Performance of Simple Genetic Algorithm Inserting Forced Inheritance Mechanism and Parameters Relaxation

Esther Lugo-González, Emmanuel A. Merchán-Cruz,
Luis H. Hernández-Gómez, Rodolfo Ponce-Reynoso,
Christopher R. Torres-San Miguel and Javier Ramírez-Gordillo
Instituto Politécnico Nacional Escuela Superior de Ingeniería Mecánica y Eléctrica
México

1. Introduction

Genetic Algorithms (GA) are powerful tools to solve large scale design optimization problems. The research interests in GA lie in both its theory and application. On one hand, various modifications have been made to allow them to solve problems faster, more accurately and more reliably.

Genetic Algorithms are a search paradigm that applies principles of evolutionary biology (crossover, mutation, natural selection) in order to deal with intractable search spaces. The power and success of GA are mostly achieved by the diversity with the individuals of a population which evolve, in parallel, following the principle of the survival of the fittest. In general, the genetic algorithms resolve combinatorial optimization problems that in (Goldberg, 1989) are mentioned, this implies a large number of responses associated with an exponential growth in solutions potentially feasible according to the magnitude of the problem. In a standard GA the diversity of the individuals is obtained and maintained using the genetic operators crossover and mutation which allow the GA to find feasible solutions and avoid premature convergence to a local maximum (Holland, 1975).

The performance of a genetic algorithm, like any global optimization algorithm, depends on the mechanism for balancing the two conflicting objectives, which are exploiting the best solutions found so far and at the same time exploring the search space for promising solutions. The power of genetic algorithms comes from their ability to combine both exploration and exploitation in an optimal way (Holland, 1975). However, although this optimal utilization may be theoretically true for a genetic algorithm, there are problems in practice. These arise because of Holland assumed that the population size is infinite, that the fitness function accurately reflects the suitability of a solution, and that the interactions between genes are very small (Beasley et al., 1993).

The evolutionary algorithm proposed in this paper is composed by a classic genetic algorithms along with the forced inheritance mechanism proposed by (Merchán-Cruz, 2005, Merchán-Cruz et al., 2008, Merchán-Cruz et al., 2007) and the regeneration mechanisms by

(Ramírez-Gordillo, 2010, Lugo González, 2010), for optimizing the trajectory generation in closed chain mechanisms and planning the effects that it has on the mechanism by relaxing some parameters. The objective is to show the behavior of relaxing the parameters of the GA's, observing what advantages and disadvantages appear when varying some parameter exceeding the recommended values established in the literature.

2. Genetic Algorithm description

Once the problem encoding and the fitness functions have been chosen, the evolution process begin. To evolve new solutions, an initial population of encoded solutions is created randomly or using some problem-specific knowledge. This population is subjected to genetic operators to create new promising solutions.

A typical genetic algorithm starts with a randomly generated population composed by genes, locus, allele, chromosome, genotype, variables and phenotype (Holland, 1975, Goldberg, 1989, Michalewicz, 1999, Coello-Coello, 2007), figure 1.

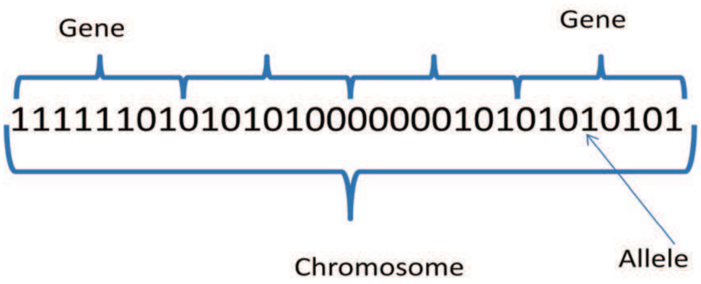


Fig. 1. Chromosome binary representation.

Individuals are probabilistically selected by evaluating the objective function. This gene has converged when at least 95% of individuals in the population share the same value of that genes. The population converges when all the genes have converged.

Different operators exist in GA´s, being the most popular (1) *selection*, (2) *crossover*, and (3) *mutation*, The steps to make a genetic algorithm, as defined in (Goldberg, 1989), are shown in the diagram of figure 2.

Initial Population is created randomly and it is encoded within the chromosome of an array with variable length. The coding can be done in a binary representation (Goldberg, 1989), based on the domain of each variable (figure 3).

In the *decodification* is necessary to have a representation of the genotype to assign the parameters within a chain of symbols known as genes. The *evaluation* uses the fitness function that reflect the value of the individual in terms of the real value of the variable in the problem's domain, but in many optimization combinatorial cases, where a great amount of restrictions exists; there is a probability in which part of the points of the search space represents no valid individuals. For example, the equation for the synthesis of planar mechanisms are:

$$F = \left(C_{xd}^{i}(v) - C_{xg}^{i}(v)\right)^{2} + \left(C_{yd}^{i}(v) - C_{yg}^{i}(v)\right)^{2} \tag{1}$$

Where C^{i}_{xd} is a set of specific points indicated by the designer and C^{i}_{xg} are the points generated by the coupler of the mechanism, and $v = r_1, r_2, r_3, r_4, r_{cx}, r_{cy}, \theta_0, x_0, y_0$, the angles $\theta_2^1, \theta_2^2, \ldots \theta_2^N$ are values for the variable θ_2, i is the rest of the quotient. The genetic algorithm maximizes solely, but the minimization can be made easily using the reciprocal of the function to avoid singularity problems (2):

$$fitnessoptimum = \frac{1}{fitness} \tag{2}$$

In order to improve the results, approaches such as elitism, regeneration stages and the forced inheritance mechanism can be inserted in the process of the algorithms:

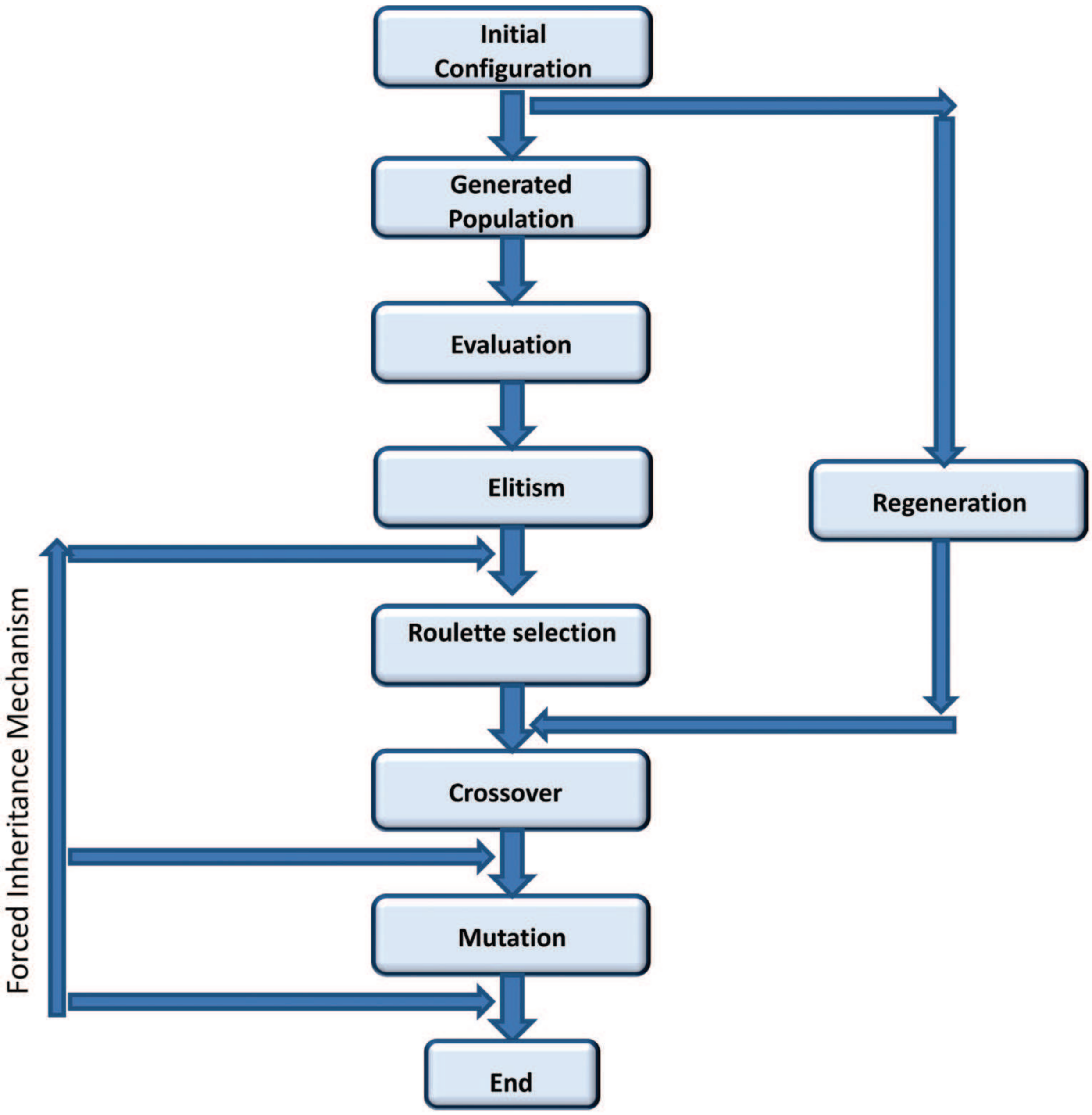


Fig. 2. Flowchart of genetic algorithms.

