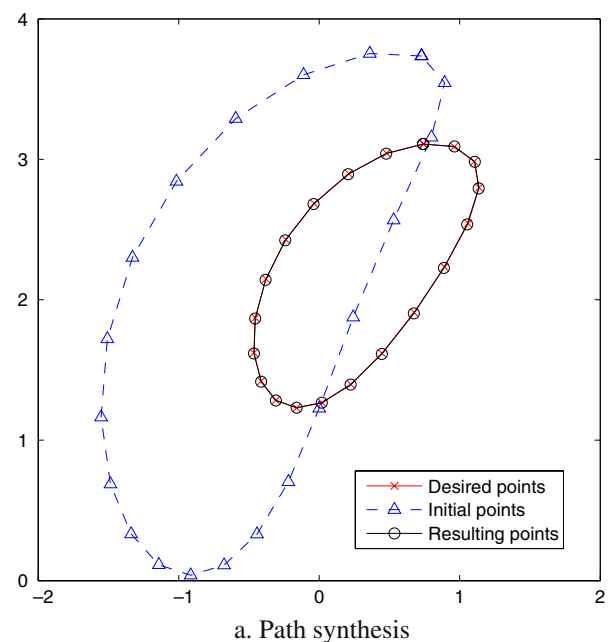


Fig. 6 Path synthesis example with the “mean update” algorithm

Fig. 6b. As can be observed, this algorithm performs well but has slow convergence. Moreover, as already mentioned, it may fail in some cases, e.g. when the minimum energy function exhibits a valley in the parameter space (e.g. see Section 4.2, Fig. 10). This justifies the new improvement developed in the next section.

3.2 Full synthesis

What is proposed here is to replace the natural lengths update by the full synthesis proposed by Alba et al. (2000) which means that all the parameters are optimized together. The new algorithm is represented in Fig. 7.

The beginning is the same as for the previous algorithm but the multiple partial optimizations (7) are performed only once. This first step provides us with hot starting points for the full optimization (4) which involves all the floating point coordinates and all the design parameters with all the contributions to the total energy.

As explained in section 2.2, the number of optimization parameters may increase rapidly (e.g. $9 + 8N = 329$ variables for the four-bar mechanism with $N = 40$ synthesis points) if the mechanism and/or the path get more complex. As the parameter space is larger, a more robust optimization algorithm is needed: for instance, the so-called “dog-leg” algorithm described in Powell (1970): this trust-region method is useful to solve systems of nonlinear

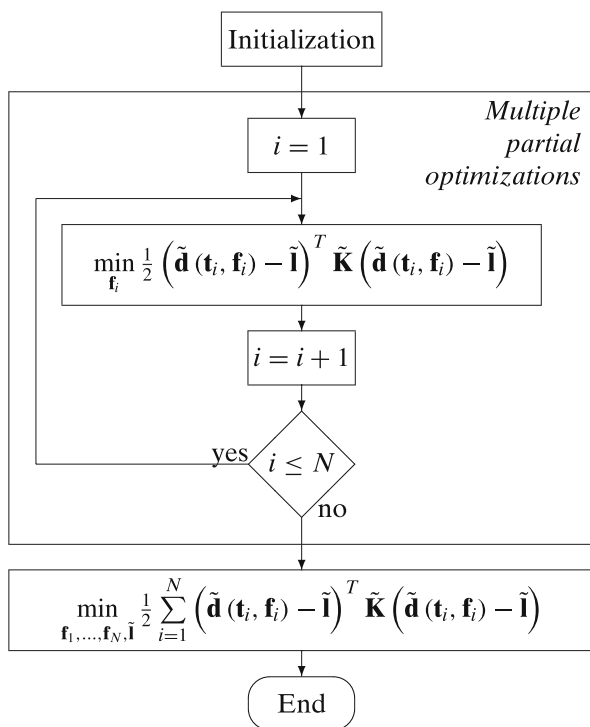


Fig. 7 Partial and total synthesis flowchart

equations. Applied to the four-bar steering synthesis, this combination of partial and full synthesis gives better results than the previous one as shown in the next section.

4 Application to four-bar steering linkage synthesis

This section presents an interesting application of function generation synthesis. The goal is to optimize steering linkages of vehicles. In the first subsection, the function to be generated is established from the Ackermann condition. Secondly, both previous optimization strategies are applied to a four-bar steering linkage and then compared.

4.1 The Ackermann condition

One of the main requirements of the steering mechanism of a vehicle is to provide the steerable wheels with correlated orientations, so that the intersection point of their axes lies on the extension of the rear wheel axis (point P in Fig. 8). As developed in Wong (2001), the Ackermann relation for a correct turn is:

$$\cot \delta_o - \cot \delta_i = \frac{l}{L}, \tag{9}$$

where δ_o and δ_i are the outer and inner wheel angles respectively, l is the wheel gauge and L the wheelbase of the vehicle. Only the wheel gauge-wheelbase ratio influences this Ackermann steering relation.

