Int. J. Emerg. Sci., 3(4), 333-344, December 2013
ISSN: 2222-4254

Selection Methods for Genetic Algorithms

Khalid Jebari, Mohammed Madiafi

Laboratory, Faculty of Sciences, Mohammed V–Agdal University, UM5A, Rabat, Morocco,
Ben M'sik Faculty of Sciences, Hassan II Mohammedia University, UH2M, Casablanca, Morocco,

khalid.jebari@gmail.com, madiafi.med@gmail.com

Abstract. Based on a study of six well known selection methods often used in genetic algorithms, this paper presents a technique that benefits their advantages in terms of the quality of solutions and the genetic diversity. The numerical results show the extent to which the quality of solution depends on the choice of the selection method. The proposed technique, that can help reduce this dependence, is presented and its efficiency is numerically illustrated.

Keywords: Evolutionary Computation, Genetic Algorithms, Genetic Operator, Selection Pressure, Genetic Diversity.

1 INTRODUCTION

The theory of genetic algorithms (GAs) was originally developed by John Holland in 1960 and was fully developed in his book " Adaptation in Natural and Artificial Systems ", published in 1975 [1]. His initial goal was not to develop an optimization algorithm , but rather to model the adaptation process, and show how this process could be useful in a computing system. The GAs are stochastic search methods using the concepts of Mendelian genetics and Darwinian evolution [2,3]. Such heuristics have been proved effective in solving a variety of hard real-world problems in many application domains including economics, engineering, manufacturing, bioinformatics, medicine, computational science, etc [4].

In principle, a population of individuals selected from the search space , often in a random manner, serves as candidate solutions to optimize the problem [3]. The individuals in this population are evaluated through ("fitness") adaptation function. A selection mechanism is then used to select individuals to be used as parents to those of the next generation. These individuals will then be crossed and mutated to form the new offspring. The next generation is finally formed by an alternative mechanism between parents and their offspring [4]. This process is repeated until a certain satisfaction condition.

More formally, a standard GA can be described by the following pseudo-code:

```
SGA_pseudo-code
{
•  Choose an initial population of individuals: P(0);
•  Evaluate the fitness of all individuals of P(0);
•  Choose a maximum number of generations: t_max;
•  While (not satisfied and t < t_max) do {
       -  t ← t+1;
       -  Select parents for offspring production;
       -  Apply reproduction and mutation operators;
       -  Create a new population of survivors: P(t);
       -  Evaluate P(t);
   }
•  Return the best individual of P(t);
}
```

As we can see from the pseudo-code above, a GA is a parametrical algorithm whose application to a given problem requires setting parameters and making decisions about:

- the way parents are selected for offspring production;
- selected parents are recombined;
- population size is adjusted;
- individuals are mutated, or crossed;
- probability of crossover and probability of mutation that explore other areas of research space are chosen [5].

The ultimate goal is the right choice of these parameters [5]. Research has therefore looked to find techniques to dynamically adjust and improve the quality of the solution. We find in the literature a lot of attention to the genetic operators exploration, and much more to population size [5]. By cons less attention is paid to the selection operator. Without this operator, genetic algorithms are only simple random methods give different values each time [3]. Individuals who have a higher fitness value have a high probability of being selected for the next generation by this operator. Selection method focuses research in promising areas of the search space.

The balance between exploitation and exploration is essential for the behaviour of genetic algorithms. It can be adjusted by the selection pressure of the selection operator and the probability of crossover and mutation. In the case of selection, a strong selection pressure may cause the algorithm converges to a local optimum, while a low selection pressure may cause the AG to random results that differ from one run to another [6]. The selection operator is aimed at exploiting the best characteristics of good candidate solutions in order to improve these solutions throughout generations, which, in principle, should guide the GA to converge to an acceptable and satisfactory solution of the optimisation problem at hand [2]. Selection operator is the most important parameter that may influence the performance of a GA [7]. It is aimed at exploiting the best characteristics of good candidate solutions in order to improve these solutions throughout generations, which, in principle, should guide the GA to converge to an acceptable and satisfactory solution of the optimisation problem at hand [6].

In the literature there are several selection methods: Roulette Wheel Selection [8], Stochastic Universal Sampling, the tournament selection and the selection of Boltzmann and others [9].

