6

Optimal Design of Mechanisms for Robot Hands

J.A. Cabrera, F. Nadal and A. Simón
University of Málaga, Department of Mechanical Engineering
Spain

1. Introduction

Scientific progress has been very important in the last two centuries, from the invention of the steam machine until the electronic age. Robots design has also suffered an important progress in the last decades and many approaches deal with this issue. Our approach carries out an optimal design of planar 1 DoF mechanisms which are used for robot hands or grippers. These kinds of mechanisms are amply used because of their simplicity and they only need one motor to move them, so many robots use this kind of mechanism as grippers.
We studied a new technique to carry out an optimal design of a gripper in this work. A searching procedure is developed, which applies genetic algorithms based on an evolutionary approach. The new method has proved to solve synthesis problems of planar mechanisms and has been used for testing a hand robot mechanism, showing that the solutions are accurate and valid for all cases.
Different techniques have been used for mechanism synthesis. In graphical techniques the use of the coupler curve atlas (Hrones & Nelson, 1951) who developed the four-bar mechanisms atlas with almost 10,000 curves is especially remarkable. The solution by (Zhang et al., 1984) focused on five-bar geared linkages. These methods are easy and fast to use but at a low precision rate. The first reference addressing analytical methods was made by (Sandor, 1959), followed by (Erdman, 1981), (Kaufman, 1978) and (Loerch et al.,1975). References about the subject by (Freudenstein, 1954), (Beyer, 1963), (Hartenberg & Denavit, 1964) also exist, solving the synthesis problem using precision points to be reached by the coupler point of the mechanism, but these methods restrict the number of precision points in order to allow the solution of the mathematical system to be closed and show problems caused by wrong sequence of the precision points followed. The great increase in computer power has permitted the recent development of routines that apply numerical methods to the minimization of a goal function. One of the first authors who studied these methods was (Han, 1966), whose work was later improved by (Kramer & Sandor, 1975), (Sohoni & Haug, 1982). They optimized one of the most common goal functions: the error between the points tracked by the coupler and its desired trajectory.
The approach to mechanism synthesis presented in this work deals with evolutionary algorithms based on a differential evolution technique. These kinds of algorithms were first introduced by (Holland, 1973,1975), whose work is included in Goldberg's book (Goldberg, 1989), and they have been extensively and successfully applied to different optimization problems. These methods define a starting population that is improved by approximations to the goal function making use of natural selection mechanisms and natural genetic laws.

Source: Frontiers in Evolutionary Robotics, Book edited by: Hitoshi Iba, ISBN 978-3-902613-19-6, pp. 596, April 2008, I-Tech Education and Publishing, Vienna, Austria

The main advantages of these methods are their simplicity in implementing the algorithms and their low computational cost. In addition, there is no need for extensive knowledge of the searching space, as it is continuous, presents local minimums or shows other mathematical characteristics demanded by traditional searching algorithms. Many researches use this technique for optimum mechanisms synthesis, but they apply a single goal function to carry out the optimization problem. The main difference in our approach is that we use several goal functions and constraints, so the optimization problem is more complex and useful for designing hand robots. Multiobjective techniques are used by (Rao & Kaplan, 1986), (Krishnamurty & Turcic, 1992). (Kunjur & Krishnamurty, 1997) use a multiple criteria optimization approach that obtains Pareto-optimal design solution sets. They apply this method to a mechanism dimensional synthesis with two objective functions and three constraints. (Haulin & Vinet, 2003) develop a multiobjective optimization of hand prosthesis four-bar mechanisms. They use the Matlab optimization toolbox and a goal attainment method for the optimization. All of these need a high power calculus and can fail because they might find the solution in a local minimum ending the search without reaching the true optimal solution.

In this work we have developed an evolutionary approach based on Differential Evolution technique (Storn & Price, 1997). Other authors, like (Cabrera et al., 2002) use this technique for the optimum synthesis of four-bar mechanism. (Shiakolas et al., 2005) also uses Differential Evolution for the optimum synthesis of six-bar linkages, but they apply a single goal function to carry out the optimization problem. We use several goal functions and constrains in our approach, so the optimization problem is more complex and useful in a great variety of problems. We apply our algorithm to hand mechanism synthesis in one-DOF robot, but it is possible to use it in different problems, only changing the goal functions and constraints.

2. Optimization method

Evolutionary algorithms (EAs) are different from more normal optimization and search procedures in four ways:

- Evolutionary algorithms work with a coding of the parameter set, not the parameters themselves.
- Evolutionary algorithms search with a population of points, not with a single point.
- Evolutionary algorithms use evaluations of goal functions, not derivatives or other auxiliary knowledge.
- Evolutionary algorithms use probabilistic transition rules, not deterministic rules.

Altogether, these four differences contribute to an evolutionary algorithm's robustness and turn out to be an advantage over other more commonly used techniques.

Definition.-The optimization problem is given by:

$$\begin{aligned} &\min \quad F(X)=\left(f_1(X), f_2(X), ..., f_n(X)\right) \\ &\textit{subject to}: \\ &\qquad g_j(X) \le 0 \quad j=1,2,..,m \\ &\qquad \Omega:\left\{x_i \in [li_i, ls_i] \quad \forall x_i \in X\right\} \end{aligned} \tag{1}$$

Where f_i are the goal functions, i.e. a set of functions where each one expresses a feature or objective to be optimized, and where each individual, X, obtains a value, its fitness. Furthermore $g_j(.)$ are the constraints defining the searching space.
The strategy of evolutionary methods for optimization problems begins with the generation of a starting population. Each individual (chromosome) of the population is a possible solution to the problem and it is formed by parameters (genes) that set the variables of the problem. Genes can be schematized in several ways. In the first approach by (Holland, 1973, 1975) they are binary chains, so each x_i gene is expressed by a binary code of size n. Another way to express the genes, as done in this work, is directly as real values. All genes are grouped in a vector that represents a chromosome, (Storn & Price, 1997) , (Wright, 1990):

$$X = \begin{bmatrix} x_1 & x_2 & . & . & x_k \end{bmatrix} \quad \forall x \in \Re \tag{2}$$

Next the starting population has to evolve to populations where individuals are a better solution. This task can be reached by natural selection, reproduction, mutation or other genetic operators. In this work, selection and reproduction are carried out sequentially and mutation is used as an independent process.Now, we will define some basic concepts that are very common in multiobjective optimization.
Definition.-Pareto dominance:
A vector, $X^* \in \Omega$, is said to dominate $Y^* \in \Omega$, denoted $X^* \preceq Y^*$, if and only if $f_i(X^*)$ is partially less than $f_i(Y^*)$, i.e.:

$$\forall i \in \{1,....,n\} : \left\{ f_i\left(X^*\right) \le f_i\left(Y^*\right) \right\} \wedge \left\{ \exists i \in \{1,..,n\} : f_i\left(X^*\right) < f_i\left(Y^*\right) \right\}$$

Definition.- Non-dominated or Pareto-optimal solution:
A vector, $X^* \in \Omega$, is said to be non-dominated if and only if there is no vector which dominates X^*, i.e., $\neg \exists Y^* \in \Omega : Y^* \preceq X^*$
Definition.- Pareto-optimal set:
A set, $P \subset \Omega$, is said to be Pareto-optimal if and only if: $\forall X^* \in P : \neg \exists Y^* \in \Omega : Y^* \preceq X^*$
Now we are qualified to explain the evolutionary algorithms that we propose. This algorithm is based on the Differential Evolution algorithm proposed by (Storn & Price, 1997), but we introduce a set of new features:

- The original Differential Evolution algorithm was used in optimization problems with one goal function. We use it with multiobjective problems.
- We use a Pareto-based approach to sort the population and this one is divided into non-dominated and dominated population. The 'best' individuals are chosen to run the Differential Evolution strategy from the non-dominated sub-population.
- We use a genetic operator called mutation, which is not used in the original algorithm. This operator is of great significance in certain problems to prevent stagnation (Lampinen & Zelinka, 2000).
- We use a function to control the number of non-dominated individuals in the population.
- We introduce a procedure for handling the constraints. This procedure is based on the work proposed by (Lampinen, 2002), but applied to multiobjective problems.