4.2 Four-bar steering linkage synthesis

The modeling of the four-bar steering linkage is set up according to the rules depicted in Section 2.1. To satisfy the Ackermann condition (9), the correlated path-following of the wheel centers are imposed while the inner wheel angle

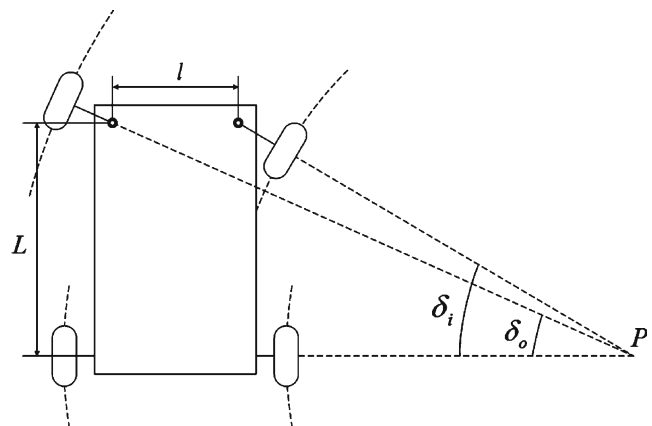


Fig. 8 The Ackermann condition

takes 40 different values between 0 and its maximum (see points P_1 and P_4 in Fig. 9). Also observe in the figure that the static points are not transformed into floating points because they do not belong to the design parameter set. These parameters consists in a priori three natural lengths: a , b and l . However, the problem symmetry reduces their number to only two— a and b —because $l = \|\overrightarrow{P_0P_5}\| - 2a$. Note that this is useful to visualize the cost function in two dimensions once minimized with respect to the f_i .

As previously mentioned, this simple example can reveal some of the drawbacks of both previous strategies. As the minimum energy function (solution of (7)) has a valley in the design parameter space, the evolution of the mean update algorithm may not converge to the local optimum (see Fig. 10). Starting from $a = 0.5m$ and $b = -0.5m$, the process stops after 3,250 iterations because the sum of average deviations no longer changes (see Fig. 11). Note that the function is penalized around the origin of the design space to avoid singular configurations of the mechanism. The objective function (3) is extended here as follows:

$$E_{ext} = \begin{cases} E & \text{if } \|\overrightarrow{P_0P_2}\| \geq d_{min} \\ \frac{1}{2} \left(\|\overrightarrow{P_0P_2}\| - d_{min} \right)^2 & \text{if } \|\overrightarrow{P_0P_2}\| < d_{min} \end{cases}, \quad (10)$$

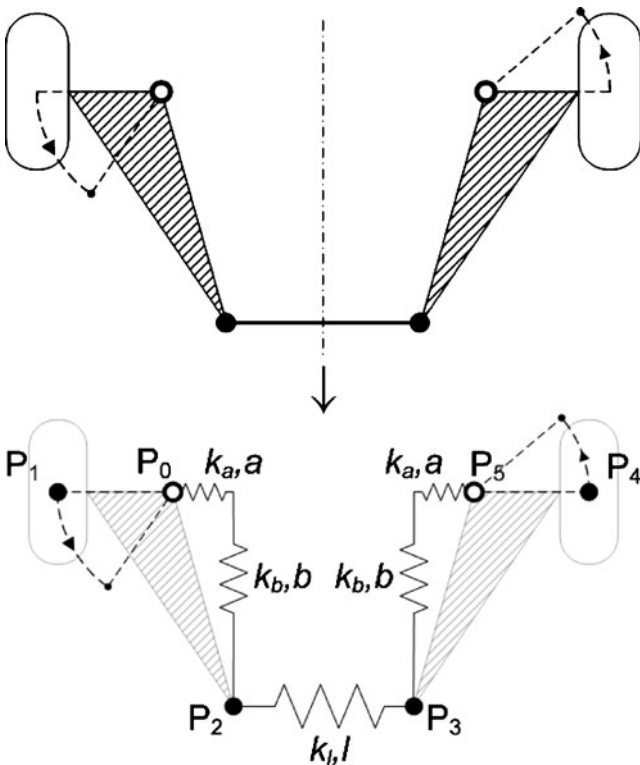


Fig. 9 Model adaptation of four-bar linkage for function-generation synthesis

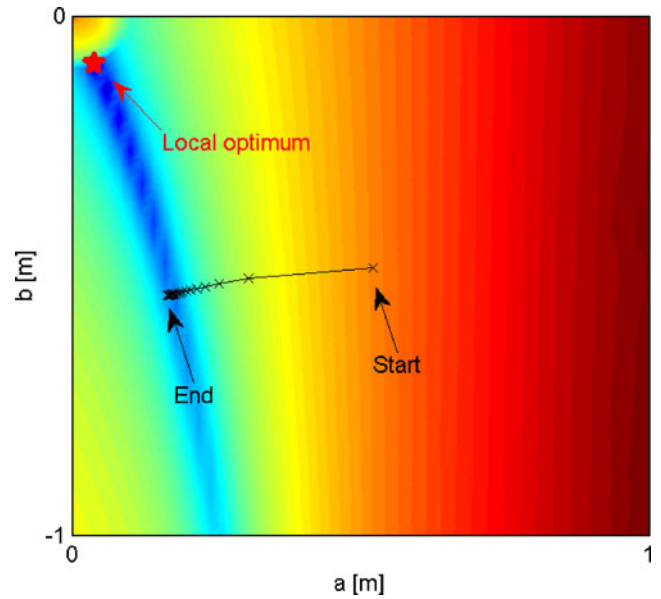


Fig. 10 Evolution of the design parameters a, b during the “mean values update” strategy

where the threshold value d_{min} is a chosen realistic minimum distance (e.g. $d_{min} = 10$ cm).