However, despite decades of research, there are no general guidelines or theoretical support concerning the way of selecting a good selection method for each problem [10]. This can be a serious problem because, as we will see through numerical results, a non-suitable selection operator can lead to poor performance of the GA in terms of both rapidity and reliability. To illustrate this problem we will consider a set of six popular selection methods that we apply to the optimization problem of four benchmark functions. In the following sections, we first provide a brief description of each studied selection method. In section 3, we outline the presentation of a technique that we propose in order to help reducing the influence of selection operator on the global performance of a GA. Numerical results are presented and discussed in section 4. Section 5 is devoted to our concluding remarks and future works.

2 SELECTION METHODS FOR GAS

As mentioned before, six different selection methods are considered in this work, namely: the roulette wheel selection (RWS), the stochastic universal sampling (SUS), the linear rank selection (LRS), the exponential rank selection (ERS), the tournament selection (TOS), and the truncation selection (TRS). In this section, we provide a brief description of each studied selection method.

2.1 Roulette Wheel Selection (RWS)

The conspicuous characteristic of this selection method is the fact that it gives to each individual i of the current population a probability $p(i)$ of being selected [10], proportional to its fitness $f(i)$

$$p(i) = \frac{f(i)}{\sum_{j=1}^{n} f(j)} \tag{1}$$

Where n denotes the population size in terms of the number of individuals. The RWS can be implemented according to the following pseudo-code

```
RWS_ pseudo-code
  {
```
- Calculate the sum $S = \sum_{i=1}^{n} f(i)$;

```
• For each individual 1≤i≤n do {
    - Generate a random number α∈[0,S];
    - iSum = 0;    j = 0;
    - Do {
        - iSum ← iSum + f(j);
        - j ← j+1;
      } while(iSum < α  and  j < n)
  -    Select the individual j;}
  }
```

Note that a well-known drawback of this technique is the risk of premature convergence of the GA to a local optimum, due to the possible presence of a dominant individual that always wins the competition and is selected as a parent.

2.2 Stochastic Universal Sampling (SUS)

The SUS [8] is a variant of RWS aimed at reducing the risk of premature convergence. It can be implemented according to the following pseudo-code:

```
SUS_ pseudo-code
   {
• Calculate the mean f̄ = 1/n ∑_{i=1}^{n} f(i);
• Generate a random number α∈[0,1];
• Sum = f(1);   delta = α × f̄;   j = 0;
• Do {
    - If (delta < Sum) {
        - select the jth individual;
        - delta = delta + Sum;
      }
      else {
        - j = j+1;
        - Sum = Sum + f(j);
      }
  } while (j < n)
  }
```

2.3 Linear Rank Selection (LRS)

LRS [9] is also a variant of RWS that tries to overcome the drawback of premature convergence of the GA to a local optimum. It is based on the rank of individuals rather than on their fitness. The rank n is accorded to the best individual whilst the worst individual gets the rank 1. Thus, based on its rank, each individual *i* has the probability of being selected given by the expression

$$p(i) = \frac{rank(i)}{n \times (n-1)} \tag{2}$$

Once all individuals of the current population are ranked, the LRS procedure can be implemented according to the following pseudo-code:

```
LRS_ pseudo-code
  {
 •  Calculate the sum v = 1/(n - 2.001);
 •  For each individual 1 ≤ i ≤ n do {
      -  Generate a random number α ∈ [0,v];
      -  For each 1 ≤ j ≤ n do {
           -  If ( p(j) ≤ α ) {
                -  Select the j^th individual;
                -  Break;
              }
         }
    }

 }
```

2.4 Exponential Rank Selection (ERS)

The ERS [10] is based on the same principle as LRS, but it differs from LRS by the probability of selecting each individual. For ERS, this probability is given by the expression:

$$p(i) = 1.0 * \exp\left(\frac{-rang(i)}{c}\right) \tag{3.a}$$

with

$$c = \frac{\left(n * 2 * (n-1)\right)}{\left(6 * (n-1) + n\right)} \tag{3.b}$$

Once the n probabilities are computed, the rest of the method can be described by the following pseudo-code:

```
ERS_ pseudo-code
```

```
{
•  For each individual 1≤i≤n do {
     -  Generate a random number α ∈ [1/9 c, 2/c];
     -  For each 1≤j≤n do {
          -  If ( p(j) ≤ α ) {
               -  Select the jth individual;
               -  Break;
             } // end if
        } // end for j
   }// end for i
   }
```