Definition.- Selection of a couple for reproduction:
For selection, two individuals are randomly chosen from the population and they form a couple for reproduction. The selection can be based on different probability distributions, such as a uniform distribution or a random selection from a population where the weight of each individual depends on its fitness, so that the best individual has the greatest probability to be chosen. In this paper, an individual randomly selected from the Pareto-optimal set of the population and two individuals randomly selected from the complete population with uniform distribution are chosen for reproduction and they make up a disturbing vector, V. The scheme, (Storn and Price, 1997), known as Differential Evolution yields:

$$\begin{aligned} &P \equiv \{X_i : i \in [1, NP]\} \\ &\zeta \subset P : Y_{r1} \in \zeta \\ &V = Y_{r1} + F \cdot (Y_{r2} - Y_{r3}) \end{aligned} \tag{3}$$

where Y_{r1} is an individual chosen randomly from the Pareto-optimal set ζ of population P, which is obtained as defined previously, Y_{r2} and Y_{r3} are two individuals randomly selected from population P among NP individuals, and F is a real value that controls the disturbance of the Pareto-optimal individual. This disturbing vector V and individual i of the population form the couple for reproduction. This way to obtain parent V, it maintains the philosophy of the original Differential Evolution algorithms, where the best individual of the population and two individuals chosen randomly are used to obtain the disturbing vector V. In some ways Y_{r1} are the 'best' individuals in the actual population, because they are chosen from the Pareto-optimal set.
Definition.- Reproduction:
Next, for reproduction, V is crossed with individual i of the current population to generate individual i of the next population. This operator is named crossover.
In natural reproduction, parents' genes are exchanged to form the genes of their descendant or descendants. As shown in Figure 1, reproduction is approached by a discrete multipoint crossover that can be used to generate X_i^N: parent X_i^G provides its descendant with a set of genes randomly chosen from its entire chromosome and parent V provides the rest. Crossover is carried out with a probability defined as CP∈ [0, 1].
Definition.- Selection of new descendents:
The following steps are performed to choose which individual X_i^N or X_i^G passes to the next population:

- If the new X_i^N descendent fulfills more constraint than parent X_i^G, then the new descendent is chosen for the next population, i.e.,
- if $\exists k : g_k(X_i^N) \le 0 \rightarrow \xi_k(X_i^N) = 1$, then:
- $\forall k \in \{1,..,m\} : \sum \xi_k(X_i^N) > \sum \xi_k(X_i^G) \rightarrow X_i^{G+1} = X_i^N$
- If parent X_i^N fulfills more constraints than the new X_i^G descendent, then the parent is chosen for the next population, i.e, $\forall k \in \{1,..,m\} : \sum \xi_k(X_i^N) \le \sum \xi_k(X_i^G) \rightarrow X_i^{G+1} = X_i^G$
- If both individuals X_i^N and X_i^G fulfill all constraints or do not fulfill some of them, then the individual which dominates is chosen for the next population, i.e.:

$$\left\{\begin{array}{l}\left\{\forall k\in\{1,..,m\}: g_k(X_i^N)\le 0\wedge g_k(X_i^G)\le 0\right\}\\ \vee\left\{\exists k: g_k(X_i^N)>0\wedge g_k(X_i^G)>0\right\}\end{array}\right\}$$
$$\wedge\left\{X_i^N\preceq X_i^G\right\}\rightarrow X_i^{G+1}=X_i^N$$

$$\left\{\begin{array}{l}\left\{\forall k\in\{1,..,m\}: g_k(X_i^N)\le 0\wedge g_k(X_i^G)\le 0\right\}\\ \vee\left\{\exists k: g_k(X_i^N)>0\wedge g_k(X_i^G)>0\right\}\end{array}\right\}$$
$$\wedge\left\{X_i^G\preceq X_i^N\right\}\rightarrow X_i^{G+1}=X_i^G$$

Therefore the population neither increases nor decreases.

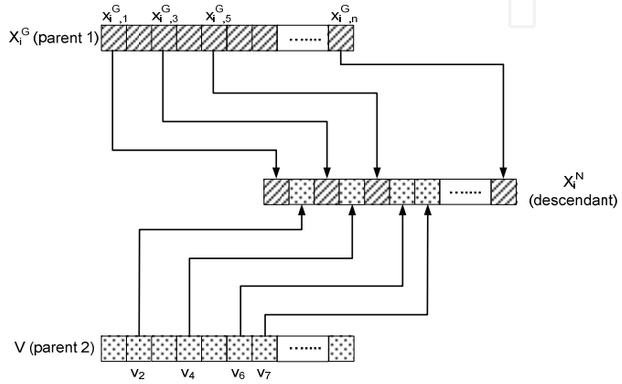


Figure 1. Reproduction scheme based on discrete multipoint crossover

Definition.-Mutation

A new mutation procedure of the parameters to be optimized is developed in this work. Mutation is an operator consisting of random change of a gene during reproduction. We have verified that this procedure is fundamental to obtain the optimum when the parameter range values are very different. The mutation procedure changes only some of these parameters allowing to find the correct optimum and not to stop in a local minimum. This problem was called stagnation in the work performed by (Lampinen & Zelinka, 2000) and it is shown in Figure 2.

The whole procedure to obtain a new descendent is shown in Figure 2a. In this case, there are two different parameters (genes) and the optimum has a very different value for these two parameters. And we suppose that the individuals of the population are situated around a local minimum due to the evolution of the population. The fundamental idea of this discussion consists of the step length adaptability along the evolutionary process. At the beginning of the generations the step length is large, because individuals are far away each from other. As evolution goes on, the population converges and the step length becomes smaller and smaller. For this reason if the mutation procedure does not work properly, it is possible to drop in a local minimum. In Figure 2a and 2b the differences between both strategies with and without mutation procedure are shown.

The way to obtain a new descendent of the next population without mutation procedure is shown in Figure 2a. In this case the V and X_i^G couple generates the X_i^N descendent, but this new chromosome may not reach the global minimum due to the fact that the absolute values of the genes that compose it are very different, and the selection plus reproduction operations are not able to make the new descendent by themselves to overcome the valley of the local minimum.