As for the full synthesis algorithm, it is observed that the optimization process may reach one of both local optima (see Collard et al. 2005b and Simionescu et al. 2002). Starting with various sets of initial parameters, it is sometimes hard to guess where it converges to. The Fig. 12 shows that running the algorithm from initial points located on a uniform seven-by-seven grid, these procedures lead to 47 relevant optimization results (starting points (0, 0) and (0.9, 0) give no optimum). Among the latter—symbolized by

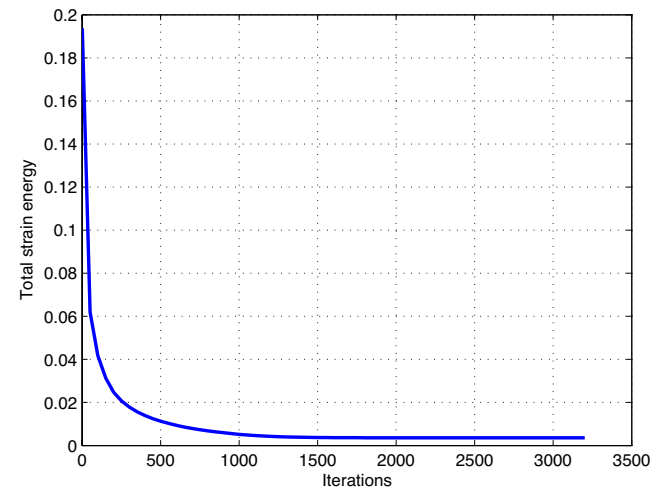


Fig. 11 Evolution of the total energy during the “mean values update” strategy

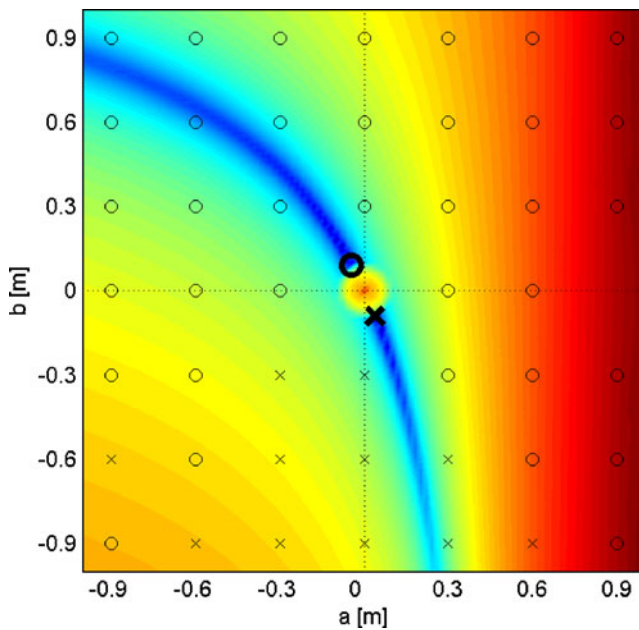


Fig. 12 Optimization of the steering linkage from various starting points

non-bold ‘o’ and ‘x’—11 converge to one local optimum—symbolized by bold ‘x’—while the 36 others reach another one—symbolized by bold ‘o’—which is actually the global one. The optimization method does not always yield the global optimum.

The two best linkages are drawn in Fig. 13: a trailing one and a leading one. Their steering error functions are plotted in Fig. 14 which represents the deviation of the outer wheel angle with respect to the Ackermann condition when the inner wheel turns from 0° to 40° . The numerical results are shown in Table 1. It is interesting to note that the optimum energy value is linked to the steering error: the leading linkage has the smallest total deformation and also the smallest steering error. Nevertheless, the small difference between both mechanisms’ performances does not justify the selection of one rather than the other: an additional design criterion has to be introduced to choose between both local optima.

5 Exploration of the design space

As regards the example of the previous section, it is obvious that the gradient-based optimization process does not

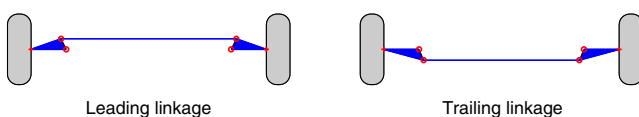


Fig. 13 Two local optima found for the four-bar steering synthesis

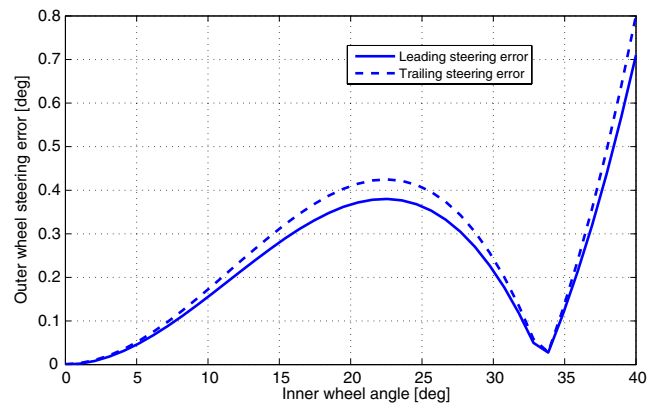


Fig. 14 Steering error of both optimum linkages

always reach the unique global optimum depending on the chosen starting point. It is worth noting that this is still the case using genetic algorithms as shown in the Section 5.1. Only adapted methods for global optimization can help us to find the unique global optimum; for instance, branch and bound techniques have been successfully applied to mechanism synthesis by Stolpe and Kawamoto (2005). However, as explained in the previous example, it may be interesting to propose several local optima to the designer. The Section 5.2 describes a simple heuristic method to explore the entire parameter space in order to find local optima. In the Section 5.3, we propose a possible criterion to choose the final best mechanism. Sections 5.2 and 5.3 are illustrated in the application of the next section.