Domain=[-60 60 -60 60 0 60 ……………0 360]

$[X_{0min}\ X_{0max}\ \ Y_{0min}\ Y_{0max}\ldots\ldots\ldots\ \theta_{min}\theta_{max}]$

Fig. 3. Structure Chromosome.

Elitism: In this case the best individual of the population at a certain time is selected like father, this reserve two slots in the next generation for the highest scoring chromosome of the current generation, without allowing that chromosome to be crossed over in the next generation. One of those slots, the elite chromosome will also not be subject to mutation in the next generation.

*Regeneration Mechanism***:** The investigations on some alive organisms that use strategies for their renovation in physiological conditions or before a damage, demonstrate the possibility of incorporating cells that appear and which are specialized in providing reserve cells of an adult organism, thanks to a particular hereditary mechanism and, under this condition, the algorithm can be considered like an evolutionary process within the population. Therefore, a small percentage of the population can be renewed, which allows increasing the formation of construction blocks with better possibilities of finding an optimal value, but as inconvenient the problem of premature convergence of an evolutionary algorithm explained by (Hidalgo and Lanchares, 2000) and (Wen-Jyi et al., 2003) is presented. Nevertheless, the biological evolution process and its mimetization, can validate the use of a regeneration factor and its fundamental preservation in the genetic operators of selection, crosses and mutation.

Forced Inheritance Mechanism: Proposed by (Merchán-Cruz, 2005), is a complementary part of the regeneration mechanism as a strategy to introduce specialized chromosomes on the basis of the elitism during the crossing process and mutation. Unlike elitism, where the aptest individuals of a population pass to the following generation without no alteration, the FIM is introduced in the process of regeneration, selection, crossover and mutation, guaranteeing that the aptest individual of the previous generation undergoes a minimal change increasing its aptitude value of consistent method. This mechanism is very useful when the number of variables to solve in the problem is considerably large.

In the same way that the best obtained chromosome is carried among generations in a simple GA, the best set of chromosomes is also carried to the GA search for the next trajectory parameters. By introducing the best set of chromosomes from the previous trajectory segment of the initial population of the current GA search, the required number of generations to produce a new trajectory segment is reduced, provided that the trajectory is stable in that particular instant, since the optimum or the near optimum solution is already coded into the initial population. If the mechanism has to change its trajectory due to kinematic constrains or any other circumstance, the carried set of chromosomes does not affect the search for a new optimum set since this one is evaluated and ranked accordingly to its corresponding fitness. Figure 4 illustrates this mechanism called Forced Inheritance Mechanism, FIM, (Merchán-Cruz, 2005).

The necessary operations for regeneration and the forced inheritance are:

1. Percentage of the population to regenerate.
2. Chose again the number of individuals, the length of the chromosome and therefore the size of the population.

3. Regeneration takes the value from the individuals by the percentage to be regenerated.
4. This population is converted to binary representation.
5. The position that will occupy the regenerated ones in the original population is determined without altering the number of individuals.
6. Reinsert the regenerated population into a sector of the original population.
7. The best individual in the regenerated population introduces itself, looking forward not to alter the number of individuals.

Following the development of the genetic algorithm, taking the best individuals from the population will pay the selection of those who have been outfitted as parents of the new generations.

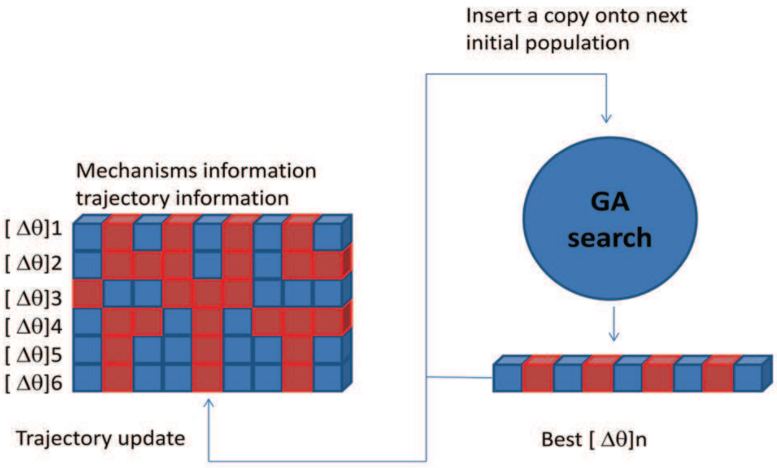


Fig. 4. Forced Inheritance Mechanism (Merchán-Cruz, 2005).

For the Parent Selection exists several techniques, but the most used is the proportional selection proposed by (Goldberg, 1989) in this each individual have a probability of being selected like parents, that is proportional to the value estimated by means of the objective function.

Crossover is based on taking two individuals correctly adapted to obtain descendants that share genes of both. There are several types of crossover mechanisms that are used depending on the scheme that is analyzed. According to (Kuri-Morales and Galaviz-Casas, 2002) the most popular are: single point crossover, two points and uniform crossover.

Mutation is an operator that is applied with probability *pm* and has the effect to invert a single bit using a probability of mutation of bit l^{-1}, being l the length of the chain of the chromosome.

While crossover needs large populations to effectively combine the necessary information, mutation works best when applied to small populations during a large number of generations. Mutation is usually a secondary search operator which performs a random search locally around a solution and therefore has received far less attention. However, in evolutionary strategies where crossover is the primary search operator, significant attention has been paid to the development of mutation operators. Several mutation operators, including adaptive techniques, have been proposed by (Lima, 2005). Clearly, mutation cannot perform this role as well as crossover.

By other hand *Crossover Probability* indicates how often will be crossover performed. If there is no crossover, offspring is an exact copy of parents. If there is a crossover, offspring is made from parts of parent's chromosome. If crossover probability is 100%, then all offspring is made by crossover. If it is 0%, a whole new generation is made from exact copies of chromosomes from old population (but this does not mean that the new generation is the same). Crossover is made expecting that new chromosomes will have good parts of old chromosomes and perhaps this will be better. However it is good to allow some part of the population survive to next generation.

Mutation probability says how often will be parts of chromosome mutated. If there is no mutation, the offspring is taken after crossover (or copy) without any change. If mutation is performed, part of a chromosome is changed. If mutation probability is *100%,* whole chromosome is changed, if it is 0%, nothing is changed. Mutation is made to prevent falling GA into local extreme, but it should not occur very often, because then GA will in fact change to *random search.*

Each operator allows that the evolutionary process progress toward promising regions in the area of search and can carry on diversity within the population and inhibit the premature convergence to an optimal local by means of new individuals sampled randomly. On the other hand is required to manipulate the information through a metric that quantifies the evolutionary process, this can be done through the design of a function that gets the more suitable individuals. This metric is known as a function of ability and it increases the ability of this individual to operate with a good performance and to get an unbeatable quality.

Problems typically contain restrictions, such as the non-linearity and inequality, which makes necessary to incorporate information on the violation of restrictions on some of the functions and the most known are *Criminalization role,* this restricts the fitness role by extending its domain by a factor of criminalization to any restriction raped. It can penalize for not being feasible or to make feasible an individual. The penalty function design must take into account how distant is an individual from the feasible area, the cost of fulfillment and the cost of expected compliance. Some of these penalties are:

- **Death Penalty.** It assigns a suitability of zero to the individual not feasible, avoiding calculate again restrictions or objective function. However, the algorithm may be truncated if the initial population does not contain any feasible individual.
- **Static Criminalization.** It defines levels of violation and chooses a coefficient of violation to each one of them.
- **Dynamic Criminalization.** The factors of criminalization change with time; they are susceptible to the values of the parameters and converge prematurely when these are not selected properly.
- **Adaptive Criminalization:** Adjusting the penalty on the basis of a feedback process.

The adaptive criminalization is used in this work.