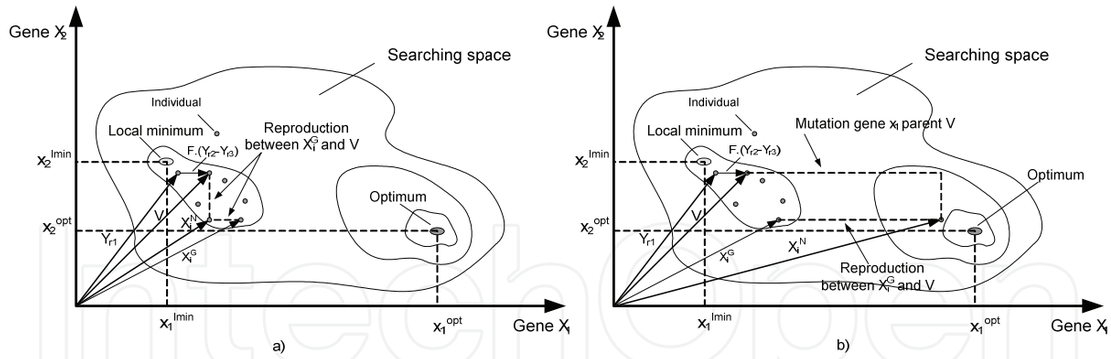


Figure 2. a) Differential Evolution without mutation procedure. b) Differential Evolution with mutation procedure

With the mutation procedure it is possible to solve the problem explained before. The generation of a new descendant using the mutation procedure is schemed in Figure 2b. Here, the value of one or several of the genes of the V and X_i^G couple is changed in a range defined by the user, when the reproduction is taking place. This fact yields a new descendent, X_i^N, which has a different fitness from the X_i^N descendent studied in the previous case. This allows the algorithm to look for individuals with better fitness in the next generation.

In this work, mutation is defined as follows: when gene x_i mutates, the operator randomly chooses a value within the interval of real values (x_i, x_i±range), which is added or subtracted from x_i, depending on the direction of the mutation.

Mutation is carried out with a probability defined as MP∈ [0, 1], much lower than CP. Once the genetic operators are described, the optimization algorithm will be explained.

3. POEMA algorithm

The proposed algorithm, which is defined as Pareto Optimum Evolutionary Multiobjective Algorithm (POEMA), has the following steps:

1. The algorithm starts with the random generation of a starting population with NP individuals.
2. Next, the algorithm calculates the Pareto-optimal set of the total population and obtains its size, N_{pr}. To preserve diversity, the number of non-dominated individuals is maintained along iterations according to the following function:

$$\forall k \in \{1,..,\text{itermax}\} : N_{pr} \leq N_0 + k \cdot \Delta N$$

   Where *itermax* is the number of iterations in the algorithm, N_0 is the number of allowed initial individuals in the Pareto-optimal set and ΔN is a parameter to increase the allowed initial individuals with the iterations. So a maximum number of non-dominated individuals are allowed. If this maximum is exceeded, the nearest neighbor distance function is adopted (Abbass, 2002).
3. To create the new population, the selection of couple, reproduction and mutation operator are used according to definitions described above.
4. If the algorithm reaches the maximum number of iterations, it finishes; otherwise return to step 2.

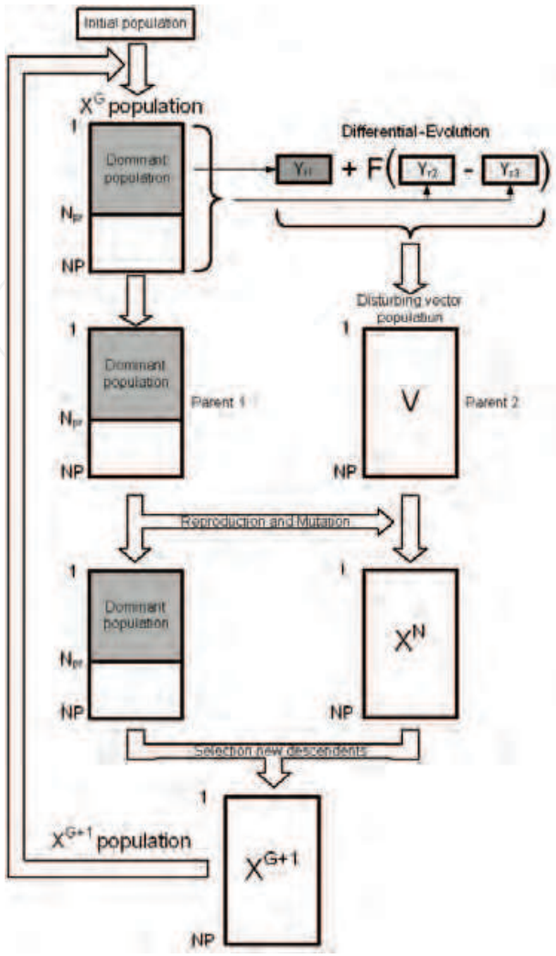


Figure 3. Scheme algorithm

A scheme of the proposed algorithm is shown in figure 3. First, we generate a parent for reproduction according to the Differential-Evolution scheme, which was defined above. Hence the couple for reproduction is the actual population, X^G, and the disturbing vector population is V. As the reproduction and mutation operator are carried out, a new population is obtained, X^N. This one is compared with the actual population, X^G, to obtain the new population, X^{G+1}. At this point, we obtain the Pareto-optimal set of the new population according to what we explained above and we run a new cycle in the algorithm. As we can observe, the new population maintains the same number of individuals as the previous one, so this algorithm does not increase the number of individuals in the population.

4. Goal function and constraint formulation in the two proposed problems

Once we have described the POEMA algorithm, we will develop the goal functions for the problem of a robot hand mechanism in this section. The advantage of using a multiobjective evolutionary algorithm is that we can include either kind of goal function that other works have resolved individually. When a mechanism is designed, several kinds of features are kept in mind:

- Geometric features: a link has to measure a specific length, etc.

- Kinematical features: a point in the mechanism has to follow a specific trajectory, velocity or acceleration law during its movements.
- Mechanical advantage: amount of power that can be transmitted by the mechanism for one complete cycle.

First problem.- In this problem we dealt with a robot hand mechanism (Figure 4). The goal functions for this problem were:

1.- A grasping index (GI) that is similar to the mechanical advantage concept, (Ceccarelli, 1999), which is obtained by means of the method of virtual work applied between the input slider (point F) and the output link (point E), obtaining:

$$-P \cdot v_y^F = 2 \cdot \left(F \cdot \cos\psi \cdot v_x^E + F \cdot \sin\psi \cdot v_y^E \right) \quad \Rightarrow \quad GI = \frac{2 \cdot F \cdot \cos\psi}{P} = \frac{v_y^F}{v_x^E + v_y^E \cdot \tan\psi} \tag{4}$$

As we can see in the previous equation, the *GI* grasping index must be maximized. However, we will convert this objective in a minimizing function, so the first goal function is:

$$f_1 = \min\left(\frac{v_x^E + v_y^E \cdot \tan\psi}{v_y^F} \right) \tag{5}$$

2.- The second objective is to minimize the acceleration in the *E* contact point to avoid a big impact on the object. So the second goal function is:

$$f_2 = \min\left(a_x^E \right) \tag{6}$$

3.-Another objective is to reduce the weight of the mechanism. If we consider all the links with the same thickness, this objective will be:

$$f_3 = \min \sqrt{\sum_{i=1}^{n} \left(x_i^2 \right)} \tag{7}$$

Where x_i is the length of the *i* link in the mechanism.