5.1 Global optimization may fail with genetic algorithms

Genetic algorithms (G.A.) are well-known to be stochastic *global* optimization methods. Instead of gradient-based algorithms, G.A. may be used to optimally explore the design space. The cost function used is the resulting total strain energy already minimized with respect to the \mathbf{f}_i as in (7). Two subsequent runs of the G.A. may give different local optima for the four-bar linkage as shown in Fig. 15 to be compared with Fig. 12. Therefore, this justifies the development of a new technique to explore the design space in the next subsection. The purpose of the following

Table 1 Two best four-bar Ackermann steering linkages

	a [cm]	b [cm]	Energy [J]	Max err. [deg]	RMS err. [deg]
Leading	4.40	-8.98	7.98×10^{-7}	0.71	0.28
Trailing	-3.90	9.21	1.01×10^{-6}	0.80	0.32

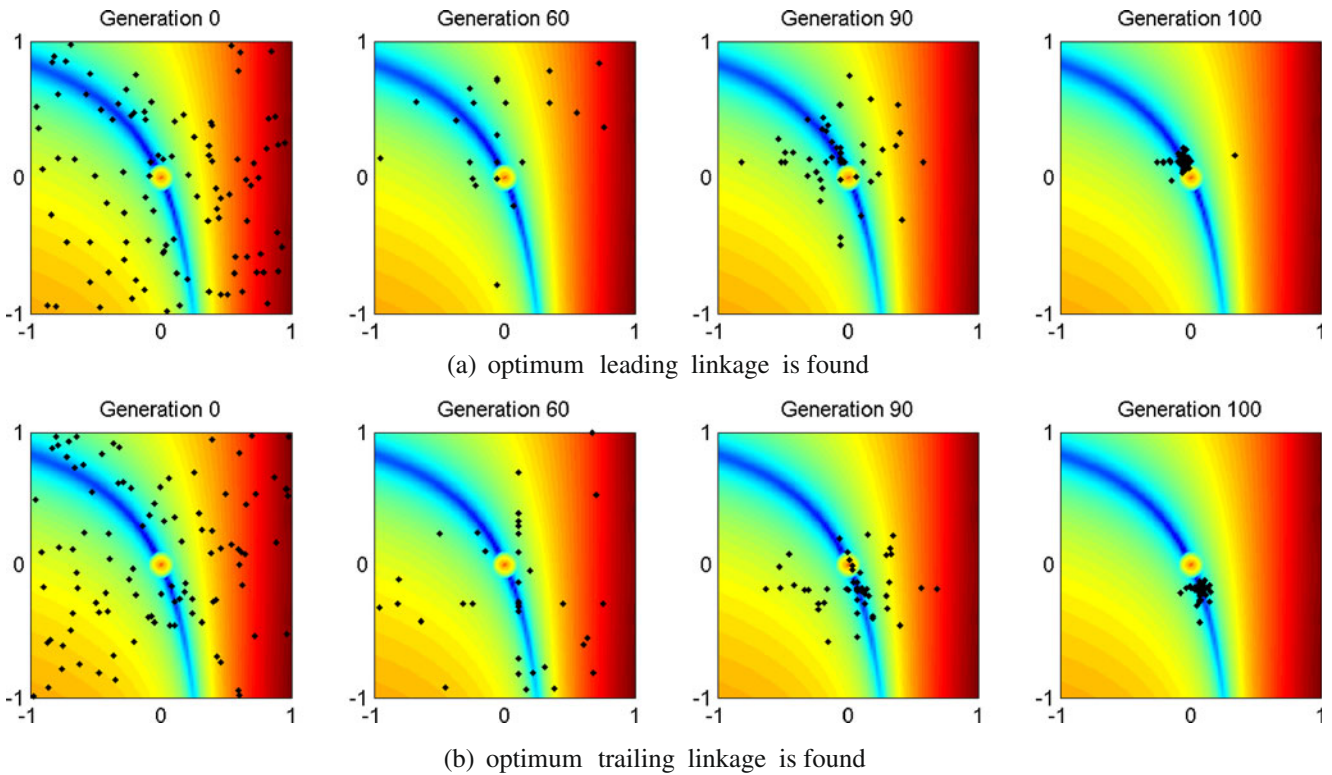


Fig. 15 Evolutions of the population distribution during two GA processes

heuristic exploration is to find several local optima but obviously does not guarantee to find all of them.

5.2 Exploration with the *nucleation* method

The first idea to explore the design space could be to perform multiple optimization processes starting from different uniformly-distributed initial points. A two-point-width uniform grid (Fig 16a) is used first and refined thereafter by adding a point between each segment: this provides a three-point-width grid (Fig 16b). Continuing the refinement this way, this enables us to reuse the computation of the previous grid as shown in the following of Fig 16 (reused points are drawn in grey). Nevertheless, it becomes very time-consuming since the number of optimization processes greatly increases (see second column of Table 2). A new method called “*nucleation*” is thus proposed to cope with this drawback.

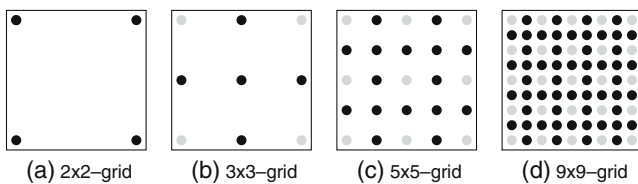


Fig. 16 Refinement of the starting points grid

This original method takes its inspiration from the nucleation process of crystal materials. Its algorithm flowchart is sketched in Fig. 17. The key idea is to create and make crystal nuclei growing from best locations on a given grid. These best locations are subsequently chosen with respect to the value of the objective function. The growth of the nuclei is stopped when they reach other nuclei or the design space boundary.