2.1 Efficiency enhancement of GA

Goldberg categorized the efficiency enhancement techniques of GA into four broad classes: parallelization, hybridization, time continuation, and evaluation relaxation (Goldberg, 2002):

1. **Parallelization:** GAs are executed on several processors and the computational load is distributed among these Processors (Cantu-Paz, 2000). This leads to significant speed-up when solving large scale problems. Parallelization can be achieved through different ways. A simple way is to have part of the GA operations such as evaluation running simultaneously on multiple processors (Bethke, 1976). Another way is to create several subpopulations and allow them evolve separately at the same time, while spreading good solutions across the subpopulations (Grosso, 1985).
2. **Hybridization:** Local search methods or domain-specific knowledge is coupled with GA. This are powerful in global search. However, they are not as efficient as local search methods in reaching the optimum on micro-scale. Therefore, hybridization which incorporates local search methods into GA will facilitate local convergence. A common form of hybridization is to apply a local search operator to each member of the population after each generation in GA (Sinha, 2002).
3. **Time Continuity**: The capabilities of both mutation and recombination are utilized to obtain a solution of as high quality as possible with a given limited computational resource (Srivastava, 2002). Time continuation exploits the tradeoff between the search for solutions with a large population and a single convergence epoch or using a small population with multiple convergence epochs.
4. **Relaxation Evaluation**: An accurate, but computationally expensive fitness evaluation is replaced with a less accurate, but computationally inexpensive fitness estimate. The low-cost, less-accurate fitness estimate can either be 1) exogenous, as in the case of surrogate (or approximate) fitness functions, where external means that it can be used to develop the fitness estimate; or 2) endogenous, as in the case of fitness inheritance (Smith, 1995) where the fitness estimate is computed internally and is based on parental fitness.

3. Adjustment in the performance of the parameters of the GA

Some authors such as (Holland, 1975), have looked into the effect of varying GA's parameters which have to be taken into account to exploit the full potential in particular applications. Accordingly to this, for a search algorithm to perform well online, one has to decide quickly which are the most promising search regions in order to concentrate the search efforts there, the off-line performance does not penalize the search algorithm to explore poor regions of the search space, provided that this will help to achieve the best possible solutions (in terms of fitness), abig generation interval and the use of an elitist strategy also improve the performance of the GA's, in which the usual recommended mutation rates between 0.001 and 0.01 for the binary representation (Goldberg, 1989), or in general, much smaller value of the crossover probability (Cabrera et al., 2002).

The main parameters that can be adjusted, by the degree of importance within the GA are:

- Population size
- Percentage of crosses
- Percentage of mutation

The design of the algorithm is limited to choose and determine the degree of control or the strategies of parameters such as the ranges and the likelihood of a mutation, crossing and extent of the population. (Sanchéz-Marín, 2000) supported their research in the determination of control parameters, experimenting with different values and selecting those that gave better results. (De Jong, 1975) recommended, after experimenting, values for the probability of the interbreeding of simple point and the movement of a bit in the mutation. In this work, the following parameters are defined: a population-based measure of 50 individuals, probability of crossing 0.6, probability of mutation of 0.001 and elitist selection; however, it presents the disadvantage that these parameters only worked for a particular problem with very specific restrictions.

(De Jong, 1975) described that the operation on-line is based on the monitoring of the best solution in every generation, while the operation off-line takes into account all the solutions in the population to obtain the optimum value. (Grefenstette, 1986) used the meta-algorithms as a method of optimization, in order to obtain values with similar parameters for the operation on-line and off-line of the algorithm.

In order to have a good performance on-line of a search algorithm, it must quickly decide where the most promising search region is and concentrate their efforts there. The performance off-line does not always penalize the search algorithm to explore poor regions of the search space, since this will contribute to achieving the best possible solutions (in terms of fitness). The best sets of parameters analyzed on and off- line were population of 30 and 80 individuals, probability of crossing 0.95 and 0.45, probability of mutation 0.01 for both, either using a strategy of elitist selection for the on-line case or not elitist for the off-line case .

(Smith, 1993) proposes a genetic algorithm which adjusts the extent of the population taking into account the likelihood of error. This is linked with the number of generations, if under the conditions of little use is determined a small value (20 to 50) to the number of evaluations, the convergence will be quick, but it is not ensured an optimum result.

(Endre Eiben et al., 1999) expose technical drawbacks of the analysis of parameters on the basis of experimentation, observing the following points:

- Parameters are not independent, but trying all possible combinations of these systematically it is almost impossible.
- The process of tuning parameters is time-consuming, but if the parameters are optimized one by one, it is possible to handle their interactions.
- For a given problem, the values for the selected parameters are not necessarily the best, but if they are used to analyze uniformly, more meaningful values will be obtained.

In general, here are listed some important observations made by authors such as (Holland, 1975),with respect to the genetic algorithms that must be considered for the use of this tool, such as:

- A high generational interval and the use of an elitist strategy also improve the performance of the GA.
- The use of large populations *(> 200)* with a high percentage of mutation *(> 0.05)* does not improve the performance of a GA.
- The use of small populations *(< 20)* with low percentage of mutation *(< 0.002)* does not improve the performance of a GA.
- The mutation seems to have greater importance in the performance of a GA.
- If the size of the population is increased, the effect of crosses seems to be diluted.

With reference to the mutation, it has been deeply analyzed the value of the probability, but the results vary with each researcher, for example (De Jong, 1975) recommend *pm=0.001*, (Grefenstette, 1986, Goldberg, 1989) recommend *0.1*, (Fogarty, 1989) indicates *0.005 to 0.01*.

In the research of (Fogarty, 1989) and (Coello-Coello, 2007) have been developed some formulas in order determine the mutation, where its main contribution is considering the time and making a change of this during the execution of the GA. If the mutation percent is *0*, does not exist any alteration, if is *1*, the mutation creates always add-ons of the original individual and if it is 0.5, there is a high probability of altering strongly the schema of an individual. In conclusion, it is possible to control the power of alteration of the mutation and its capabilities for exploration, to have an equivalent weight within the AG as the crossing.

On the other hand for the crossing some common values for this are 0.6 indicated by (De Jong, 1975), *0.95 by* (Grefenstette, 1986), *0.75 to 0.95 by* (Fogarty, 1989). (Endre Eiben et al., 1999) specify that is more common to use the results obtained in own experimentation and is rarely used a value less than 0.6. When it is looking for locating the global optimum of a problem, the mutation may be more useful, but when it is in the cumulative gain, crossing offers greater benefits. From these research works it can be said that there are needs of large populations in the crossing, to combine effectively the necessary information, but in mutation best results are obtained when applied to small populations in a large number of generations.

Evolutionary strategies, where the mutation is the principal search operator, include several operators of mutation, as well as technical adaptation, proposed by (Lima et al., 2005, Rechenberg, 1973). (Whitley et al., 1998) reported comparative studies between the operators of crossover and mutation, demonstrating that there were important features of each operator that were not captured by the other.

In this work is demonstrated, through experimentation, that the maximum limit for individuals have an acceptable performance of the GA is 3000, this depends completely on study cases, since as it increases the number of variables in the problem to be analyzed it is necessary an increase in the population. With this amount of individuals the process of analysis is very slow, but it is in direct function of the mechanism type, the trajectory and the precision points required, in addition to the restrictions on the domain to get the angles and the links dimensions.

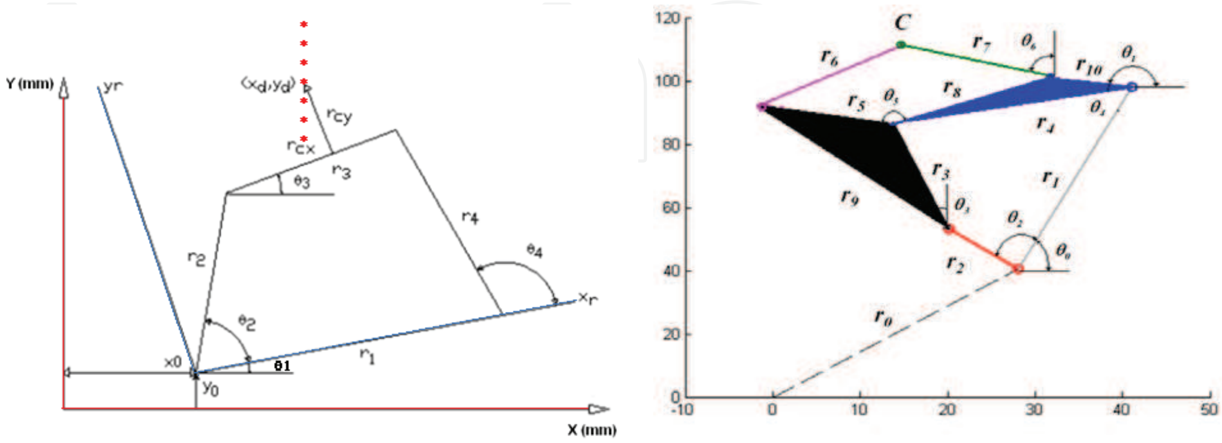
4. Study cases

The case of study is based on mechanisms synthesis, for that reason the basic concepts are presented.