4.- The last objective is to standardize the link length in the mechanism to avoid a great difference between the length of the different links. So the fourth goal function is:

$$f_4 = \min \frac{\sqrt{\sum_{i=1}^{n} \sum_{j=1}^{n} \left(x_i - x_j \right)^2}}{\sqrt{\sum_{i=1}^{n} \left(x_i^2 \right)}} \tag{8}$$

The constraints for this problem are:

$$g_1 \equiv v_x^E > 0 \tag{9}$$

$$g_2 \equiv \overline{EJ} = D_{mec} \tag{10}$$

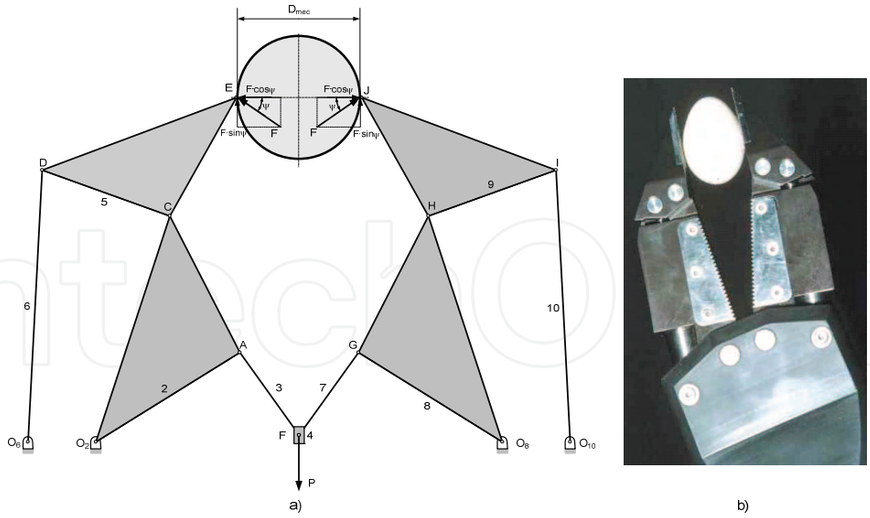


Figure 4. a) Hand robot mechanism for problem 1 and 2. b) Real robot manipulator

The first constraint says that the velocity of the *E* contact point must go in the same direction as the grasping. And the second constraint says that the distance between the two contact points, $\overline{EJ}$, must be object size D_{mec}. Hence the whole optimization problem is shown in the next equation:

$$\begin{aligned} &\min\{f_1(X), f_2(X), f_3(X), f_4(X)\} \\ &subject\ to: \\ &g_1(X) \\ &g_2(X) \\ &l_{\min} \le X \le l_{\max} \end{aligned} \tag{11}$$

Where, the X vectors are the design variables. We also introduce a boundary constraint of the design variables. To find the design variables, it is necessary to do a kinematic analysis of the mechanism. We use the Raven method to determine the position, velocity and acceleration of the *E* contact point, because these variables are in f_1 and f_2 goal functions in the first problem and in f_1, f_2 and f_3 goal functions in the second problem. The rest of the goal functions in both problems only need the link lengths of the mechanism. Hence, we establish the following scheme according to Figure 5:

$$\vec{E} = \vec{r}_{1x} + \vec{r}_{1y} + \vec{r}_6 + \vec{g}_6 \tag{12}$$

Then to obtain the position of the contact point E:

$$E_X = -r_{1x} + r_6 \cdot \cos\theta_6 + g_6 \cdot \cos(\theta_5 + \delta) \tag{13}$$

$$E_y = r_{1y} + r_6 \cdot \sin\theta_6 + g_6 \cdot \sin(\theta_5 + \delta) \tag{14}$$

$$\psi = f(\theta_5, \delta, g_6, r_5) \tag{15}$$

To obtain the velocity of the contact point E:

$$v_x^E = -r_6 \cdot \omega_6 \cdot \sin\theta_6 - g_6 \cdot \omega_5 \cdot \sin(\theta_5 + \delta) \tag{16}$$

$$v_y^E = r_6 \cdot \omega_6 \cdot \cos\theta_6 + g_6 \cdot \omega_5 \cdot \cos(\theta_5 + \delta) \tag{17}$$

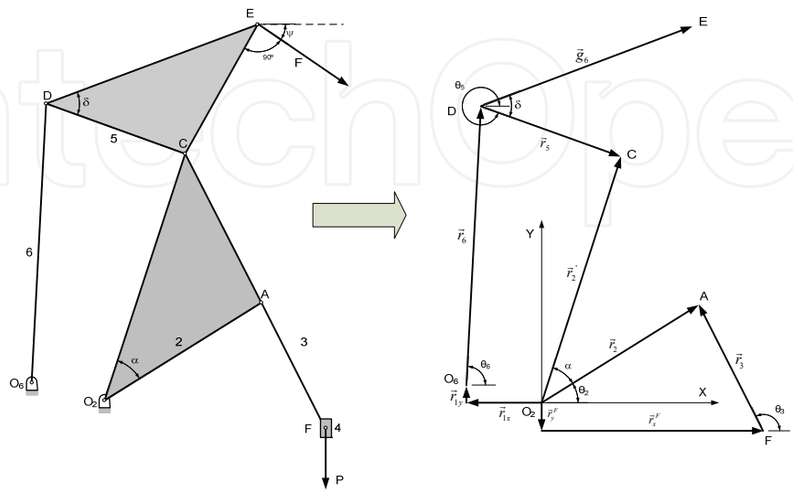


Figure 5. Kinematic study of the mechanism

And the acceleration:

$$a_x^E = -r_6 \cdot \omega_6^2 \cdot \cos\theta_6 - r_6 \cdot \alpha_6 \cdot \sin\theta_6 - g_6 \cdot \omega_5^2 \cdot \cos(\theta_5 + \delta) - g_6 \cdot \alpha_5 \cdot \sin(\theta_5 + \delta) \tag{18}$$

In the previous equations we have obtained all the variables that we need in the two problems proposed, but we also have to develop the following schemes:

$$\vec{r}_y^F + \vec{r}_x^F + \vec{r}_2 = \vec{r}_3 \tag{19}$$

$$\vec{r}_{1x} + \vec{r}_{1y} + \vec{r}_6 + \vec{r}_5 = \vec{r}_2^{'} \tag{20}$$

The first equation solves the slider-crank mechanism (2-3-4) and the second one solves the four-link mechanism (2-5-6), so the design variables for the proposed problems are:

$$X = \left\{ r_x^F, r_y^F, r_2, r_3, r_{1x}, r_{1y}, r_2^{'}, r_5, r_6, \alpha, \delta, g_6 \right\} \tag{21}$$

The design variables are the same for the two problems. The only difference is that in the second problem the variable r_x^F, what it is the actuator position, has different positions to obtain different contact point positions.

