# Decision Strategies in Evolutionary Optimization

Arita Takahashi[1], Arkady Borisov[2]

[1] Institute of Information Technology, Technical University of Riga,
1, Kalkyu St., Riga LV-1658, Latvia
arita.takahashi@delfi.lv
[2] Institute of Information Technology, Technical University of Riga,
1, Kalkyu St., Riga LV-1658, Latvia
aborisov@egle.cs.rtu.lv

**Abstract.** This paper describes a sequel of the previous experiments on real number genetic algorithm behavior [2], [3]. A particular example of multi-criteria optimization is discussed. The behavior of the two previously explored genetic algorithms is compared with a simple evolutionary algorithm. The main idea of the experiments is to stimulate the algorithm to find the Pareto set without measuring dominance and non-dominance. The implication of maxi-min decision method is affecting optimization so that the final solutions lay closer to the Pareto set than those obtained without any decision method. This theoretical concept is tested and analyzed graphically by picturing populations after a certain number of generations. The differences in the algorithm behavior and causes of such differences are explained.

## 1 Introduction

As a rule, multi criteria optimization tasks are extremely complicated. The main reason for this is that usually no solution can be seen as supreme to others. But finding a region of good solutions raises another problem: what region exactly contains the best solutions? One may think the Pareto set is a very clear representation of good solutions for an algorithm to find. Although the indication of the Pareto set is very strict from the mathematical point of view, the physical borders of the Pareto region are gradual, not abrupt. This study of genetic algorithm behavior was started not to find reward mechanisms for non-dominated and punishments for dominated solutions. The research was made to clarify what part of the solution space the genetic algorithm will accept as preferable by itself.

### 1.1 Problem Formulation

Let us assume that two criteria have to be maximized simultaneously as shown in Fig.1, case a. An assumption is made in this study that both criteria have the same scale of measurement. They depend on two parameters, $x_1$ and $x_2$ (see the mathematical expressions (3), (4) and (5) at the end of this paper).

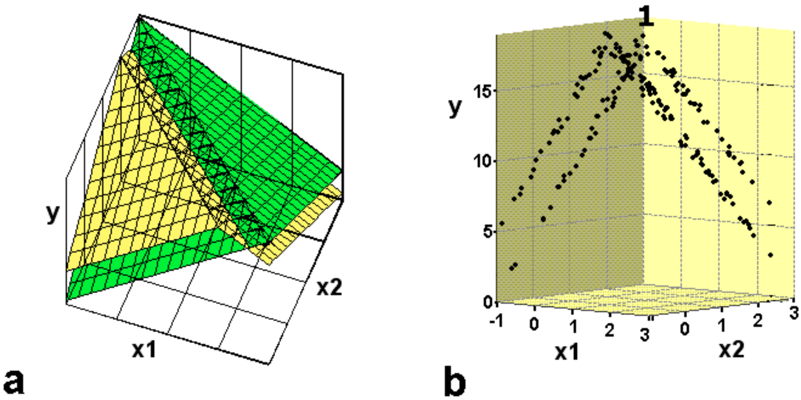



**Fig. 1.** The two criteria: (a) mathematical functions, (b) initial population cloud

Value Y is one of the criterion values. It will be identical to the first criterion's functional value $f_1(x_1,x_2)$ when a random number generated specifically for Y definition is $n_{rand}<0.5$, and it will be identical to the second criterion's functional value $f_2(x_1,x_2)$ when the random number is $n_{rand}\geq0.5$. Fig. 1 (see case b) shows randomly generated initial population for maximization of the criteria displayed in Fig.1, case a.

Each individual of the population has only one fitness value Y. The problem is that the same individual will get a different fitness value each time when the individual's fitness switches from one criterion to another. Using of maxi-min strategy was the idea for the solution of the problem. The hypothesis should be approved or disapproved by experiments.

### 1.2 Plan of Experiments

To analyze the effect of the decision strategy more correctly, three different approaches were compared:

1. Maximization with no decision strategy used;
2. Maximization with maxi-min strategy;
3. Maximization with maxi-max strategy.

Three different algorithms were tested:

1. The classic genetic algorithm based on the crossover of real numbers, mutation, random selection and random coupling, described in [2];

2. The restricted genetic algorithm built on the crossover of real numbers, mutation, random selection and purposeful coupling, the specification of which is given in [3];

3. The evolutionary algorithm based on real number mutation operator and random selection operator. We will call this algorithm evolutionary because it has no crossover operator in it.