A mechanism is a set of rigid members that are jointed together in order to develop a specific function. The mechanisms design, which is described by (Varbanov et al., 2006), consists of two parts: the analysis and synthesis. The first one consists of techniques to determine position, velocities and accelerations of points onto the members of mechanisms and the angular position, velocities and accelerations of those members. The second type explains the determination of the optimal length of the bars and the spatial disposition that best reproduces the desired movement of the coupler link. The optimal dimensional synthesis problem of mechanisms can be seen as a minimization process, since it is required that the structural error being as small as possible. The point of the coupler link will have to be able to generate a trajectory defined through separate points, with a minimum error. The generation of a desired trajectory consists controlling a point of the coupler link, figure 5 (case I four-bar mechanism and case II six-bar mechanism), so that its described trajectory drives the coupler through a discreet set of giving points, known as precision points (Norton, 1995). In order to determine this point it is necessary to obtain the open and close chain mechanism.

In the last century, have been developing a variety of mechanisms synthesis methods. These are usually based on graphical procedures originally developed by (Freudenstein, 1954); or on analytical methods of research of (Denavit and Hartenberg, 1964). Other techniques include the application of least squares in the finite synthesis of four-bar spatial synthesis proposed by (Levitski and Shakvazian, 1960), or on the mathematical model and simulation for the exact mechanisms synthesis as is described in (A. K. Mallik and A. Ghosh, 1994) and (Tzong-Mou and Cha'o-Kuang, 2005). However while these works have represented major contributions in the area, the principal restriction are the number of points of precision that can be taken into account to define the desired path. The foregoing refers to the fact that each point of precision defined for the desired path represents a new set of equations to be solved. For example, the synthesis of a four- bar mechanism involves a set of 7 holonomics restrictions that describe the kinematic relationship of the links that make it up; if the designer consider 4 points of accuracy, the problem to be solved is a set of 28 non-linear equations with 29 unknowns, which represents a non-linear indeterminate problem with an infinite number of possible solutions.

With all these arguments in mind and taking into account that exist a wide variety of applications that require a large number of precision points to define more accurately the trajectory to be reproduced by the mechanism, the synthesis of these can be seen as an optimization multi-objective problem. For this purpose, researchers have developed different methodologies that include non-linear optimization (Levitski and Shakvazian, 1960), genetic algorithms (Quintero-R et al., 2004, Laribi et al., 2004, Cabrera et al., 2002, Michalewicz, 1999, Roston and Sturges, 1996), neuronal networks (Vasiliu and Yannou, 2001), (Starosta, 2006), (Walczak, 2006)), Monte Carlo optimization (Kalnas and Kota, 2001), or the controlled method (Bulatovic and Djordjevic, 2004). All the above methods have been used for four-bar mechanisms synthesis and have helped to identify the constraints of space that lead to the synthesis of mechanisms and programs developed for applications.



a) Coupler point on the coupler link of a four bar linkage

b) Type 6-bar Watt mechanism.

Fig. 5. Diferent mechanisms configuration.

4.1 Optimal design in the mechanisms synthesis

The formulation of this problem demands the definition of several aspects like the space of design, the objective function, the algorithm of optimization and the restrictions (Lugo-Gonzalez et al., 2010). In the case corresponding to the synthesis of mechanisms, it is desired to diminish the error between the desired and generated trajectories besides analyzing the changes in the response of the algorithm when modifying parameters like the probability of mutation and crossing, the number of individuals and the maximum of generations, that will be evaluated by the proposed equation (3) that has characteristics applied to the approximated evaluation of the function, that involves the addition of the penalty to the presented original version in the works of (Goldberg, 1989), which is:

$$F = \frac{1}{N} * \sqrt{\frac{\sum_{i=1}^{N} \left(C_{xd}^{i}(v) - C_{xg}^{i}(v) \right)^2 + \left(C_{yd}^{i}(v) - C_{yg}^{i}(v) \right)^2}{ni}} \tag{3}$$

Applying a division of the number of individuals *ni* in addition to a factor of division by the reciprocal of *N,* that is the number of precision points, it adds a penalty whose objective is to recover the individuals that do not fulfill the initial restrictions known as the conditions of Grashof.

In order to finalize, the optimization algorithm uses four criteria of convergence that are defined as:

reng= Is the first restriction, this one is the first evaluation in which it is verified if the population fulfills the restrictions of Grashof (specific condition of mechanism synthesis).

maximogen = Defines the maximum number of times that the algorithm can evaluate the objective function. An additional call to this implies the conclusion of the search without reaching a solution.

minimerror = Defines the minimum value of error allowed in the objective function to being compared with the generated function. A change of value in the parameter of minimum error implies the conclusion of the search without reaching a solution.

condrep = Defines the number of times that the same value can be repeated into the evaluation before proceeding to the following operation.

Being fulfilled these last conditions to the evaluation; the algorithm will stop its search having presented the optimal values that better satisfy the restrictions and conditions.

4.2 Elliptical trajectory with parameters optimization of GA, 18 precision points, and a four- bar mechanism

The obtained research results in (Cabrera et al., 2002, Laribi et al., 2004, Starosta, 2006) are taken as a basis for describing an elliptical path with a four-link mechanism. The study case was proposed for the first time by (Kunjur and Krishnamurty, 1997). The synthesis was carried out using some variants of application using genetic algorithms or combining these with tools such as fuzzy logic. In the table 1 is shown the desired precision points to be followed by the mechanism. In the figure 6 is showed the corresponding graphic.

Point	1	2	3	4	5	6	7	8	9
X	0.5	0.4	0.3	0.2	0.1	0.005	0.02	0.0	0.0
Y	1.1	1.1	1.1	1.0	0.9	0.75	0.6	0.5	0.4
Point	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**
X	0.03	0.1	0.15	0.2	0.3	0.4	0.5	0.6	0.6
y	0.3	0.25	0.2	0.3	0.4	0.5	0.7	0.9	1.0

Table 1. Precision points of desired elliptical trajectory by(Kunjur and Krishnamurty, 1997).

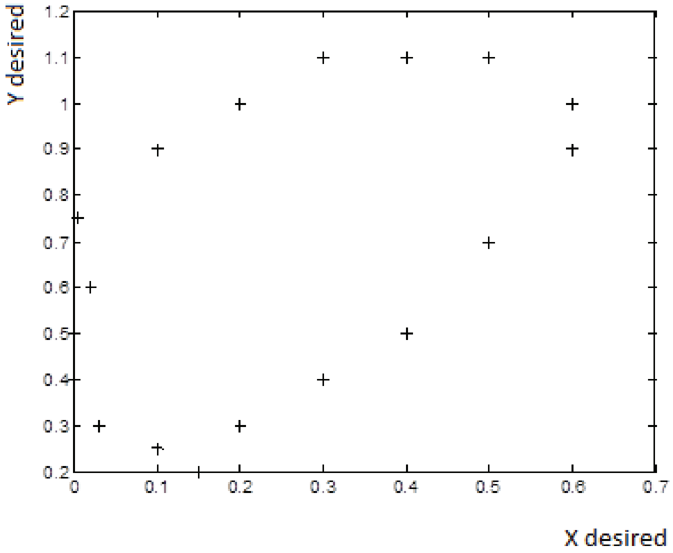


Fig. 6. Elliptical trajectory, by(Kunjur and Krishnamurty, 1997).