2.5 Tournament Selection (TOS)

Tournament selection [11] is a variant of rank-based selection methods. Its principle consists in randomly selecting a set of k individuals. These individuals are then ranked according to their relative fitness and the fittest individual is selected for reproduction. The whole process is repeated n times for the entire population. Hence, the probability of each individual to be selected is given by the expression:

$$p(i) = \begin{cases} \frac{C_{n-1}^{k-1}}{C_n^k} \text{ if } i \in [1, n-k-1] \\ 0 \text{ if } i \in [n-k, n] \end{cases} \quad (4)$$

Technically speaking, the implementation of TOS can be performed according to the pseudo-code:

```
TOS_ pseudo-code
{
 •  Create a table t where the n individuals are placed in
    a randomly chosen order
 •  For i=1 to n do {
     -  for j=1 to n do {
        -  i1 = t(j);
        -  For m=1 to n do {
         -  i2 = t(j+m);
         -  If ( f(i1) > f(i2) ) the select i1 else select i2;
         }// end for m
        -  j = j+k;
       } // end for j
```

```
  }// end for i
}
```

2.6 Truncation Selection (ERS)

The truncation selection [10] is a very simple technique that orders the candidate solutions of each population according to their fitness. Then, only a certain portion p of the fittest individuals are selected and reproduced $1/p$ times. It is less used in practice than other techniques, except for very large population. The pseudo-code of the technique is as follows:

```
TRS_ pseudo-code
{
```

- Order the n individuals of $P(t)$ according to their fitness;
- Set the portion p of individuals to select (e.g. $10\% \le p \le 50\%$);
- $sp = \mathrm{int}(n \times p)$ // selection pressure;
- Select the first sp individuals;

```
}
```

3 DESCRIPTION OF THE PROPOSED TECHNIQUE

After implementing the six selection methods described in the previous section and tested them on the optimization problem of a variety of test functions we found that results differ significantly from one selection method to another. This poses the problem of selecting the adequate method for real-world problems for which no posterior verification of results is possible.

To help mitigating this non-trivial problem we present in this section the outlines of a new selection procedure that we propose as an alternative, which can be useful when no single other technique can be used with enough confidence. Our technique is a dynamic one in the sense that the selection protocol can vary from one generation to another. The underling idea consists in finding a good compromise between proportional methods, which decrease the effect of selection pressure and assure some genetic diversity within the population, but may increase the convergence time; and elitist methods that reduce the convergence time but may increase the effect of selection pressure and, therefore, the risk of converging to local minima.

To achieve this goal, more than one selection method are applied at each generation, but in a competitive way meaning that only results provided by the selection operator with the best performance are actually taken into account. To assess and compare the performance of candidate selection methods two objective criteria are employed. The first criterion is the quality of solution; it can easily be measured as a function of the fitness f^* of the best individual. The second criterion is the genetic diversity, which is less evident to quantify than the first one. In this work, as a measure of the genetic diversity within the population $P(t)$ we propose the mean population diversity at generation ***t*** over 30 runs is calculated according to the formula:

$$Div(t) = \frac{1}{30}\sum_{k=1}^{30}\left(\frac{1}{\max(d_{ij})\ln(n-1)}\sum_{i=1}^{n}\sum_{j\neq i}^{n} d_{ij}(k,t)\right) \quad (5)$$