## 2 Decision Methods

There are three main decision strategies one may use in multi-attribute solution space [1]: dominance/non-dominance strategy, maxi-min strategy and maxi-max strategy. In the experiments described in this paper only maxi-min and maxi-max methods were used. The maxi-min method or the strategy of pessimist builds its outcome only on the worst attribute values, whereas the maxi-max method or the strategy of an optimist considers only the best attribute values.

## 3 Evolutionary Strategies

Wright [4] describes the nature of a crossover operator in the case when the phenotypes of binary strings are real numbers of parameter values. It can be seen that if two binary strings are crossed over in the place that separates the two representations of real values, the so-called real crossover occurs. It means the strings of real parameters may be used instead of those of binary symbols. For example, if the crossover had been performed between the first and the second parameter, then the two parents would have obtained the following children due to the crossover:

$$\begin{matrix} Parent\ 1 = (x_1, x_2, \dots x_n) \\ Parent\ 2 = (x'_1, x'_2, \dots x'_n) \end{matrix} \Rightarrow \begin{matrix} Child\ 1 = (x_1, x'_2, \dots x'_n) \\ Child\ 2 = (x'_1, x_2, \dots x_n) \end{matrix} \quad \textbf{(1)}$$

### 3.1 Strategies of Real Crossover

For better understanding, let us clarify the case of real crossover in two-parameter space. For example, if parameters of one parent are (2.5, 0.15) and parameters of the other parent are (3.7, 0.4) then these two children may be obtained:

$$\begin{matrix} Parent1 & (2.5, & \vert 0.15) \\ Parent2 & (3.7, & \vert 0.4) \end{matrix} \Rightarrow \begin{matrix} Child1 & (2.5, 0.4) \\ Child2 & (3.7, 0.15) \end{matrix} \quad \textbf{(2)}$$

In parameter space children are located on apices of parallelepiped. In the world of living beings, the offspring variety seems immeasurable. To make a model of the evolution of living beings the following definition of an offspring was introduced: For phenotypes $X=(x_1, x_2, x_3, \dots x_n)$ and $X'=(x'_1, x'_2, x'_3, \dots x'_n)$ a descendant is the phenotype whose parameters in the real numbers domain lay between the parameters of its parents. Therefore the first co-ordinate of the descendant is a random number lying between $x_1$ and $x'_1$, but the second co-ordinate is a random number between $x_2$ and $x'_2$ and so on. The potential identities of a child in a two-parameter case are uncountable as they occupy the whole space between the parameters' values of the parents (see Fig.2.). On the other hand both parents' coordinates determine limitations or boundaries for every parameter.

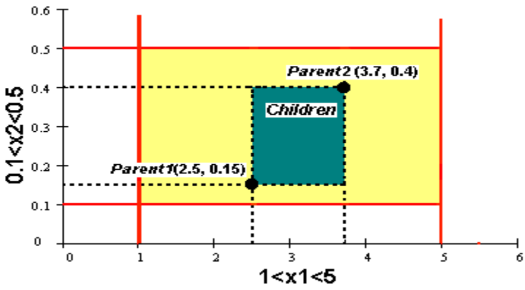



**Fig. 2.** The area of potential identities of a child

### 3.2 Mutation of Real Parameter

The mutation operator is executed by shifting the real number. The direction of the shift is random, there are no rules like “if parameter $x_1$<value$_1$ then its shift is positive, otherwise – negative”. Mutation is uncontrollable, but random parameter values vary randomly. There are two aspects of mutation operator that have to be taken into account when creating a genetic algorithm based on genetic operators:

- By executing mutation operator a new step of search is done and a good solution may be obtained.
- Mutation operator may cause a loss of a very good solution.

In the case when the algorithm is based on the crossover operator depicted in Fig.2, there may be no need for mutation because the shift of parents’ parameters is included in the definition of the crossover operator. On the other hand, there is a lower possibility of loosing a good solution without obtaining a better one or at least the one as valuable as the lost. The crossover operator is not an uncontrollable shift as it is limited by parents’ parameter values. If all population individuals are crawling towards better solutions and if the space of better solutions occupy a settled place in the parameter space then the population shrink and the occupied area represent the surroundings of the best solution (see Fig.3).