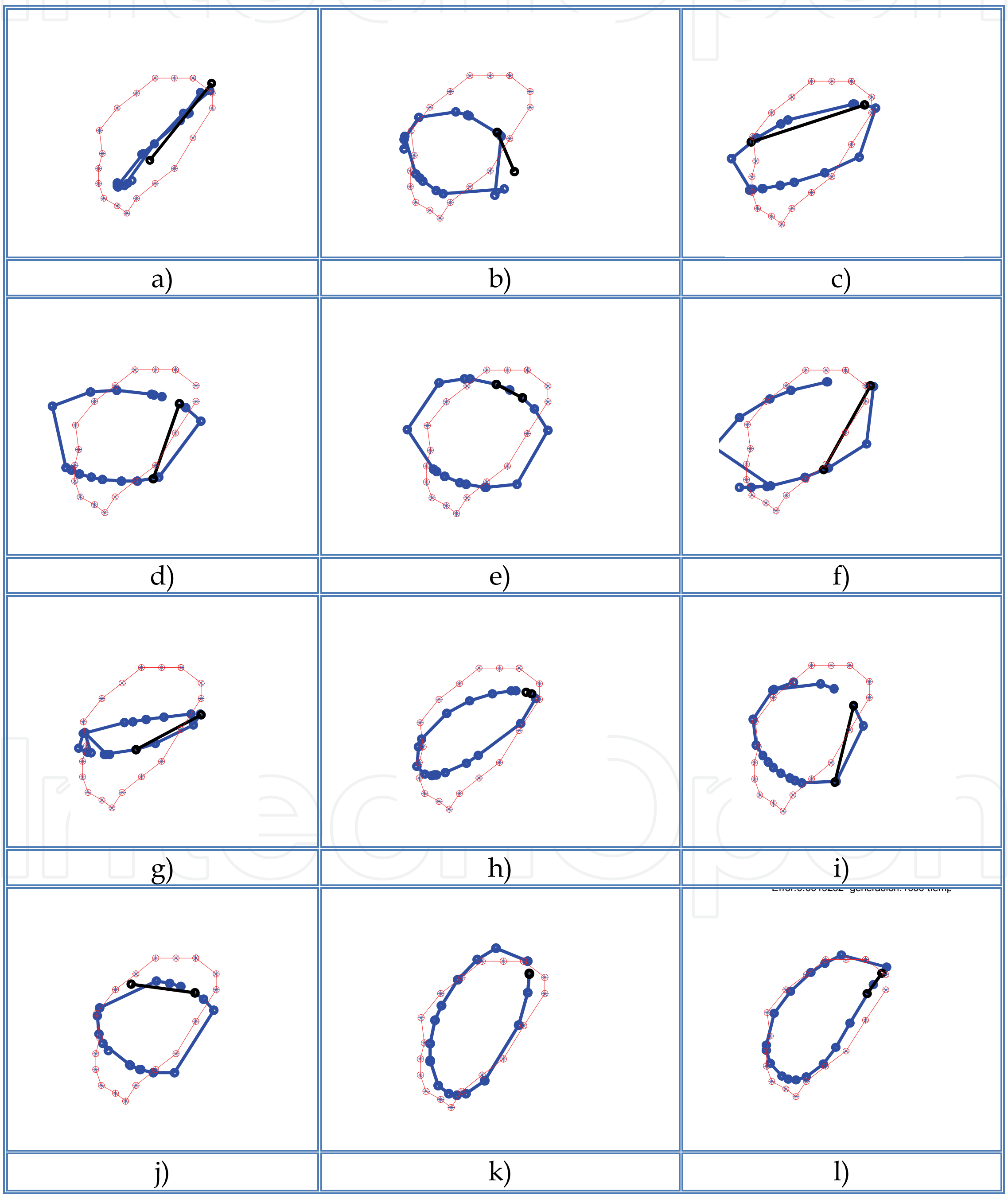
The realized parameters change is presented in table 2 and the results obtained by (Kunjur and Krishnamurty, 1997, Cabrera et al., 2002, Laribi et al., 2004, Starosta, 2006) and our results are shown in table 3. The analysis procedure is shown in figures 7 and 8. But this indicates that is necessary to make a change of value in the parameters of crossover and mutation. The changes are the number of individuals, crossover and mutation, affecting with this time and the number of generations for the convergence. It has a maximum number of 1500 generations and a precision of 6 digits.

Of this series of tests one concludes that:

The individual number is an important factor for the convergence, since although a response time with a small number of individuals is obtained, it does not make sure that the result is the optimal one. With a greater number of individuals the response time increases but the possibility of obtaining a better result also increases. As it has been mentioned previously, the program will have an optimal rank of individuals to operate satisfactorily, but this must be verified by trial and error, being a program that has as a basis the random generation of the population. However, the performance of the algorithm when the FIM is implemented only registers a minor reduction compared with the one obtained for the previously considered systems.

Do not exist a rule to determine the optimal value for the crossing and the mutation probability. Not always the maximum values, that produce a total change in the individual, give the best results, as it is observed in table 2. For this study case, the best result appears in interjection ***k*** with the value of minimum error. This value is affected directly by the dominion of the variables and in addition to the number of individuals.

The dominion is a determining value to obtain the optimal result, since all the variables are related to each other by the calculations required for the synthesis. For example if the restriction of some angles are modified, it changes the value of lengths of the links and by consequence the value of the error, since perhaps the bars must increase or decrease them length to cover the specified trajectory. Although the parameters are designed well, if this definition of variables are incorrect, it does not fulfilled the objective.



a) b) c)

d) e) f)

g) h) i)

j) k) l)

Fig. 7. Different parameters in elliptical trajectory.

	Population number	*Pc*	*Pm*	*Error*	*Generation*	*Time (S)*
a)	500	0.6	0.01	0.12607	974	124.70116
b)	1000	0.6	0.01	0.116874	992	200.83390
c)	1000	0.8	0.8	0.146653	994	281.61255
d)	1000	0.8	0.7	0.140036	992	318.54909
e)	1500	0.8	0.7	0.128140	992	499.81671
f)	1500	0.85	0.85	0.113155	979	384.667514
g)	2000	0.3	0.1	0.253548	992	416.934751
h)	2000	0.6	0.2	0.2020683	988	374.043950
i)	2000	0.6	0.4	0.1852588	991	431.125409
j)	2000	0.7	0.2	0.0986130	988	335.516387
k)	2000	0.7	0.4	0.0854537	995	402.078071
l)	2000	0.85	0.85	0.09922667	989	776.17100

Table 2. The parameters modification for a generated elliptical figure by a four bar mechanism.

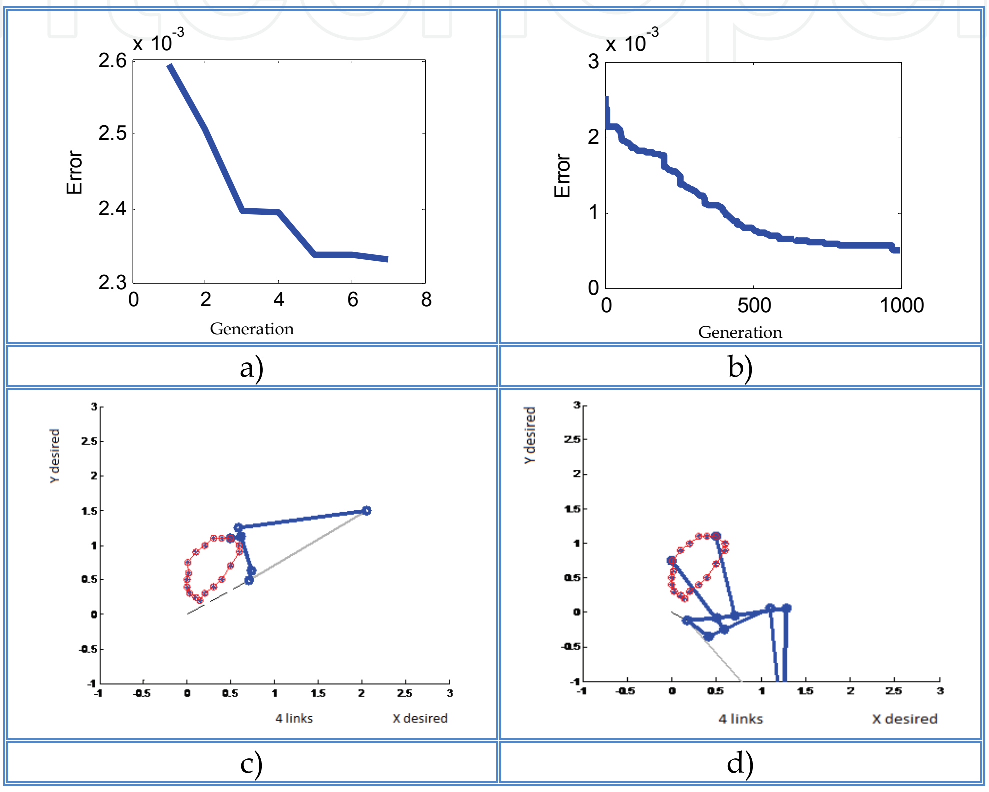
In table 3 is presented the comparison of the researchers mentioned above with the proposed algorithm. With these results it can be seen that there is a correspondence of values in the bars length, angles and among desired and generated trajectory. Another variable not found in the mentioned investigation is time, a factor that is critical for the optimization. This will depend on the crossover probability, mutation parameters, individuals and generation number. Varying a small value to these parameters can mean a short time in convergence but not always the optimal value is guaranteed. With the specific parameters, the time elapsed by the GA optimization analysis is 280.508318 seconds.

Autor	*Xo*	*Yo*	*R2*	*R1*	*R4*	*R3*	*R5*
Kunjur	1.132062	0.663433	0.274853	1.180253	2.138209	1.879660	0.91
Cabrera	1.776808	-0.641991	0.237803	4.828954	2.056456	3.057878	2
Laribi	-3.06	-1.3	0.42	2.32	3.36	4.07	3.90
Starosta	0.074	0.191	0.28	0.36	0.98	1.01	0.36
A-G prop.	3.88548	0.907087	0.286753	4.52611	3.59121	4.29125	3.613847
Autor	*Error*	*No. Eval.*					
Kunjur	0.62	5000					
Cabrera	0.029	5000					
Laribi	0.20						
Starosta	0.0377	200					
A-G prop.	0.0152	2000					

Table 3. Dimensions and angles definition of an elliptical path obtained by some authors.