Practically, the algorithm of Fig. 17 begins with the discretization of the overall design space into a grid of equally-spaced points. The objective function e is computed over the grid and these points are sorted according to their objective value. In our case, this objective is the total strain energy (7) already used for the G.A. in the previous subsection. Then, each point of this sorted list is scanned to classify

Table 2 Numerical results of the design space exploration of six-bar linkage by *nucleation* method

Grid width	Number of points	Number of nuclei	Reduction factor	Number of local opt.
2 pts	16	1	16	1
3 pts	81	2	41	2
5 pts	625	13	48	6
9 pts	6,561	48	137	12
17 pts	83,521	284	294	9

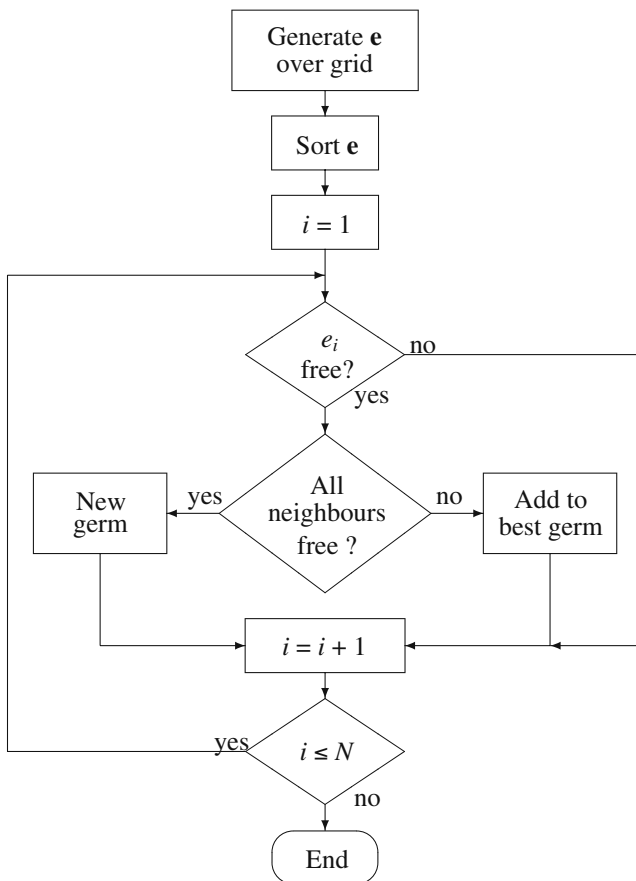


Fig. 17 Nucleation algorithm flowchart

it: if all its neighbors are free, i.e. not yet scanned and with worse objective values, a new germ is created with the point and its neighbors; otherwise, it is added to the nucleus of its best neighbor. All these operations are carried out until the complete sorted list has been fully scanned. The final result is the grouping of all the points around different nuclei. The best point of each nucleus then becomes a local optimum for the partial problem (7).

Taking the simple example of Fig. 18, a 3×3 grid is explored. The integer numbers represent the values of the energy at each point. Five iterations are needed to group the 9 points. The first two iterations (Fig. 18a, b) create the first two nuclei around points of energy 1 and 2. The considered neighbors, inside the boundaries, are located up, down, left and right from the center point. In iteration 3 (Fig. 18c), point 7 is added to the nucleus of its best neighbor (point 3). In the same way, point 8 is added to the nucleus of point 3 and point 9 to the nucleus of point 4 (Fig. 18d, e). The final configuration with two nuclei is illustrated in Fig. 18e.

If the four-bar steering linkage synthesis is now considered with the *nucleation* algorithm, four nuclei are obtained using only the partial minimization (7) as objective function. Thanks to this technique, the full optimization may be

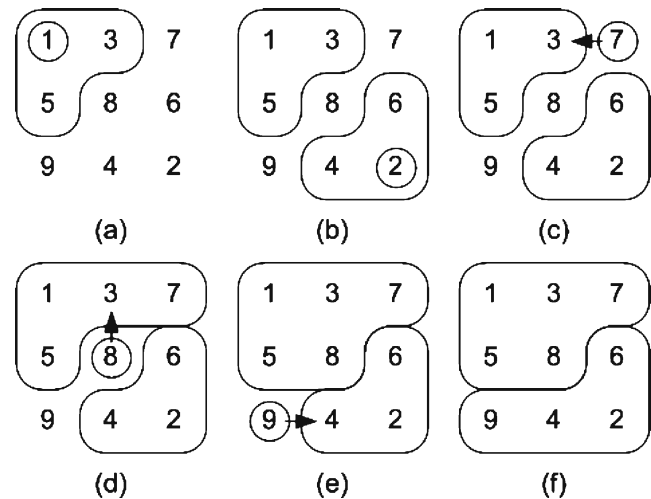


Fig. 18 Nucleation algorithm example

run only four times instead of 49 (see Section 4.2) to obtain the two local optima.

5.3 Final choice among the local optima

Once the design space is explored to form nuclei, the complete synthesis may begin, starting from the best candidate of each nucleus. This provides us with many local optima with respect to the minimum deformation. Obviously, as for the G.A., this does not guarantee reaching a global optimum but this is no longer the goal. Indeed, a last design optimization step is performed to find the final mechanism. After minimizing the mechanism's deformation, a different design criterion is applied to the corresponding *rigid* mechanisms. This last objective function is obviously different from the one used to create the nuclei. For example, as regards the steering linkage synthesis, the *nucleation* process is based on the minimum deformation energy (7) but the last objective will be the actual steering error of the rigid mechanism (as shown in Fig. 14 for the four-bar linkage) or even a more practical objective to choose the best candidate (see end of Section 6). It is therefore computed by simulation of the rigid mechanism instead of the extensible one. This is illustrated in the next section.

It is important to note that if the deformed mechanism still has a high energy after minimization, the corresponding *rigid* mechanism may have a quite different kinematic behavior from that of the deformed one. This may occur in the case of large motions (e.g. full rotating four-bar mechanisms) where the mechanism may reach branch configurations. It might be decided to take only mechanisms with a sufficiently small energy into account for the final choice. This would ensure a similar behavior between deformable and *rigid* mechanisms but this problem did not arise in the following application.