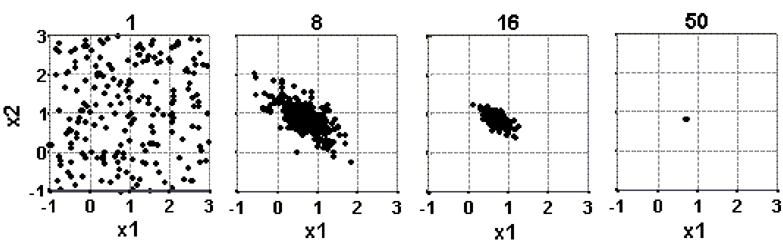



**Fig. 3.** Shrinking of the population towards the best solution (classic genetic algorithm, population size is 200, maxi-min strategy)

Crawling in the right direction is achieved by using operators of selection and crossover. In the first approximation the surroundings of the best solution cover all the parameter space. In the next approximations it includes only the area occupied by the population. To represent this process more visually, the 1st, 8th, 16th and 50th generations of the population in two-parameter space are shown in Fig.3.

A special case may occur when both parents have identical parameters. This leads to an impasse because all the children produced by them are copies of the same solution. As a result of this, two negative aspects may appear:

- The stage of search for better solution is impossible.
- Assimilation process can take place.

Hence the following corrections can be recommended to improve the convergence of the algorithm:

1. Obtain a child by random initialization of a new individual independently from parents;
2. Shift by mutation one parameter of one parent.

Afterwards both parents and children participate in the selection of the next generation.

In an evolutionary algorithm, which does not use crossover, the only operator for producing a child is mutation. In two-parameter space one parameter is altered randomly while the other remains unchanged. Afterwards both parents and children have to compete for selection.

### 3.3 Purposeful Coupling

The analysis of marriage models in the human environment shows a variety of restrictions. For instance, in geographical terms, there will be a biological restriction imposed by the human race: there are individuals who are unable to get accustomed to very cold weather conditions because their body is adapted to life in a hot climate, and vice versa. In their turn, restrictions for mutual contacts and understanding may be imposed by the mother tongue or nationality. The principal idea here is that individuals may be crossed over only within the limits of one subpopulation, called a nation, whereas the selection is performed on the whole population.

The following model of this marriage in the restricted genetic algorithm is employed:

The example shown in Fig.1 relates to the two-parameter $x_1$, $x_2$ space (see Fig.4). Parameter $x_1$ varies from $x_{1min}$ to $x_{1max}$, while parameter $x_2$ varies from $x_{2min}$ to $x_{2max}$. First, phenotypes are ordered in the ascending sequence of the parameter $x_1$ values; the first n/5 phenotypes which are the nearest to $x_{1min}$ will be assigned to the first race, and the next n/5 ones will be ascribed to the second race and so on. Then, the phenotypes are ordered in the ascending sequence of the parameter $x_2$ values; the first m/3 phenotypes of the first race will be assigned to the first nation, and the next m/3 ones will be ascribed to the second nation and so on. Thus, each phenotype will be assigned the “race” and “nation” ordinal numbers. Essentially, this algorithm may be repeated indefinitely.

The third restriction for marriage is represented by the so-called estate meaning that a person belonging to a specific estate marries a person of the same estate. It is in a sense a model of the Middle Ages relationships where upper class person has to marry equal position person by the conventional rules of the society. Although the poor have to marry the poor it seems that reproduction does not depend on an individual’s welfare.

To model this process all phenotypes of a nation are arranged in the descending sequence of fitness values Y. After that individuals are coupled - the first best one with the second best one, the third with the fourth, and so on. The two parents have k (for example, 2) children. After that, the parents together with the children take part in the competitive activity for survival.

The classic genetic algorithm is a model of casual contacts between parents. One individual with high fitness can be a parent to all the children of the population if it is chosen for crossover as the first or the second parent. At the same time the other individual with low fitness is not participating in parenthood at all. Relationships “one individual with many others” may cause assimilation of population by one particular kind of the individuals (see [2]).

In contrast to the classic genetic algorithm, the restricted genetic algorithm is a model of devoted relationship between parents. The restrictions of coupling of parents lead to a more local search process (see [2] and [3]). This process can be seen in Fig.4.