Where $D_{i\,j}$ (k, t) is the Euclidean distance between the i^{-th} and j^{-th} individuals at generation t of the k^{-th} run, n is the population size, max(d_{ij})is the maximum distance supposed between individuals of the population and *t* the number of generations or iterations. As a measure of the quality of the solution at each generation we used the criterion

$$Quality = \frac{f^*}{\sqrt{f_{\max}^2 + f_{\min}^2}} \quad (6)$$

Where $f_{\max}$ and $f_{\min}$ denote respectively the maximum and the minimum values of the fitness at generation t, and $f^* = f_{\max}$ or $f^* = f_{\min}$ depending on the nature of the problem, which can be either a maximization or a minimization problem. Finally, in order to combine the two criteria in a unique one we used the relation

$$C = \frac{1}{t}Div + \frac{t-1}{t}Quality \quad (7)$$

4 EXPERIMENTAL STUDY

In this section, we present some examples of numerical results obtained by applying the six selection methods, as well as the proposed Combined Selection (CS) procedure, described in the previous sections, to the problem of genetic optimization of a set of four well-known benchmark functions [12]. The first function is a classical multi modal and high dimension function defined on the interval [-30,30] by the expression:

$$\min(f(x)) = -20 \times e^{\left(-0.2\times\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2}\right)} - e^{\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right)} \quad (8)$$

The second test function is also a mono-dimensional example but it is a deceptive one in the sense that it possesses, as depicted by Fig. 1, two local maxima in which optimization algorithms may be trapped.

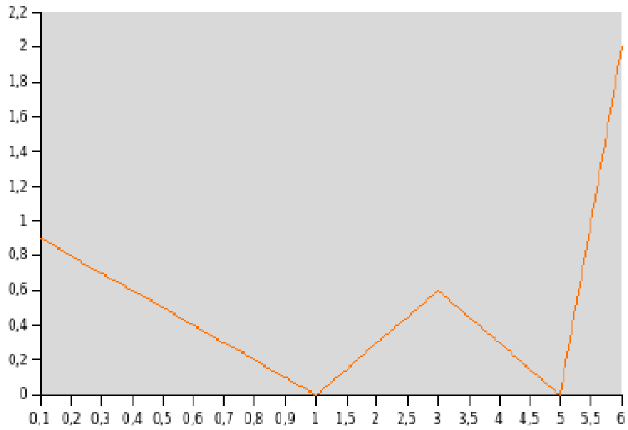

Figure 1. The deceptive test function $f_2(x)$.

$f_2(x)$ is defined on the interval $[0,6]$ by the expression

$$f_2(x)=\begin{cases}1-x \ \text{ for } x\in[0,1]\\ 0.3\times x-1 \ \text{ for } x\in[1,3]\\ -0.3\times x-5 \ \text{ for } x\in[3,5]\\ 2\times x-10 \ \text{ for } x\in[5,6]\end{cases} \tag{9}$$

The third example is the bi-variable Shubert function [3] defined, for $-10\le x_1\le 10$ and $-10\le x_2\le 10$, by

$$f_3(x_1,x_2)=(\sum_{i=1}^{5} i\cos((i+1)x_1+i)(\sum_{i=1}^{5} i\cos((i+1)x_2+i)) \tag{10}$$

And the last example is the bi-variable instance of the more general Rastrigin function [4] defined by:

$$f_4(x_1,x_2,\ldots,x_n)=10n+\sum\nolimits_{i=1}^{n}(x^2-10\cos(2\pi\times x_i)) \tag{11}$$

for $-5.12\le x_i\le 5.12,\ i=1,2,\ldots,n$

Results are summarized in Tables 1 to 3. Table 1 shows for each selection and each test function, the number of generations the GA needed to converge to an acceptable solution.

Analysis of Table I shows significant differences in convergence speed of the GA for the six studied selection methods, particularly in the case of the deceptive example, f_2.

Table 1. Number of Generations needed for convergence

Test Functions	Selection Method						
	RWS	SUS	LRS	ERS	TOS	TRS	CS
f_1	40	34	48	16	7	5	9
f_2	229	173	290	290	14	18	27

f_3	43	16	18	18	10	8	14
f_4	27	9	14	14	3	3	5

Table 2. Percentage of Selection Pressure

Test Functions	Selection Method						
	RWS	SUS	LRS	ERS	TOS	TRS	CS
f_1	12.3	12.2	22.8	22.9	80	70	84
f_2	2.3	2.3	1.3	5.4	75	33	70
f_3	10	4.8	4.8	4.8	80	65	80
f_4	25.1	19	4.8	4.8	65	50	70

Table 2 shows the percentage of selection pressure for each studied selection method and each function. The selection pressure, sp, of a given selection method is defined as the number of generations after which the best individual dominates the population. As to the percentage of selection pressure, it is defined by $sp_{\min}/sp$ where $sp_{\min}$ denotes the minimal selection pressure observed among all the studied parent selection methods. We can remark that proportional methods maintain the genetic diversity more than elitist ones.