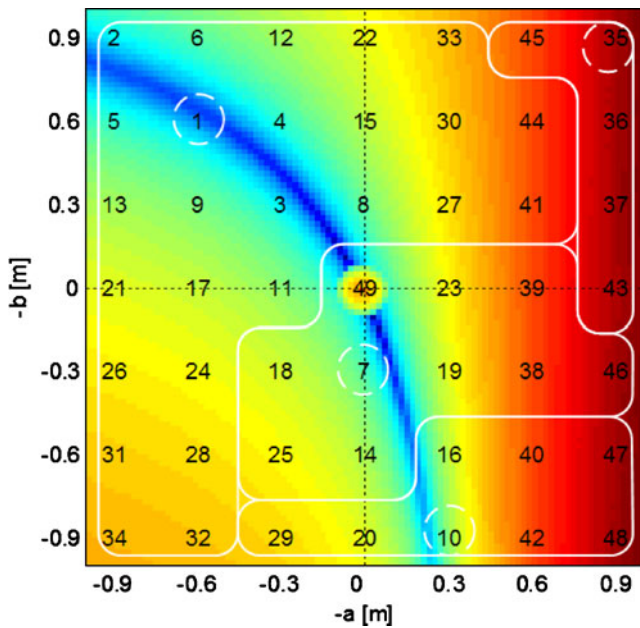


Fig. 19 Four-bar steering example with nucleation algorithm

6 Application to six-bar steering linkage synthesis

The goal of this application is the same as for the four-bar synthesis of Section 4.2. The main difference consists in a higher model complexity. The four-bar was parameterized with only two dimension variables. Here, the six-bar model, sketched in Fig. 20, requires five design parameters— a, b, l_1, l_2, y —which are reduced to four because of symmetry as explained in Simionescu et al. (2000). For example, l_1 can be expressed in terms of a, b, l_2 and y :

$$l_1 = \sqrt{\left(\frac{l-l_2}{2} - a\right)^2 + (b-y)^2} \tag{11}$$

Applying the nucleation points on a given grid is even more relevant for the six-bar than the four-bar linkage as shown in Table 2. Let us recall that the nucleation process enabled to reduce the number of full synthesis runs from 49— 7×7 grid—to four for the four-bar (See Section 5.2 and Fig. 19). In the case of the six-bar, this reduction factor— $\frac{\#points}{\#nuclei}$ —increases with the grid size and can reach 294 for a 17-point-width grid, compared to $49/4 = 12$ for the four-bar. This represents a considerable time reduction.

What could be surprising in Table 2 is the evolution of the number of local optima when the grid width increases from nine points to 17 points: the number of local optima would be expected to increase because the points of the 9×9 -grid are reused to build the 17×17 -grid. Nevertheless, The 12 local optima from the 9×9 -grid are not as good as the nine local optima from the 17×17 -grid. The latter nine all have their optimum energy below 10^{-6} while the

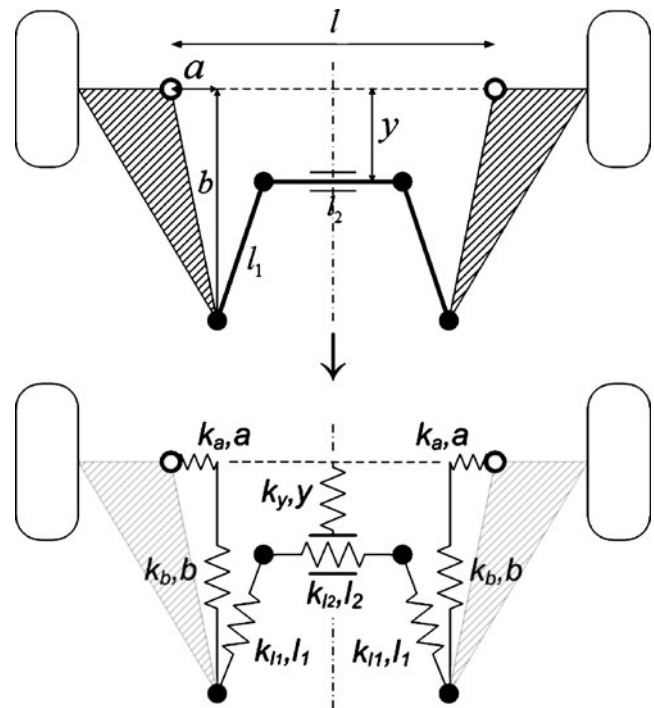


Fig. 20 Model adaptation of six-bar steering linkage

9×9 -grid provides only eight local optima below this value. Indeed, a rougher grid yields worse candidates making the full synthesis harder. The refinement of the grid provides more candidates, which are also better, and leads to an easier full synthesis: more syntheses converge to the same local optimum in this case.

Starting from the best candidate of each nucleus, the full optimization (4) is performed. The number of local optima is reported in Table 2. The nucleation process applied to the most refined grid highlights nine local optimum mechanisms reported in Table 3 and sketched in Fig. 21. They are sorted according to their strain energy value: the least deformed mechanisms give the highest potential steering precision. This is checked in the following. The ‘Popularity’ column of Table 3 informs us about the total