In spite of applying more generations that in the last researcher work, satisfactory results are obtained. Due to the high amount of generations, computation time is more demanding, but this offers less error among the generated and desired path, and therefore, greater precision.

Figure 8 shows how the error behavior decreases at the beginning of the path and at the end of the evaluation in each generation (*a* and *b*). Figures *c* and *d* illustrates the four-bar mechanism along the path, covering the first and sixth precision point, which were randomly chosen to display the specified path.



a)

b)

c)

d)

Fig. 8. Four-bar mechanism evolution to cover 18 precision points.

4.3 Six bar mechanisms optimization

There are two main six-link configuration mechanisms Watt and Stephenson type, whose features make them suitable for the manufacture of polycentric prostheses such as (Radcliffe, 1977, Dewen et al., 2003).

The first example illustrates a six-bar mechanism for covering 21 precision points. To evaluate the effectiveness of the analyzed mechanism a Watt-I type will follow a path with arbitrarily proposed restrictions on the initial 18 points, being the conditions reported in table 4.

Figure 9 presents the proposed path to be followed by the Watt-I type mechanism. As in the previous cases, settings in population, crossing and mutation probability, time and number of generation analysis, will be varied in order to demonstrate that these adjustments are not independent and that they are affected each other.

Polycentric Mechanisms	**Description**	
Characteristic		
Desired points **Variables limits**	x_d=[25 10 5 10 20 10 5 10 15 25 40 43 50 55 50 40 50 55 50 40 25]	
	y_d=[[130 120 100 80 65 55 35 20 15 10 10 15 20 40 55 65 80 100 120 130 130]	
Restriction for each links	*r1,r2,r3,r4,r5,r6,r7,r8,r9,r10∈ [−60,60] inmm*	
	x0,y0∈ [−60,60] in mm	
Movements range	*0° to 360°* degrees	
Population numbers	niindividuals	200
Crossover probability	Proportional type	varied
Mutation probability	Only one point	varied
Precision	Digits after point	6
Maximum number of generations	1000 generations	

Table 4. Six-bar mechanism restriction.

The path is obtained as a result of the evolution of the synthesis of the genetic mechanism (figure 10). In the subsequent figures and in table 5 it can be seen how decrease the error, while passing generations and changing some parameters to obtain the best adjustment.

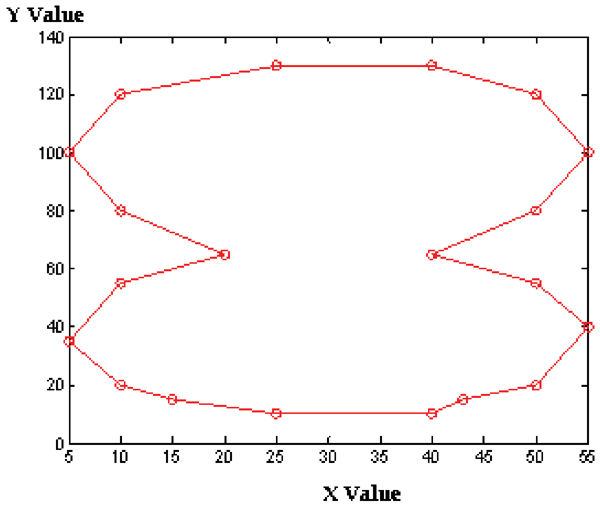


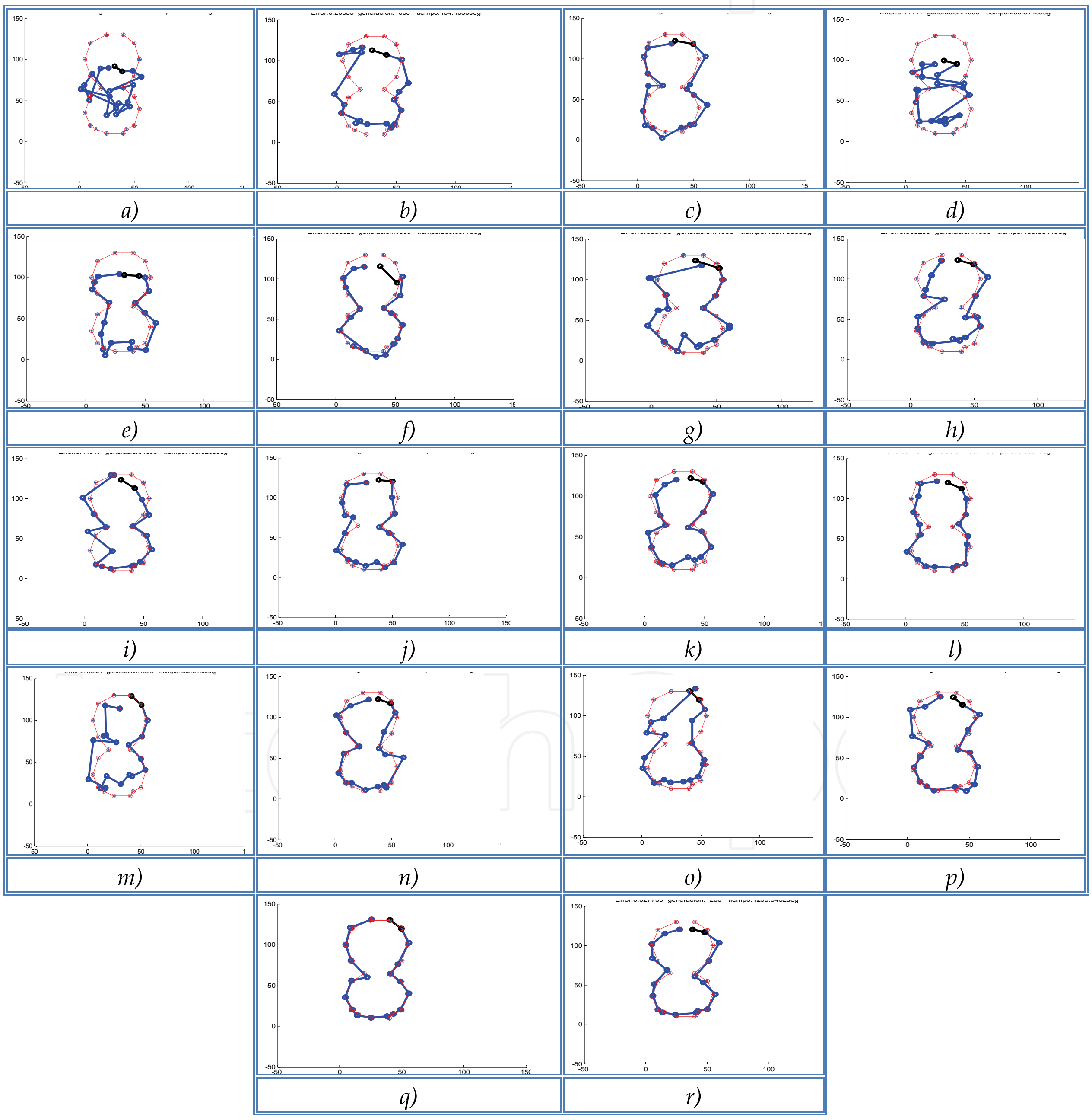
Fig. 9. Trajectory of 20 points for a six-bar Watt-I type mechanism.

This path was proposed with the objective of demonstrating that a six-link mechanism can follow paths that would be difficult to follow by a four-bar mechanism.

From this analysis it is concluded that:

- An increase in precision points is directly proportional to the number of individuals in the population, since to obtain a minor error, it is necessary to have a greater field of search.

- A small number of individuals decreases the search and does not offer satisfactory results.
- In order to obtain the optimal values is necessary to increase the value of the probability of crossing at least greater than 0.6.
- The rate of mutation can vary up to a maximum of 0.9, because if it increases to 1, it would be completely changing the individual without having a real meaning of the best for the evaluation, which was obtained with the elitism and the forced inheritance mechanism.
- High values of probability of crossover and mutation do not ensure that the best value of convergence is achieved.



a) b) c) d)

e) f) g) h)

i) j) k) l)

m) n) o) p)

q) r)

Fig. 10. Adjustment of parameters for a specific path.

	ni	*Pc*	*Pm*	*error*	*time*	*generations*
a)	200	0.6	0.01	0.4530713342399	180.693674	978
b)	200	0.6	0.4	0.2088516234	162.480752	991
c)	200	0.8	0.8	0.1039548356200	168.681711	996
d)	500	0.6	0.01	0.1441113	250.446997	981
e)	500	0.6	0.4	0.07266868	282.191	960
f)	500	0.8	0.8	0.0558228894	290.697	987
g)	1000	0.6	0.01	0.059796068	468.693084	999
h)	1000	0.6	0.4	0.0532988776	480.819397	947
i)	1000	0.7	0.5	0.119467646451	457.490696	999
j)	1000	0.7	0.7	0.03260650948	524.067239	989
k)	1000	0.8	0.5	0.099396876739	536.369984	993
l)	1000	0.85	0.8	0.0311870033413	550.612374	972
m)	1500	0.6	0.1	0.1962062488	619.52535	981
n)	1500	0.6	0.4	0.090105144	672.77304	990
o)	1500	0.95	0.85	0.163192355	1046.2808	968
p)	2000	0.7	0.7	0.08380448	1116.16188	999
q)	2000	0.85	0.8	0.0114246856933	1295.874818	987
r)	2000	0.95	0.85	0.0277589482798	1306.641231	1000

Table 5. Adjustment of six-bar mechanism parameters for a specific path.