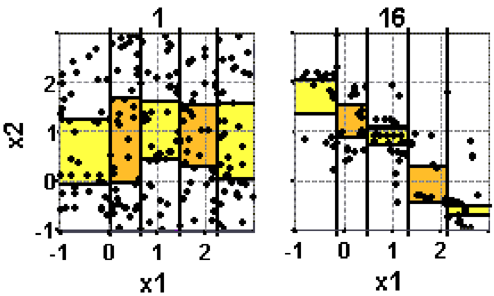



**Fig. 4.** Population of 15 nations (restricted genetic algorithm, population size is 200, maxi-min strategy): 1st generation and 16th generation

While searching for good solutions inside each nation, the solutions that belong to the Pareto set are found. Comparing with 16th generation, depicted in Fig.3, the population of the restricted genetic algorithm concentrates around many different solutions, not around one single point.

### 3.4 Selection Operator

After the appearance of children, the size of the population is doubled. Because the crossover rate is 1.0, every couple after crossover in the genetic algorithms has two children. In the evolutionary algorithm each individual has a child obtained by mutation, therefore the population has double size as well. If the population size is 200 then the doubled population size is 400. By performing selection operator 200 times next generation is produced. In the experiments performed the roulette method of selection was used. To increase the frequency of choosing the best solutions, the relative fitness value $Y-Y_0$ was calculated and applied instead of the fitness value Y ($Y_0$ is the population's minimal value of Y, or "relative zero").

# 4 Description of algorithms

There is a short description of all the algorithms.

As the experiments were not aimed to observe an influence of algorithms' parameter changes, the same parameters were used in all the experiments. Population size in all the experiments was 200. In every experiment 100 generations (iterations) were performed. The rate of crossover was 1,0 but the number of children obtained in one crossover operator was 2. In the evolutionary algorithm the rate of mutation was 1,0 (every individual was mutated) and the number of children obtained in one mutation operator was 1. In the restricted algorithm there were 5 races and 3 nations in every race (altogether 15 sub-populations).

## 4.1 The classic genetic algorithm

The "classic genetic" algorithm is executed as follows:
1. Create the initial population by random generation of 200 individuals;
2. Calculate the fitness value Y and $Y-Y_0$ for each individual;
3. In loop 100 times: two times perform selection for choosing both parents, then cross over (operator, displayed in Fig.2) the parents and obtain 2 children (but if the parents coincide then initialize two new children and mutate the first parent by altering one parameter);
4. Calculate the fitness value Y and $Y-Y_0$ for each individual;
5. Collect the data for results;
6. Check the loop counter value: if it is 100 then finish, otherwise go back to step 3.

## 4.2 The restricted genetic algorithm

The restricted genetic algorithm has the following steps:
1. Create an initial population by a random generation of 200 individuals;
2. Calculate the fitness value Y and $Y-Y_0$ for each individual;
3. Collect data for the results;
4. Assign the number of race:
    4.1. For the first 40 individuals having low values of parameter $x_1$ the race number is 1;

4.2. For the next 40 individuals having larger values of parameter $x_1$ the race number is 2;
4.3. For the next 40 individuals having larger (than in races 1 and 2) values of parameter $x_1$ the race number is 3;
4.4. For the next 40 individuals having larger (than in races 1, 2 and 3) values of parameter $x_1$ the race number is 4;
4.5. For the next 40 individuals having larger (than in races 1, 2, 3 and 4) values of parameter $x_1$ the race number is 5.

5. Divide each race into nations by assigning a nation number:
   5.1. For the first 13 individuals having low values of parameter $x_2$ the nation number is 1;
   5.2. For the next 13 individuals having larger values of parameter $x_2$ the nation number is 2;
   5.3. For the next 14 individuals having larger (than in nations 1 and 2) values of parameter $x_2$ the nation number is 3.
6. Arrange the individuals belonging to the same race and nation by Y value in the descending sequence. Assign couple numbers as follows: for the first two individuals number 1, for the second two number 2 and so on;
7. Cross over (operator, displayed in Fig.2) the parents (The individuals within the same race and the same nation having the same couple numbers) and obtain 2 children (but if the parents coincide then initialize two new children and mutate the first parent by altering one parameter);
8. Calculate the fitness value Y and $Y\text{-}Y_0$ for each individual;
9. Perform the selection operator 200 times;
10. Check the loop counter value: if it is 100 then finish, otherwise go back to step 3.