Table 3 provides sample results related to another aspect of this study. It is the aspect of quality assessment of the optimum provided by the GA for each test function and for each selection method, including the combined selection method we proposed in this work. To measure this quality, we used the relative error

$$\frac{\Delta f}{f} = \frac{\left| f^* - f \right|}{f} \tag{12}$$

where f^* is the optimum provided by the algorithm and f the actual optimum, which is a priori known.

Table 3. Relative errors in percentage of the optima

Test Functions	Selection Method						
	RWS	SUS	LRS	ERS	TOS	TRS	CS
f_1	18	7	7	7	5	7	**4**
f_2	5	7	0	5	0	0	**0**
f_3	34	34	34	34	34	34	**0**
f_4	98	98	96	96	95	96	**0**

By analysing this table we can see clearly that the proposed method of selection performs better than the six studied selection methods.

5 CONCLUSION

In this paper, six well-known selection methods for GAs are studied, implemented and their relative performance analysed and compared using a set of four benchmark functions. These methods can be categorised into two categories: proportional and elitist. The first category uses a probability of selection proportional to the fitness of each individual. Selection methods of this category allow maintaining a genetic diversity within the population of candidate solutions throughout generations, which is a good property that prevents the GA from converging to local optima. But, on the other hand, these methods tend to increase the time of convergence.

By contrast, selection methods of the second category select only the best individuals, which increases the speed of convergence but at the risk of converging to local optima due to the loss of genetic diversity within the population of candidate solutions.

Starting from these observations, we have conducted a preliminary study aimed at combining the advantages of the two categories. This study resulted in a new combined selection procedure whose outlines are presented in this paper. The main idea behind this procedure is the use of more than one selection method in a competitive way together with an objective criterion which allows choosing the best selection method to adopt at each generation.

The proposed technique was successfully applied to the optimisation problem of a set of well-known benchmark functions, which encourages farther developments of this idea.

REFERENCES

1. Holland. J, "Adaptation in Natural and Artificial Systems", University of Michigan Press, An Arbor, MI, USA, 1975.
2. Goldberg. D. E, "Genetic Algorithms in Search, Optimization and Machine Learning", Addison-Wesley, New York, NY 1989.
3. Michalewicz. Z, "Genetic Algorithms + Data Structures=Evolution Programs", Third Edition, Springer, 2007.
4. DeJong. K. A, "Evolutionary Computation : a unified approach", MIT Press, 2007.
5. Eiben. A. E, Smit. S. K "Parameter tuning for configuring and analyzing evolutionary algorithms", Swarm and Evolutionary Computation, 2011; 1(1) :19–31.
6. Bäck. T, "Selective pressure in evolutionary algorithms: a characterization of selection mechanisms", In Evolutionary Computation, 1994, IEEE World Congress on Computational Intelligence, 1994(1):57-62.
7. Bäck. T, Hoffmeister. F, "Extended selection mechanisms in genetic algorithms". In Proceedings of the Fourth International Conference on Genetic Algorithms 1991; 92-99.

8. Blickle. T, Thiele. L, “A comparison of selection schemes used in genetic algorithms”, TIK-Report 11, TIK Institut fur Technische und Kommunikationsnetze, Swiss Federal Institute of Technology, Dec. 1995.

9. Goldberg D. E, Deb.K, “A comparative analysis of selection schemes used in genetic algorithms”, In Foundations of Genetic Algorithms, Ed., Morgan Kaufmann, 1991; 69-93.

10. Hancock. P, “A comparison of selection mechanisms”, . In Handbook of Evolutionary Computation, Eds . IOP Publishing and Oxford University Press., Bristol, UK, 1997

11. K. Hingee. K, Hutter. M, “Equivalence of probabilistic tournament and polynomial ranking selection”, In Evolutionary Computation, 2008. CEC 2008. (IEEE World Congress on Computational Intelligence). IEEE Congress on 2008; 564-571.

12. Talbi. E, “*Metaheuristics : from design to implementation*”, John Wiley and Sons, USA, 2009.