Second problem.- In this problem we will use the same hand robot mechanism (Figure 4), but in this case the mechanism will be able to grasp different objects with different sizes, i.e., the size of the object is within a determined range. Hence, in this problem the input slider has different positions that determine the different positions (precision points) of the output link. In this case the goal functions are:

1.- As the mechanism is in movement and the output link has different positions, i.e., the *E* contact point follows several precision points to determine the range of the size of the object,

we will try to make the *E* contact point follow a determined trajectory as well, so the first goal function is:

$$f_1 = \min\left(\sum_{i=1}^{n}\sqrt{\left(E_{x,i}^{obj} - E_{x,i}^{mech}\right)^2 + \left(E_{y,i}^{obj} - E_{y,i}^{mech}\right)^2}\right) \tag{22}$$

Where $E_{x,i}^{obj}$ and $E_{y,i}^{obj}$ are the *x* and *y* coordinates that the *E* contact point has to follow in each *i* position and $E_{x,i}^{mech}$ and $E_{y,i}^{mech}$ are the *x* and *y* coordinates of the *E* contact point of the designed mechanism in each *i* position. Hence, this goal function measures the error between the desired trajectory and the mechanism trajectory of the *E* point.

2.-The second goal function minimizes the grasping index (GI) developed in the previous problem, but applied to each *i* position of the *E* contact point, i.e, we obtain an average grasping index.

$$f_2 = \min\left(\frac{\sum_{i=1}^{n}\frac{v_{x,i}^{E} + v_{y,i}^{E}\cdot\tan\psi_i}{v_{y,i}^{F}}}{n}\right) \tag{23}$$

In this case we obtain v_x^E, v_y^E, v_y^F and ψ_i in each *i* precision point. And *n* is the number of precision points. The following goal functions are the same as in the previous problem:

$$f_3 = \min\left(\sum_{i=1}^{n} a_{x,i}^{E}\right) \tag{24}$$

$$f_4 = \min\sqrt{\sum_{i=1}^{n}\left(x_i^2\right)} \tag{25}$$

$$f_5 = \min\frac{\sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n}\left(x_i - x_j\right)^2}}{\sqrt{\sum_{i=1}^{n}\left(x_i^2\right)}} \tag{26}$$

In this problem the constraints are related to the velocity of the *E* contact point. This velocity, as in the previous problem, must be greater than zero, but in this case we have different *E* contact point velocities, so we have as many constraints as precision points.

$$\begin{aligned} g_1 &\equiv v_{x,1}^{E} > 0 \\ g_2 &\equiv v_{x,2}^{E} > 0 \\ &\vdots \\ g_n &\equiv v_{x,n}^{E} > 0 \end{aligned} \tag{27}$$

Once the problem has been defined, we can show it in the next equation:

$$\begin{aligned} &\min\{f_1(X), f_2(X), f_3(X), f_4(X), f_4(X)\} \\ &subject\ to: \\ &g_1(X) \\ &g_2(X) \\ &\vdots \\ &g_n(X) \\ &l_{\min} \le X \le l_{\max} \end{aligned} \tag{28}$$

As in the previous problem, the X vectors are the design variables.

5. Results

In the first place, we show the results of the first problem. In this case, the algorithm parameters are: (number of individuals in the population) *NP*=100, (maximum iteration number) *itermax*=5000, (disturbing factor) *F*=0.5, (crossover probability) *CP*=0.2, (mutation probability) *MP*=0, (initial number of non-dominated individuals) N_o=40, (non-dominated individual growth) ΔN=0.012, (size of the object) D_{mec}=100, (actuator velocity) v_y^F =-1, (actuator acceleration) a_y^F =1.

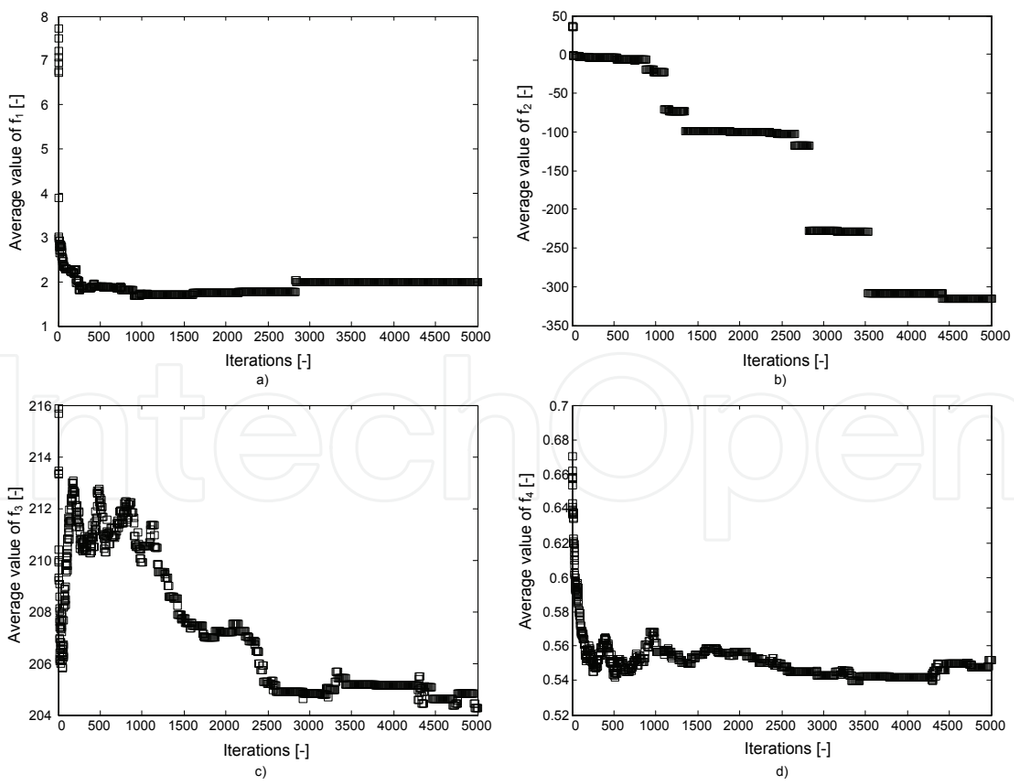


Figure 6. Average value evolution of the goal functions along iterations in the first problem

We show the average value evolution of the goal functions along iterations in Figure 6. The average values are obtained by the following equation:

$$f_{k,iter} = \frac{\sum_{i=1}^{NP} f_k(X_i^{iter})}{NP} \tag{29}$$

Where $k \in \{1,2,3,4\}$ are the numbers of goal functions in the first problem and NP is the number of individuals in the population.

We show the average value behavior of the goal functions in the previous figure. We can observe how the average values have decreased in every case. The average values are mean values of the goal functions of the all the individuals in the population and at the beginning of the iterations there are a few 'good individuals', i.e., non-dominated individuals, so those values are bad when the algorithm starts the iterations and they improve when the algorithm finishes. However, the algorithm does not always finish with the best average values as we can see in Figure 6a and 6d. This fact happens because the number of non-dominated individuals is not the total number of individuals in the population and the average values can be worse in the final populations because the dominated individuals in the final populations make the average value worse.

The non-dominated individuals' behavior can be observed in Figure 7. We can see how non-dominated individuals at the beginning of the iterations follow the established law, increasing the number of non-dominated individuals linearly with the iterations. At the end of the iterations, the number of non-dominated individuals is lower than the allowed non-dominated individuals. Hence the non-dominated individuals in the final populations are not the whole number of individuals in the population.

We also show the three 'best' mechanisms of the final population in the following figure. We draw the mechanisms that have the best value of one of the goal function values, but this does not mean that these mechanisms are the best mechanisms in the final population, as the final population has about eighty-four non-dominated mechanisms (see Figure 7).