### 4.3 The evolutionary algorithm

This algorithm is very simple and includes the following steps:
1. Create the Initial population by random generation of 200 individuals;
2. Calculate the fitness value Y and $Y\text{-}Y_0$ for each individual;
3. Collect data for the results;
4. Mutate each individual by altering one of the parameter values;
5. Calculate the fitness value Y and $Y\text{-}Y_0$ for each individual;
6. Perform the selection operator 200 times;
7. Collect data for the results;
8. Check the loop counter value: if it is 100 then finish, otherwise go back to step 4.

## 5 Experiments

As the maximization criteria, two mathematical functions were used (see Fig.1, case a). They can be expressed as follows:

$$Y = 18.5 + \frac{1}{2} \cdot F(x_1, x_2) + [-1 + 4 \cdot \mathrm{sgn}(1 - (1 + F(x_1, x_2)))] \cdot F(x_1, x_2) \quad \textbf{(3)}$$

*where*

$$F(x_1, x_2) = 1 - (x_1 + x_2) \quad or \quad F(x_1, x_2) = 1 - (x_1 + x_2 - 1) \quad \textbf{(4)}$$

$$and \quad \mathrm{sgn}(x) = \begin{cases} +1 & \forall x > 0 \\ 0 & x = 0 \\ -1 & \forall x < 0 \end{cases} \quad \textbf{(5)}$$

The parameters intervals were the following:

$$-1 \le x_1 \le 3 \quad \textbf{(6)}$$

$$-1 \le x_2 \le 3 \quad \textbf{(7)}$$

As can be seen from the above expression the maximal value for both criteria is 18.5, but the fitness function's maximal value is 18.5 only under the maxi max strategy. However, as the difference $Y\text{-}Y_0$ is used for selection, the absolute function values do not play any role at all.

## 6 The Results

The most important traits of the experiments are shown in the appendix.

The results obtained were stored in the database and analyzed graphically by using SPSS graphical tools. The 1st, 8th, 16th, 24th, 50th and the 100th generations were saved. In patterns A, B, C one can see a visualization of the classic genetic algorithm, but D, E, F show the restricted algorithm behavior and G, H, L display the population of evolutionary algorithm. It can be seen that the root of the most marked analogy of behavior is the one built in the algorithm. Different ways of crossover operator, modifications in the coupling affect the solution space found by genetic or evolutionary algorithm. Therefore the classic genetic algorithm every time prefers only one point, usually in the middle of good solution area. The restricted genetic algorithm has the highest skill to cover all the area of good solutions although it has asymmetries in each subpopulation as well. The evolutionary algorithm has the largest variety of solutions although the strong migration caused by too high rate of mutation does not allow the algorithm to settle some part of population unchangeably.

The dark area is the target area. Although the size of this area is a disputed question (because the sensitivity of various algorithms is different and nobody can tell which precision is acceptable and which is not). The density of solutions inside the target areas is larger than the density outside of them. As the evolutionary algorithm is not sensitive enough it is difficult to determine if its concentration of solutions in target areas is larger.

The behavior of the classic genetic algorithm is illustrated in Fig. 5. There is depicted the 8th generation of the classic genetic algorithm, using maxi-min strategy. One can see that all individuals of the population are located inside the rectangle and the children individuals obtained from such population will also be located inside this rectangle. Therefore the algorithm is unable to create a child belonging to good solution area A or B.

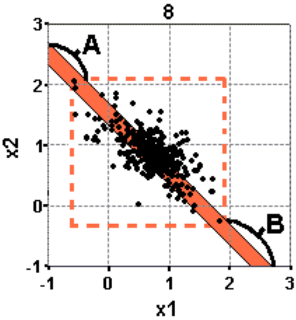



**Fig. 5.** The 8th generation of the classic genetic algorithm (maxi-min strategy)

As compared to the classic genetic algorithm, the restricted one has a different search tactic, because it arranges all individuals by their fitness values and because each individual can get only one partner for crossover and can not be selected for other individuals repeatedly.