5. Discussion

The optimization process is iterative, and it was demonstrated with the tests that were realized varying the parameters of the genetic algorithm to analyze the behavior of the system, which means that they can be modified until finding a system whose behavior satisfies the expectations and requirements of the designer. The parameters of the GA usually interact with each other in a nonlinear relation, that's why they cannot be optimized in an independent way, been demonstrated in the presented study cases. When existing a change in the population size, this fact will be reflected in time of convergence and accuracy in the path generation.

It was demonstrated that the diversity of individuals in the population is obtained and it remains along with the operator of crossing and the genetic mutation, since in all the analysis, they allow to find better solutions and avoid premature convergence to the maximum premises. Although also it must be mentioned that the elitism and the forced inheritance help to limit the number of individuals that will cover the imposed restrictions. On the other hand, it was observed that the GA has few possibilities of making considerable or necessary a number of reproductions for the optimal solution if it has an insufficient or small population.

Besides, the excessive population causes that the algorithm requires of a greater time of calculation to process and to obtain a new generation. In fact, there is not a limit wherein it is inefficient to increase the size of the population since it neither obtains a faster speed in the resolution of the problem, nor the convergence makes sure. For the referred study cases in this chapter, when increasing the population to 3500 individuals no acceptable results are presented and the program became extremely slow. If the population remains so large, like for example 1000 individuals, this means that it can improve the performance of the algorithm, although this is affected by slower initial responses. It is important to do emphasis on the relation that exists among the population size and the probabilistic relation in the solution space of the problem.

The study cases of this work are over determined and nonlinear type, which implies by necessity a space of multidimensional, nonlinear and non-homogenous solution, therefore, large initial values cover different regions of the solution space wherein the algorithm could converge prematurely to a solution that implies optimal premises costs, but when maintaining a low probability of mutation is not possible to assure that the population, although extensive in the number of individuals, continues being probabilistic representative of the problem solution. With this in mind and considering that the computation time to evaluate and to generate a new population of individuals from the present initial or, directly is the bound to the number of individuals of this one, requires a greater number of operations to obtain a new generation of possible solutions.

6. Conclusions

When operating with a population reduced in number of individuals, a sufficient representative quantity of the different regions of the solution space is not achieved, but the necessary computation time to create a new generation of possible solutions diminishes dramatically. When considering a high percentage of the probability of mutation in the algorithm, one assures a heuristic search made in different regions of the solution space, this combined with the forced inheritance mechanism has demonstrated that for the problem treated in this work, it is a strategy that power the heuristic capacities of the GA, for nonlinear multidimensional problems, non-homogenous, becoming the algorithm meta-heuristic; it is demonstrated then that an important improvement in the diminution of the error is obtained, around 20% with respect to the reported works previously. Also it was observed that the increase in the percentage of mutation improves the off-line performance, since all the solutions in the population are taken into account to obtain the optimal value. The off-line performance does not penalize the algorithm to explore poor regions of the search space, as long as it contributes to reach the best possible solutions in terms of aptitude.

It was verified that for the crossover the rule is fulfilled of which applying values smaller to 0.6, the performance is not optimal and it does not change the expected result for a specific problem. In the case of mutation, one demonstrated that this one can change no mattering the number of times and increasing its value to obtain optimal results, reaching almost at the unit, but avoiding to muter totally all the chromosomes eliminating the benefits created by the elitism and the forced inheritance mechanism.

By means of the trial and error, also one concludes that the parameters are not independent, and searching systematically to obtain all the possible combinations of these, is almost impossible; but if the parameters were optimized one at the time, it is then possible to handle its interactions and, for a given problem, the values of the selected parameters are not necessarily the optimal ones, but if they are analyzed uniformly they will generate more significant values.

7. Acknowledgment

The authors thank to Instituto Politécnico Nacional, Project Number. 20113426, for the facilities and means for the development of this research.

8. References

A. K. Mallik & A. Ghosh 1994. Kinematic Analysis and Synthesis of Mechanisms. *CRC-Press*, 688.

Beasley, D., Bull, D. R. & Martin, R., And R. 1993. An overview of genetic algorithms: part 1, fundamentals. *University Computing,* 15, 58-69.

Bethke, A. 1976. Comparison of Genetic Algorithms and Gradient-Based Optimizers on Parallel Processors: Efficiency of Use of Processing Capacity. *Tech. Rep. No. 197, Logic of Computers Group, University of Michigan*.

Bulatovic, R. R. & Djordjevic, S. R. 2004. Optimal Synthesis of a Four-Bar Linkage by Method of Controlled Deviation. *The first international conference on computational mechanics (CM'04),* 31,No.3-4, 265-280.

Cabrera, J. A., Simon, A. & Prado, M. 2002. Optimal synthesis of mechanisms with genetic algorithms. *Mechanism and machine theory (Mech. mach. theory),* 37 No10, 1165-1177.

Cantu-Paz, E. 2000. Efficient and Accurate Parallel Genetic Algorithms. *Kluwer, Boston, MA*.

Coello-Coello, C. A. 2007. Introducción a la computación evolutiva. *In:* CINVESTAV-IPN (ed.). Mèxico.

De Jong, K. A. 1975. *An analysis of the behaviour of a class of genetic adaptive systems.* Tesis doctoral, University of Michigan.

Denavit, J. & Hartenberg, R. S. 1964. Kinematic Synthesis of Linkages. *USA: Mc. Graw Hill.*

Dewen, J., Ruihong, Z., Ho, D., Rencheng, W. & Jichuan, Z. 2003. Kinematic and dynamic performance of prosthetic knee joint using six-bar mechanism *Journal of Rehabilitation Research and Development,* 40,No. 1, 39–48.

Endre Eiben, A., Hinterding, R. & Michalewicz, Z. 1999. Parameter Control in Evolutionary Algorithms. *IEEE Transactions on Evolutionary Computation,* 3, No. 2.

Fogarty, T. 1989. Varying the probability of mutation in the genetic algorithm. *Proc. 3rd Int. Conf. Genetic Algorithms, J. D. Schaffer, Ed. San Mateo, CA: Morgan Kaufmann*.

Freudenstein, F. 1954. An analitical approach to the design of four link mechanism. *Transactions of the ASME 76*, 483-492.

Goldberg, D. 2002. Lessons from and for Competent Genetic Algorithms. *Kluwer, Boston, MA*.

Goldberg, D. E. 1989. *Genetic algorithms in search, optimization, and machine learning,* USA, Addison - Wesley.

Grefenstette, J. J. 1986. Optimization of control parameters for genetic algorithms. *IEEE Trans. Systems, Man, Cybern,* 16, no. 1, 122-128.

Grosso, P. 1985. Computer Simulations of Genetic Adaption: Parallel Subcompnent Interaction in a Multilocus Model. *Ph.D Dissertation, University of Michigan*.

Hidalgo, J. I. & Lanchares, H. R. 2000. Partitioning and placement for multi-fpga systems using genetic algorithms. *In Proceedings of the Euromicro DSD 2000*.

Holland, J. H. 1975. Adaptation in natural and artificial system. *Ann Arbor, The University of Michigan Press*.

Kalnas, R. & Kota, S. 2001. Incorporating Uncertaintly intoMechanism Synthesis. *Mechanism and machine theory (Mech. mach. theory),* 36, No.3**,** 843-851.

Kunjur, A. & Krishnamurty, S. 1997. Genetic Algorithms in Mechanism Synthesis. *Journal of Applied Mechanisms and Robotics,* 4 No. 2**,** 18-24.

Kuri-Morales, A. & Galaviz-Casas, J. 2002. Algoritmos Genéticos. *Instituto Politécnico Nacional, Universidad NAcional Autonoma de México, Fondo de Cultura Económica,***,** 202.

Laribi, M. A., Mlika, A., Romdhane, L. & Zeghloul, S. 2004. A combined genetic algorithm-fuzzy logic method (GA-FL) in mechanisms synthesis. *Mechanism and machine theory (Mech. mach. theory),* 39**,** 717-735.