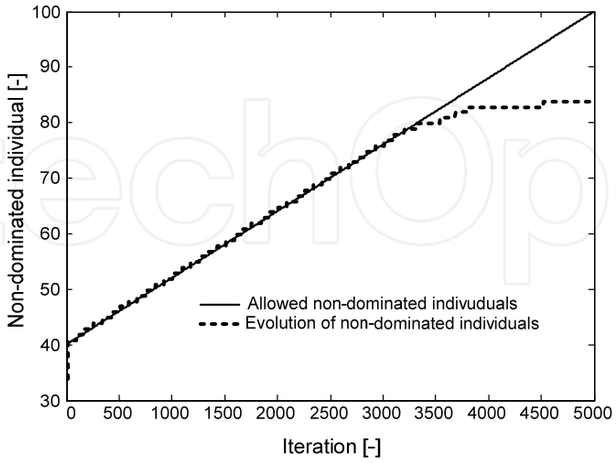


Figure 7. Evolution of non-dominated individuals along iterations

The design variable values of these three mechanisms are shown in the following table:

	r_x^F [L]	r_y^F [L]	r_2 [L]	r_3 [L]	r_{1x} [L]	r_{1y} [L]	r_2' [L]	r_5 [L]	r_6 [L]	α [-]	δ [-]	g_6 [L]
(a)	47.9	-15.3	34.2	42.7	-69.1	-17.4	81.8	63.4	124.6	1.3	1.38	134.4
(b)	50.2	-1.42	32.1	37.6	-41.4	-2.77	87.7	39.2	109.9	0.8	1.23	54.41
(c)	49.7	-10.9	33.9	35.1	-68.6	-20.6	80.5	55.7	93.21	1.2	0.84	117.8

Table 1. Design variable values of the three selected mechanisms in the first problem

And the values of every goal function for the three drawn mechanisms are shown in Table 2. The design variables and the goal function values of the three mechanisms correspond to the three mechanisms drawn in Figure 8. As we can see, mechanism (b) has the minimum value of the f_1 and f_2 functions, i.e., it has the minimum value of the contact point acceleration and the minimum value of its dimensions, but it has the worst value of grasping index f_1 and of link proportion f_4. Instead, mechanism (a) has the best grasping index and mechanism (c) has the best link proportion. Also, the contact point distances are shown in Figure 8. These distances are similar to the three cases and they are very close to our objective.

	f_1 [-]	f_2 [L/T²]	f_3 [L]	f_4 [-]
Mechanism (a)	0.4207	-0.1266	234.2706	0.4183
Mechanism (b)	1.7828	-901.79	175.9284	0.7677
Mechanism (c)	1.0479	-617.61	205.5393	0.3581

Table 2. Goal function values from three selected mechanisms in the first problem

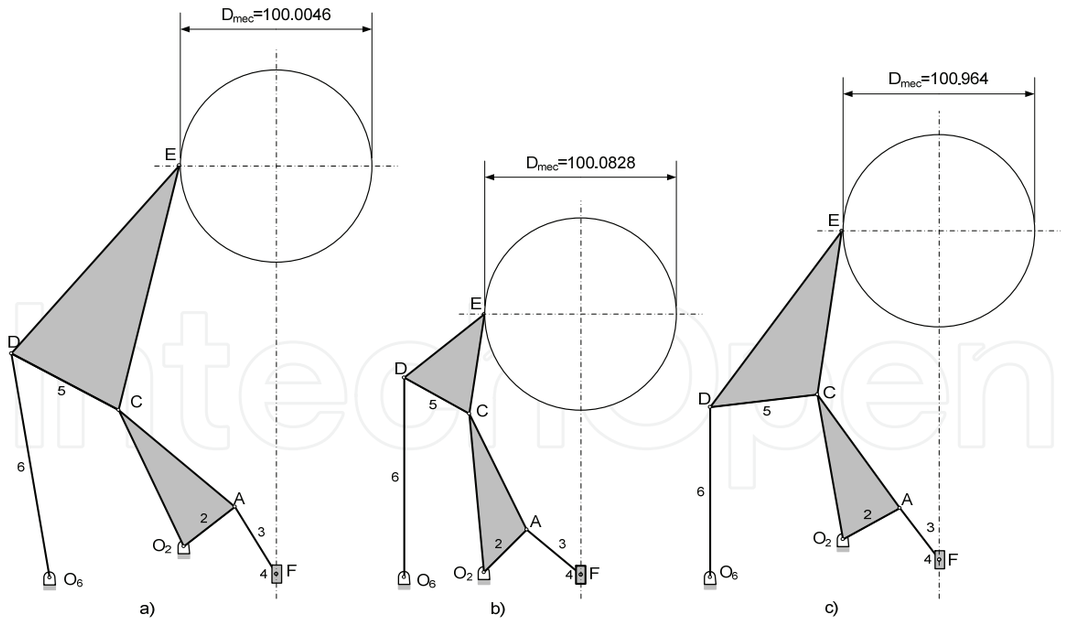


Figure 8. Three mechanisms of the final population in the first problem

We have to highlight that the three selected mechanisms are the ones with the best values of one of the goal function values, but this fact does not imply that these mechanisms are the best among the eighty-four non-dominated mechanisms. Hence, the designer will have to choose which mechanism among the non-dominated mechanisms is the best for him.

Now, we show the results of the second problem. In this case, the algorithm parameters are: (number of individuals in the population) NP=100, (maximum iteration number) $itermax$=5000, (disturbing factor) F=0.5, (crossover probability) CP=0.2, (mutation probability) MP=0, (initial number of non-dominated individuals) N_o=40, (non-dominated individual growth) ΔN=0.012, (actuator velocity) v_y^F =-1, (actuator acceleration) a_y^F =1.

Again, we show the average value evolution of the goal functions along iterations in Figure 9. The average values are obtained the same way as in the previous problem.