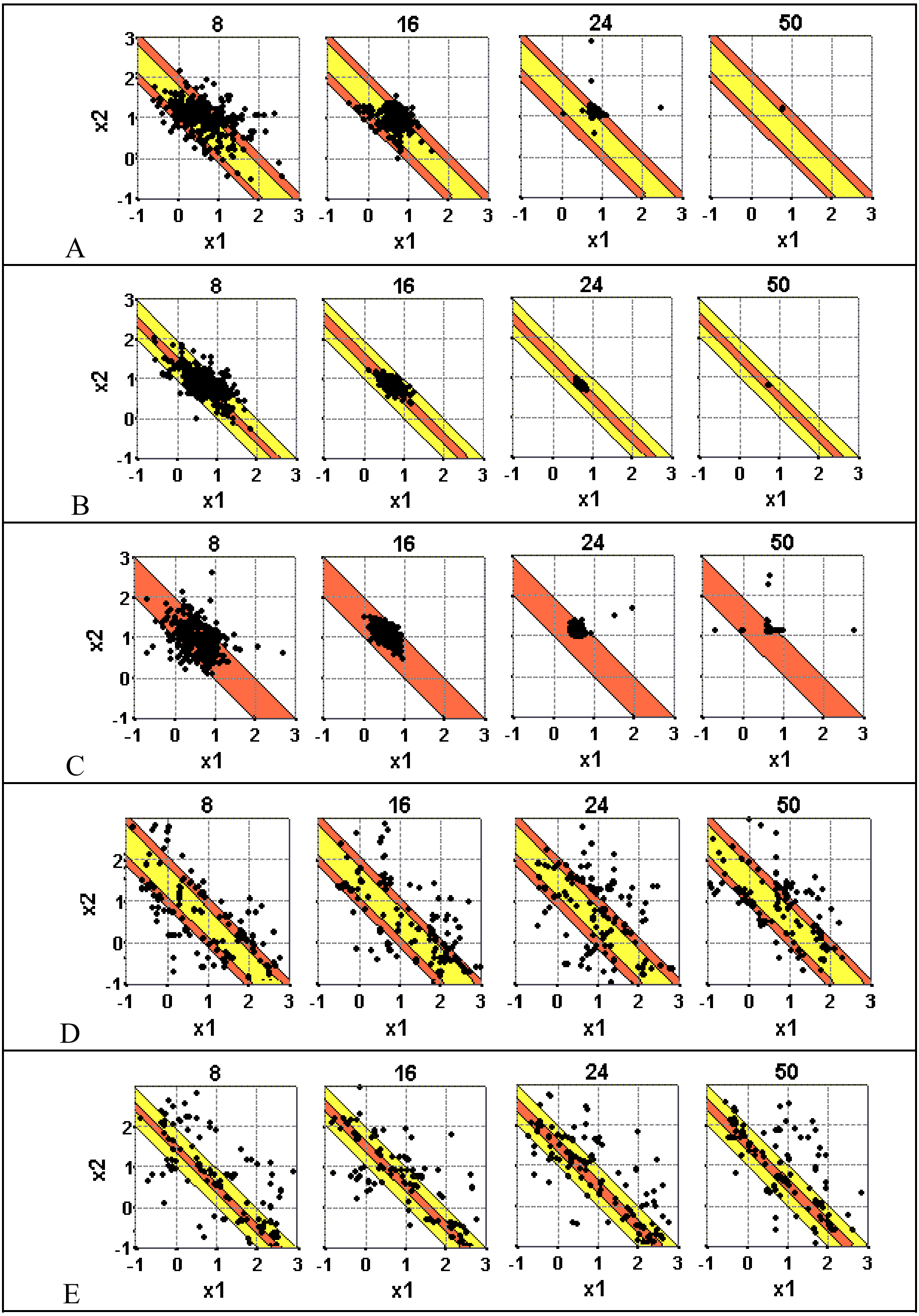
## 7 Conclusions

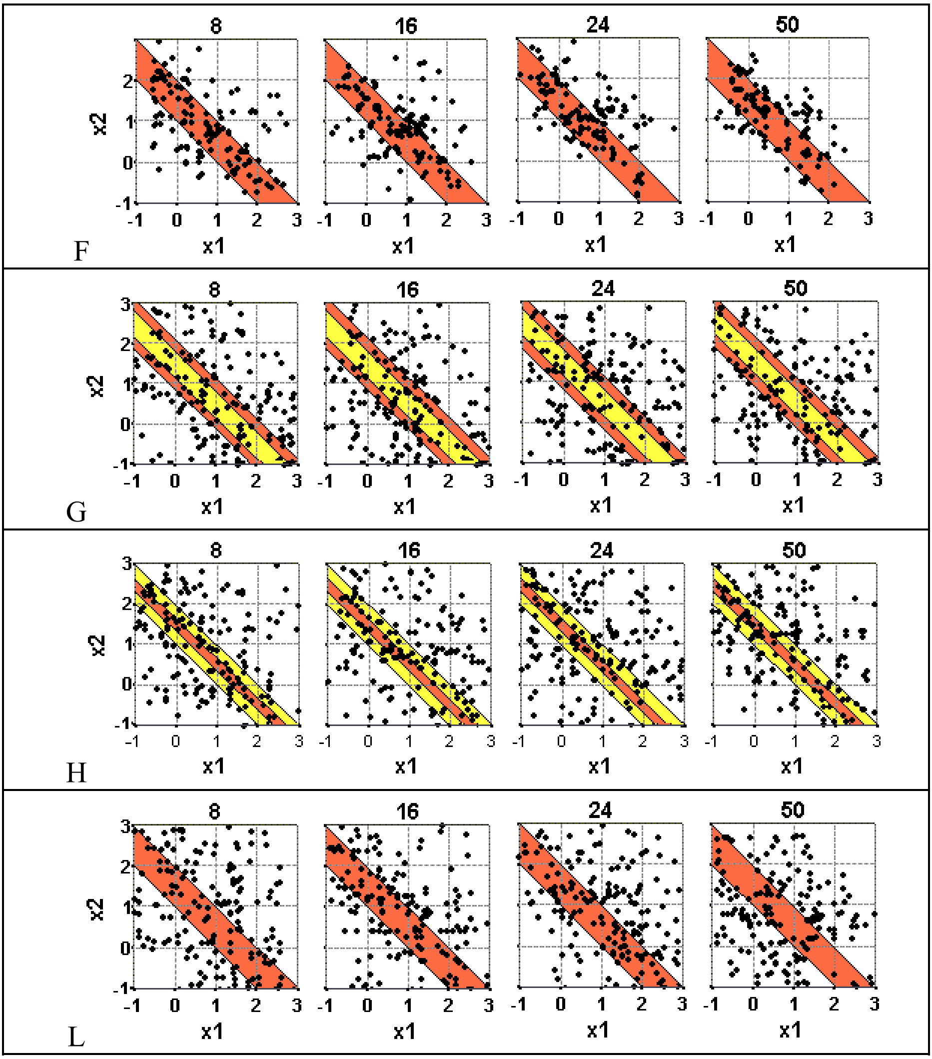
The experiments showed the ability of classic and restricted genetic algorithm to response to different strategies by producing different final sets. The sensibility of the evolutionary algorithm was not strong enough. The use of maxi-min method increased the number of points belonging to the outcome of pessimist strategy. But as the number of points located outside the Pareto set depends to a high degree on the algorithm behavior, it is not possible to decrease it by trying the maxi-min decision strategy. Further research will be dealt with performing experiments aimed to measure the dominance and reward it by high fitness.

## References

1. Hwang, C.L., Yoon, K.: Multiple attribute decision-making methods and application: A state of the art survey. Berlin, Springer Verl. (1981) 58–66
2. Takahashi, A.: Using Genetic Algorithms to Search for Maximum of the Multimodal Function. In: Proceedings of the 6th International Conference on Soft Computing MENDEL 2000, Brno University of Technology, Brno, Czech Republic (2000) 159–164
3. Takahashi A.: Genetic Algorithm Behaviour in the Multi-criteria Optimisation Task. In Proceedings of the Fourth International Conference on Knowledge-Based Intelligent Engineering Systems & Allied Technologies KES'2000, University of Brighton, Sussex, U.K. (2000) 514–520
4. Wright A. H.: Genetic Algorithms for Real Parameter Optimization. In: Foundations of GA. Gregory J.E. Rowlins, ed., Morgan Kaufmann Publishers (1991) 205–218

## Appendix: Results - The 8$^{th}$ , 16$^{th}$ , 24$^{th}$ , and 50$^{th}$ generations





8
16
24
50
x1
x2
-1
0
1
2
3
F
G
H
L