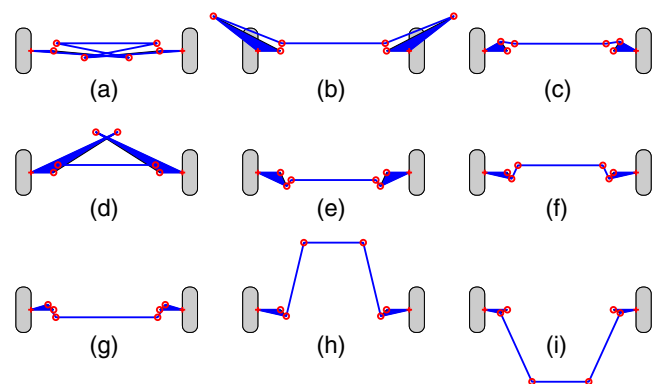


Fig. 21 Nine local optima for the six-bar steering linkage synthesis

Table 3 Nine local optimum six-bar steering linkages

	a [m]	b [m]	l_2 [m]	y [m]	Energy [J]	Popularity [%]
(a)	-1.00	0.09	1.33	0.100	5.88×10^{-13}	8.5
(b)	0.89	-0.46	1.37	0.100	1.41×10^{-12}	11.2
(c)	0.09	-0.12	1.21	0.095	3.38×10^{-11}	42.1
(d)	-0.85	-0.53	1.29	0.099	2.16×10^{-10}	5.0
(e)	-0.08	0.18	1.12	-0.100	2.49×10^{-09}	4.6
(f)	-0.06	0.08	1.12	0.095	2.30×10^{-08}	10.5
(g)	0.07	-0.07	1.35	-0.100	1.01×10^{-07}	7.3
(h)	-0.08	0.08	0.79	0.888	1.08×10^{-07}	0.1
(i)	0.09	0.04	0.75	-0.952	1.30×10^{-07}	10.7

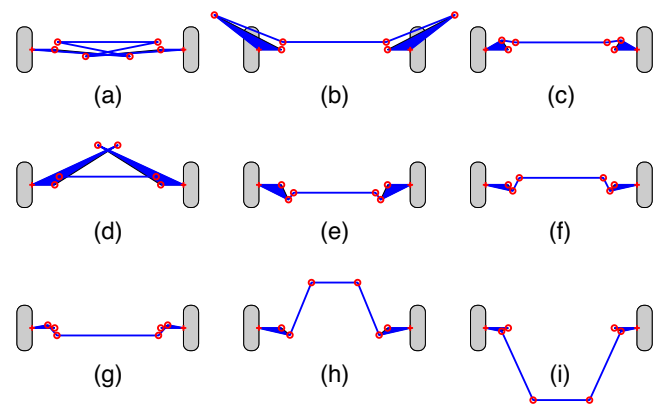
size of the nuclei reaching each local optimum with respect to the grid size. For example, the third one (c) has an attraction pool of 35,158 points, which represents 42.1% of the grid (83,521 points) whereas the eighth one (h) influences only 0.1% of it.

As previously explained, the selection of the most relevant local optimum is not straightforward and depends on other design criteria. For example, one might choose the third mechanism a priori because of its compactness. But here, the main criterion is the steering error as explained in Section 5.3. This last objective is computed by simulating the steering behavior of the corresponding rigid mechanism as shown in Fig. 14. The results of this optimization refinement are reported in Table 4 linked with Fig. 22.

Compared with the results of the four-bar steering linkage in Fig. 14, all the nine local optima improve both the maximum and the RMS steering errors. Moreover, these values are so small that it could be interesting to add a more practical criterion. A robustness criterion is proposed: the sensitivity analysis of the steering error with respect to the optimum design parameters. A perturbation of 0.5 mm—which could correspond to absolute precision machining—is chosen for each of the four dimensions and

Table 4 The nine local optima after optimization refinement

	a [m]	b [m]	l_2 [m]	y [m]	Energy [J]	Max error [deg]	RMS error [deg]	Mean deviation [percent of center]
(a)	-0.99	0.09	1.33	0.100	6.78×10^{-12}	0.0015	0.0003	3,627
(b)	0.89	-0.46	1.37	0.101	1.72×10^{-11}	0.0024	0.0005	16,206
(c)	0.09	-0.12	1.21	0.095	2.01×10^{-08}	0.0220	0.0031	609
(d)	-0.83	-0.53	1.28	0.108	2.10×10^{-09}	0.0685	0.0151	232
(e)	-0.09	0.20	1.08	-0.104	4.48×10^{-08}	0.1026	0.0335	368
(f)	-0.06	0.08	1.11	0.093	1.56×10^{-05}	0.3215	0.1232	45
(g)	0.09	-0.04	1.34	-0.099	9.20×10^{-06}	0.3659	0.1245	140
(h)	-0.11	0.09	0.61	0.601	1.11×10^{-06}	0.5977	0.1923	2
(i)	0.09	0.04	0.75	-0.953	2.34×10^{-04}	0.5647	0.1718	37

**Fig. 22** The nine local optima after optimization refinement

also for all combinations of these. The steering error cost function is thus computed 16 times around each of the nine optima. In the last column of Table 4, the mean deviation of the objective over the 16 neighbors of each optimum is represented. This may give a basis for the last decision of the designer. Undeniably, mechanism (h) is the most robust in a sensitivity sense. However, the designer could choose mechanism (f) because it is more compact than (h) or (i). All these considerations obviously depend on how each criterion is taken into account by the designer.

7 Conclusion and prospects

Based on a strain energy approach of extensible-link mechanisms coupled with the use of natural coordinates, an original optimization method has been developed to solve path synthesis and function generation problems of planar mechanisms. Divided into two stages, the first proposed method alternatively tries to minimize the total deformation energy with respect to the point coordinates and updates the natural lengths of the springs accordingly. This method was then extended to a full synthesis approach. It seems to be

more efficient but does not always reach the global optimum as illustrated via the four-bar steering linkage application.