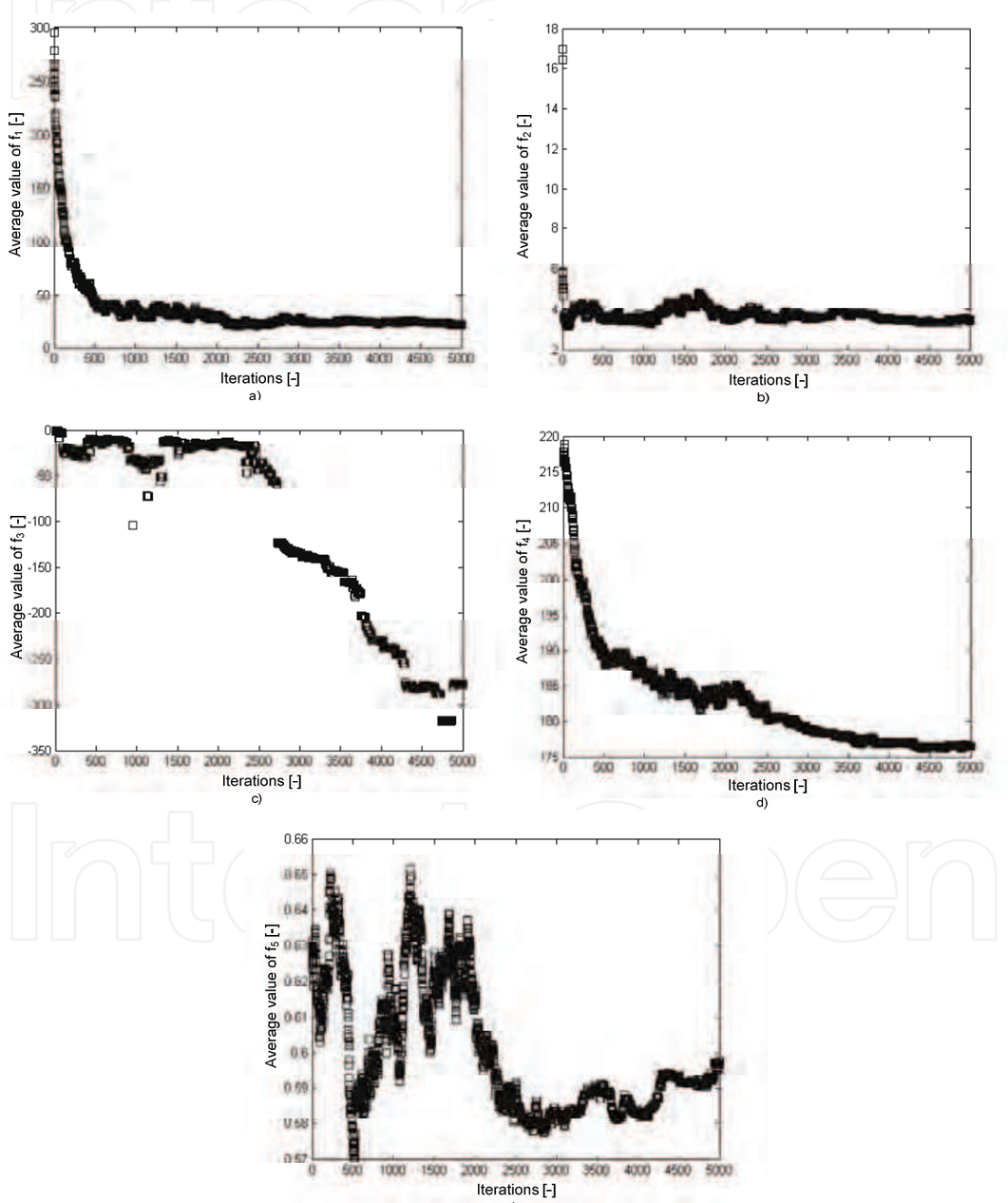


Figure 9. Average value evolution of the goal functions along iterations in the second problem

In this case, the average values of the goal functions also decrease along with the iterations in every case. Only the f_3 and f_5 goal functions have a different behavior when the algorithm finishes and the average values are worse than in the previous iterations. Instead, the new f_1 goal function has a clear decreasing behavior and the average value in the final iterations is the best one.

At the end we show three mechanisms of the final non-dominated population which have the best value of one of the goal functions (Figure 10). In this case, the mechanisms have to follow certain precision points:

$$E_x^{Obj} = \{-10, 10, 40\} \quad E_y^{Obj} = \{160, 170, 165\}$$

The coordinates of the previous precision points are measured from the O_6 fixed point in the mechanism. Hence the design variable values are:

	r_x^F [L]	r_y^F [L]	r_2 [L]	r_3 [L]	r_{1x} [L]	r_{1y} [L]	r_2' [L]	r_5 [L]	r_6 [L]	α [-]	δ [-]	g_6 [L]
(a)	22.5	-6.96 3.04 13.04	28.5	27.0	-57.6	-25.4	76.1	54.27	89.6	0.6	1.18	83.3
(b)	26.03	-12.6 -2.59 7.41	19.36	21.73	-45.2	-36.3	72.7	43.62	92.3	0.9	0.99	78.7
(c)	32.74	-19.3 -9.29 0.71	32.56	20.21	-20.9	-0.57	92.8	40.15	125.0	1.4	1.95	50.8

Table 3. Design variable values of three selected mechanisms in the second problem

The r_y^F value has three positions in the previous table because the *E* contact point is compared in these three positions of the input slider.

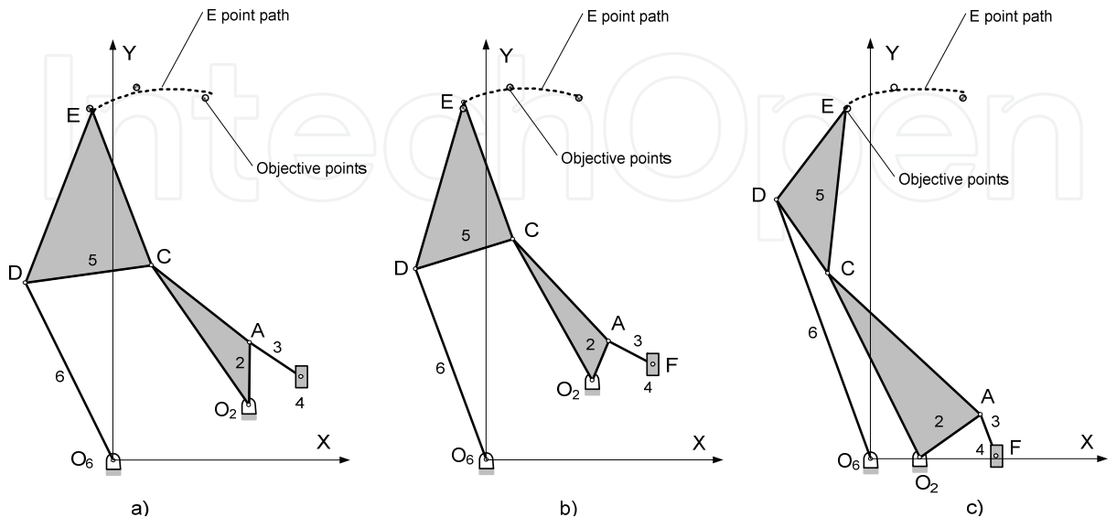


Figure 10. Three mechanisms of the final population in the second problem

We also show the goals function values of these three mechanisms.
Mechanism (a) has the best value of the f_3 and f_5 goal functions, i.e., this mechanism has the minimum value of *E* contact point acceleration and it has the best link proportion. Instead, mechanism (b) has the best f_1 and f_4 goal functions, so the *E* contact point path fits objective points more accurately and it also has the minimum value in its dimensions. Finally, mechanism (c) has best average f_2 grasping index.

	f_1 [L]	f_2 [-]	f_3 [L/T²]	f_4 [L]	f_5 [-]
Mechanism (a)	3.86	1.33	-0.4051	172.38	0.4125
Mechanism (b)	3.54	1.97	-0.1413	163.67	0.5620
Mechanism (c)	3.66	0.1412	-0.0638	177.21	0.9785

Table 4. Goal function values of three selected mechanisms in the second problem

6. Conclusions

In this paper, we showed a new algorithm (POEMA) based on the Differential Evolution strategy, but it has been extended to tackle multiobjective optimization problems. For this purpose, new features have been developed. This work uses the Pareto-based approach to classify the population into non-dominated and dominated individuals.
The algorithm is used to optimize several goal functions in a hand robot mechanism, subject to different constraints. The same method can be applied to optimize any other goal functions in other different problems.
One of the features of the used method is that there is not an unique solution to the problem, as the method finds several solutions which are called non-dominated solutions and every non-dominated solution is a good solution to the proposed problem. Hence, the designer must choose which is the best in every case, i.e., he must determine which characteristic or goal function is a priority and which is not.
An individual evolution study has been made and the obtained results have been satisfactory. We have shown several final mechanisms to the two proposed problems and each one has a good value of one o more features or goal functions.
Another advantage depicted by the method is its simplicity of implementation and that it is possible to use the method in other different mechanism problems by simply changing the goal function formulation for those problems.