Levitski, N. L. & Shakvazian, K. K. 1960. Synthesis of four element spatial mechanisms with lower pairs. *International Journal of Mechanical Sciences* 2**,** 76-92.

Lima, C. A. F. 2005. Combining Competent Crossover and Mutation Operators:a Probabilistic Model Building Approach. *GECCO'05*.

Lima, C. F., Sastry, K., Goldberg, D. E. & Lobo, F. G. 2005. Combining Competent Crossover and Mutation Operators: a Probabilistic Model Building Approach. *GECCO'05,* ACM 1595930108/ 05/0006.

Lugo-González, E., Hernández-Gómez, L. H., Ponce-Reynoso, R., Velázquez-Sánchez, A. T., Urriolagoitia-Sosa, G., Merchán-Cruz, E. A. & Ramírez-Gordillo, J. 2010. Performance Optimization of GA Based Planar Mechanism Synthesis. *In Proceedings of the 2010 IEEE Electronics, Robotics and Automotive Mechanics Conference (September 28 - October 01, 2010). IEEE Computer Society, Washington, DC***,** 126-131.

Lugo González, E. 2010. *Diseño de mecanismos utilizando algoritmos genéticos con aplicaciòn en prótesis para miembro inferior.*Doctorado, Instituto Politécnico Nacional.

Merchán-Cruz, E. A. 2005. *Soft-computing techniques in the trajectory planning of robot manipulators sharing a common workspace.* Doctor of Philosophy, Sheffield.

Merchán-Cruz, E. A., Hernández-Gómez, L. H., Velázquez-Sánchez, A. T., Lugo-González, E. & Urriolagoitia-Sosa, G. 2007. Exploiting monotony on a genetic algorithm based trajectory planner (GABTP) for robot manipulators.). F. *In the 16th IASTED international Conference on Applied Simulation and Modelling (Palma de Mallorca, Spain, August 29 - 31, 2007,* De Felice, Ed. ACTA Press, Anaheim, CA**,** 300-305.

Merchán-Cruz, E. A., Urriolagoitia-Sosa, G., Ramírez-Gordillo, J., Rodríguez-Caňizo, R., Campos-Padilla, I. Y., Muňoz-César, J. J. & Lugo-González, E. 2008. GA Based Trajectory Planner for Robot Manipulators Sharing a Common Workspace with Adaptive *Population Size. In Proceedings of the 2008 Electronics, Robotics and Automotive Mechanics Conference (September 30 - October 03, 2008). IEEE Computer Society, Washington, DC***,** 520-525.

Michalewicz, Z. 1999. Genetic Algorithms + Data Structure = Evolution Programs. *tercera ed. Nueva York: Springer*.

Norton, R. L. 1995. *Diseño de Maquinaria,* Impreso en México, Mc. Graw Hill.

Quintero-R, H., Calle-Trujillo, G. & Dáz-Arias, A. 2004. Síntesis de generación de trayectorias y de movimiento para múltiples posiciones en mecanismos, utilizando algoritmos genéticos. *Scientia et Technica,* 10 No.25.

Radcliffe, C. 1977. The knud Jansen lecture:above-knee mechanisms: kinematics, alignment and prescription criteria.: Prosthetic and orthetic practice.

Ramírez-Gordillo, J. 2010. *Planeación de trayectorias en sistemas de manipuladores robóticos múltiples.*Doctorado, Instituto Politécnico Nacional.

Rechenberg, I. 1973. Evolutionsstrategie Optimierung technischer systeme nach prinzipien der biologischen evolution. *Friedrich Frommann Verlag, Stuttgart-Bad Cannstatt*.

Roston, G. P. & Sturges, R. H. 1996. Genetic Algorithm Synthesis of Four-bar Mechanisms. *Artificial Intelligence for Engineering Design,Analysis and Manufacturing,* 10, pp. 371-390.

Sanchéz-Marín, F. T. 2000. *Contribución a la Síntesis Dimensional de Mecanismos Planos para Generación de Trayectorias.*Doctor en Ciencias, Universitat Jaume-I.

Sinha, A. 2002. Designing Efficent Genetic and Evolutionary Algorithm Hybrids. *M.S. Thesis, University of Illinois-Urbana-Champaign*.

Smith, B. D. A. S. S., R. 1995. Fitness Inheritance in Genetic Algorithms. *in: Proc. of the ACM Symposium on Applied Computing***,** 345-350.

Smith, R. 1993. Adaptively resizing populations: An algorithm and analysis. *Proc. 5th Int. Conf. Genetic Algorithms, S. Forrest Ed. San Mateo, CA: Morgan Kaufmann***,** 653.

Srivastava, R. 2002. Time Continuation in Genetic Algorithms. *M.S. Thesis, University of Illinois-Urbana-Champaign*.

Starosta, R. 2006. On some application of genetic algorithm in mechanism synthesis. *Annual Meeting of GAMM, Book of Abstracts,* 77.

Tzong-Mou, W. & Cha'o-Kuang, C. 2005. Mathematical model and its simulation of exactly mechanism synthesis with adjustable link. *Applied Mathematics and Computation,* 160**,** 309-316.

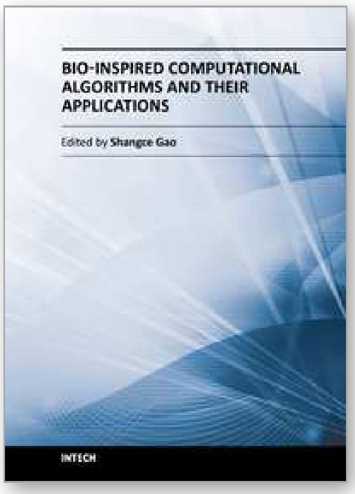
Varbanov, H., Yankova, T., Kulev, K. & Lilov, S. 2006. S&AExpert system for planar mechanism design. *Expert Systems with Applications,* 31**,** 558-569.

Vasiliu, A. & Yannou, B. 2001. Dimensional Synthesis of planar Mechanism Using Neural Network: Application to path generator lnkages. *Mechanism and machine theory (Mech. mach. theory,* 36 No. 2**,** 229-310.

Walczak, T. 2006. Mechanism synthesis with the use of neural network. *in Annual Meeting of GAMM, Book of Abstracts.Berlin,* 77.

Wen-Jyi, H., Chien-Min, O. & Chin-Ming, Y. 2003. Robust Vector Quantization for Burst Error Channels Using Genetic Algorithm. *European Symposium on Artificial Neural Networks***,** 267-274.

Whitley, D., Rana, S. & Heckendorn, R. 1998. Representation Issues in Neighborhood Search and Evolutionary Algorithms. *In D. Quagliarella, J. P´eriaux, C. Poloni, and G. Winter, editors, Genetic Algorithms and Evolution Strategies in Engineering and Computer Science. Recent Advances and Industrial Applications, chapter 3***,** 39-57.


BIO-INSPIRED COMPUTATIONAL ALGORITHMS AND THEIR APPLICATIONS
Edited by Shangce Gao
INTECH

Bio-Inspired Computational Algorithms and Their Applications
Edited by Dr. Shangce Gao

ISBN 978-953-51-0214-4
Hard cover, 420 pages
Publisher InTech
Published online 07, March, 2012
Published in print edition March, 2012

Bio-inspired computational algorithms are always hot research topics in artificial intelligence communities. Biology is a bewildering source of inspiration for the design of intelligent artifacts that are capable of efficient and autonomous operation in unknown and changing environments. It is difficult to resist the fascination of creating artifacts that display elements of lifelike intelligence, thus needing techniques for control, optimization, prediction, security, design, and so on. Bio-Inspired Computational Algorithms and Their Applications is a compendium that addresses this need. It integrates contrasting techniques of genetic algorithms, artificial immune systems, particle swarm optimization, and hybrid models to solve many real-world problems. The works presented in this book give insights into the creation of innovative improvements over algorithm performance, potential applications on various practical tasks, and combination of different techniques. The book provides a reference to researchers, practitioners, and students in both artificial intelligence and engineering communities, forming a foundation for the development of the field.

How to reference
In order to correctly reference this scholarly work, feel free to copy and paste the following:

Esther Lugo-González, Emmanuel A. Merchán-Cruz, Luis H. Hernández-Gómez, Rodolfo Ponce-Reynoso, Christopher R. Torres-San Miguel and Javier Ramírez-Gordillo (2012). Performance of Simple Genetic Algorithm Inserting Forced Inheritance Mechanism and Parameters Relaxation, Bio-Inspired Computational Algorithms and Their Applications, Dr. Shangce Gao (Ed.), ISBN: 978-953-51-0214-4, InTech, Available from: http://www.intechopen.com/books/bio-inspired-computational-algorithms-and-their-applications/performance-of-simple-genetic-algorithm-inserting-forced-inheritance-mechanism-and-parameters-relaxa

INTECH
open science | open minds

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

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