Following that observation, the question of finding the global optimum has been discussed. Genetic algorithm optimizations have been shown to also miss the global optimum mechanism. This question has thus been reviewed and extended with the exploration of the design space to find the majority of local optima. To avoid “astronomical” CPU time, a simple method inspired from crystal nucleation has been proposed to divide the design space into nuclei centered on local optima. Starting from the latter, the optimization has been refined and a last criterion applied to choose the final mechanism.

In terms of prospects, our effort will concentrate on developing other exploration strategies of the design space. The comparison of these various strategies and their results will be useful in validating them. It could be also interesting to classify the various local optima based on different types of criteria. We also intend to extend the method further to topologies with prismatic joints or to three-dimensional mechanisms. Extending the application field is also an interesting prospect: other kinematic objectives instead of path or function-generator synthesis or even dynamical ones could be investigated. The more challenging issue of topology optimization of mechanisms could be tackled on the basis of this work, involving for example an additional higher-level optimization process as proposed by Sedlacek et al. (2005), or simultaneously optimizing the topology and the dimensions of mechanisms as developed by Stolpe and Kawamoto (2005), or Kawamoto (2005) for path-generation problems using truss representation.

Acknowledgments This research was sponsored by the Belgian Program on Interuniversity Attraction Poles, *Advanced Mechatronic Systems-AMS* (IUAP5/06), initiated by the Belgian State—Prime Minister’s Office—Science Policy Programming (IUAP V/6). The scientific responsibility is assumed by its authors.

References

- Alba J, Doblaré M, Gracia L (2000) A simple method for the synthesis of 2D and 3D mechanisms with kinematic constraints. *Mech Mach Theory* 35:645–674
- Avilés R, Vallejo J, Ajuria G, Agirrebeitia J (2000) Second-order methods for the optimum synthesis of multibody systems. *Struct Multidisc Optim* 19:192–203
- Cabrera J, Simon A, Prado M (2002) Optimal synthesis of mechanisms with genetic algorithms. *Mech Mach Theory* 37:1165–1177
- Collard JF, Fiset P, Duysinx P (2005a) Contribution to the optimization of closed-loop multibody systems: application to parallel manipulators. *Multibody Syst Dyn* 13:69–84
- Collard JF, Fiset P, Duysinx P (2005b) Optimal synthesis of mechanisms using time-varying dimensions and natural coordinates. In: *Proc of the sixth world congress of structural and multidisciplinary optimization*
- Hansen JM (2002) Synthesis of mechanisms using time-varying dimensions. *Multibody Syst Dyn* 7:127–144
- Jensen OF, Hansen JM (2003) Treating the problem of non-assembly for dimensional synthesis of spatial mechanisms. In: Ziegler S (ed) *The fifth world congress of structural and multidisciplinary optimization*. ISSMO, Italian Polytechnic Press, pp 431–432
- Jiménez JM, Alvarez G, Cardenal J, Cuadrado J (1997) A simple and general method for kinematic synthesis of spatial mechanisms. *Mech Mach Theory* 32(3):323–341
- Kawamoto A (2005) Path-generation of articulated mechanisms by shape and topology variations in non-linear truss representation. *Int J Numer Methods Eng* 64:1557–1574
- Laribi M, Mlika A, Romdhane L, Zeghloul S (2004) A combined genetic algorithm-fuzzy logic method GA-FL in mechanisms synthesis. *Mech Mach Theory* 39:717–735
- Lio MD, Cossalter V, Lot R (2000) On the use of natural coordinates in optimal synthesis of mechanisms. *Mech Mach Theory* 35:1367–1389
- Mariappan J, Krishnamurty S (1996) A generalized exact gradient method for mechanism synthesis. *Mech Mach Theory* 31(2):413–421
- Marín FTS, González AP (2003) Global optimization in path synthesis based on design space reduction. *Mech Mach Theory* 38:579–594
- Minnaar R, Tortorelli D, Snyman J (2001) On nonassembly in the optimal dimensional synthesis of planar mechanisms. *Struct Multidisc Optim* 21:345–354
- Powell MJD (1970) A new algorithm for unconstrained optimization. In: *Nonlinear programming*. Academic, pp 31–65
- Pramanik S (2002) Kinematic synthesis of a six-member mechanism for automotive steering. *ASME J Mech Des* 124:642–645
- Sancibrian R, Viadero F, García P, Fernández A (2004) Gradient-based optimization of path synthesis problems in planar mechanisms. *Mech Mach Theory* 39:839–856
- Sedlacek K, Gaugele T, Eberhard P (2005) Topology optimized synthesis of planar kinematic rigid body mechanisms. In: *Proc of the multibody dynamics 2005, Ecomas Thematic Conference*
- Shiakolas P, Koladiya D, Kebrle J (2005) On the optimum synthesis of six-bar linkages using differential evolution and the geometric centroid of precision positions technique. *Mech Mach Theory* 40:319–335
- Simionescu P, Beale D (2002) Optimum synthesis of the four-bar function generator in its symmetric embodiment: the Ackermann steering linkage. *Mech Mach Theory* 37:1487–1504
- Simionescu P, Smith M, Tempea I (2000) Synthesis and analysis of the two loop translational input steering mechanism. *Mech Mach Theory* 35:927–943
- Stolpe M, Kawamoto A (2005) Design of planar articulated mechanisms using branch and bound. *Math Program* 103:357–397
- Vallejo J, avilés R, Hernández A, Amezuza E (1995) Nonlinear optimization of planar linkages for kinematic syntheses. *Mech Mach Theory* 30(4):501–518
- Wong J (2001) *Theory of ground vehicles*, 3rd edn. Chap handling characteristics of road vehicles. Wiley, New York
- Zhou H, Cheung EH (2001) Optimal synthesis of crank-rocker linkages for path generation using the orientation structural error of the fixed link. *Mech Mach Theory* 36:973–982