7. References

Hrones, J.A. and Nelson, G.L. (1951). Analysis of the Four bar Linkage, MIT Press and Wiley, New York.

Zhang, C., Norton R.L. and Hammond, T. (1984). Optimization of Parameters for Specified Path Generation Using an Atlas of Coupler Curves of Geared Five-Bar Linkages, *Mechanism and Machine Theory* 19 (6) 459-466.

Sandor, G.N. (1959). A General Complex Number Method for Plane Kinematic Synthesis with Applications, *PhD Thesis*, Columbia University, New York.

Erdman, A.G. (1981). Three and Four Precision Point Kinematic Synthesis of Planar Linkages, *Mechanism and Machine Theory* 16 (5) 227-245.

Kaufman, R.E. (1978). Mechanism Design by Computer, *Machine Design Magazine*. 94-100.

Loerch, R.J., Erdman, A.G., Sandor, N., Mihda, A. (1975). Synthesis of Four Bar Linkages with Specified Ground Pivots, *Proceedings of 4th Applied Mechanisms Conference*, Chicago. 101-106.

Freudenstein, F. (1954). An Analytical Approach to the Design of Four-Link Mechanisms, *Transactions of the ASME* 76. 483-492.

Beyer, R. (1963). The Kinematic Synthesis of Mechanism, McGraw-Hill, New York.

Hartenberg, R. and Denavit, J. (1964). Kinematic Synthesis of Linkages, McGraw-Hill, New York.

Han, C. (1996). A General Method for the Optimum Design of Mechanisms, *Journal of Mechanisms*. 301-313.

Kramer, S.N. and Sandor, G.N. (1975). Selective Precision Synthesis. A General Method of Optimization for Planar Mechanisms, *Journal of Engineering for Industry* 2. 678-701.

Sohoni, V.N. and E.J. Haug, E.J. (1982). A State Space Technique for Optimal Design of Mechanisms, *ASME Journal of Mechanical Design* 104. 792-798.

Holland, J.H. (1973) Genetic Algorithms and the Optimal Allocations of Trials, *SIAM Journal of Computing* 2 (2) . 88-105.

Holland, J.H. (1975). Adaptation in Natural and Artificial Systems, The University of Michigan Press, Michigan.

Goldberg, D.E. (1989). Genetic Algorithms in Search, Optimization and Machine Learning, Addison Wesley, Massachusetts.

Rao, S.S. and Kaplan, R.L. (1986). Transaction ASME Journal of Mechanisms Transmission Automatic Des. 454-460.

Krishnamurty, S. and Turcic, D.A. (1992). Optimal Synthesis of Mechanisms Using Nonlinear Goal Programming Techniques, *Mechanisms and Machine Theory* 27 (5). 599-612.

Kunjur, A. and Krishnamurthy, S. (1997). A Robust Multi-Criteria Optimization Approach. *Mechanism and Machine Theory*, Vol. 32 No. 7 . 797-810.

Haulin, E.N. and Vinet, R. (2003). Multiobjective Optimization of Hand Prosthesis Mechanisms. *Mechanism and Machine Theory*, Vol. 38. 3-26.

Storn, R. and Price, K. (1997). Differential Evolution. A Simple and Efficient Heuristic Scheme for Global Optimization over Continuos Spaces, *Journal of Global Optimization* 11. 341-359

Cabrera, J.A. Simon, A. and Prado, M. (2002). Optimal Synthesis of Mechanisms with Genetic Algorithms. *Mechanism and Machine Theory* 37. 1165-1177.

Shiakolas, P.S., Koladiya, D. and Kebrle, J. (2005). On the Optimum Synthesis of Six-bar Linkages Using Differential Evolution and the Geometric Centroid of Precision Positions Technique. *Mechanism and Machine Theory* 40. 319-335.

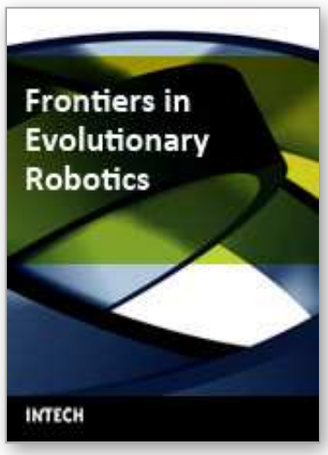
Wright, A.H. (1990). Genetic Algorithms for Real Parameter Optimization, *Proceedings of First workshop on the Foundations of Genetic Algorithms and Classifier Systems*, Indiana. 205-218.

Lampinen, J. and Zelinka, I. (2000). On Stagnation of the Differential Evolution Algorithm. *Proceedings of MENDEL 2000*, pp. 76-83. Brno, Czech.

Lampinen, J. (2002). A Constraint Handling Approach for the Differential Evolution Algorithm. *Proceedings of the 2002 Congress on Evolutionary Computation*, Vol. 2, pp. 12-17.

Abbass, H.A. (2002). The Self-Adaptive Pareto Differential Evolution Algorithm. *Proceedings of the 2002 Congress on Evolutionary Computation*, Vol. 1, pp. 831-836.

M. Ceccarelli, M. (1999). Grippers as Mechatronics Devices. *Advanced in Multibody Systems and Mechatronics*. 115-130.



Frontiers in Evolutionary Robotics

Edited by Hitoshi Iba

ISBN 978-3-902613-19-6

Hard cover, 596 pages

Publisher I-Tech Education and Publishing

Published online 01, April, 2008

Published in print edition April, 2008

This book presented techniques and experimental results which have been pursued for the purpose of evolutionary robotics. Evolutionary robotics is a new method for the automatic creation of autonomous robots. When executing tasks by autonomous robots, we can make the robot learn what to do so as to complete the task from interactions with its environment, but not manually pre-program for all situations. Many researchers have been studying the techniques for evolutionary robotics by using Evolutionary Computation (EC), such as Genetic Algorithms (GA) or Genetic Programming (GP). Their goal is to clarify the applicability of the evolutionary approach to the real-robot learning, especially, in view of the adaptive robot behavior as well as the robustness to noisy and dynamic environments. For this purpose, authors in this book explain a variety of real robots in different fields. For instance, in a multi-robot system, several robots simultaneously work to achieve a common goal via interaction; their behaviors can only emerge as a result of evolution and interaction. How to learn such behaviors is a central issue of Distributed Artificial Intelligence (DAI), which has recently attracted much attention. This book addresses the issue in the context of a multi-robot system, in which multiple robots are evolved using EC to solve a cooperative task. Since directly using EC to generate a program of complex behaviors is often very difficult, a number of extensions to basic EC are proposed in this book so as to solve these control problems of the robot.

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen

IntechOpen