International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 2 Issue 11, November - 2013

Parent Selection Operators for Genetic Algorithms

Khalid Jebari[1], Mohammed Madiafi[2], Abdelaziz Elmoujahid[1]

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

In this paper, an experimental study of six well known selection methods has conducted to a new technique of selection. Dynamic selection (DS), the proposed technique, exploits the advantages of each selection methods in terms of quality of solution and population diversity. Indeed, dynamic selection is based on two parameters that allow to decide the quality of candidate solutions and the genotypic diversity. The famous 0-1 Knapsack Problem is used to illustrate the efficiency of DS.

***Index Terms*— Genetic algorithms, evolutionary computation, selection operators, selection pressure, Genotypic diversity.**

I. Introduction

Genetic algorithms (GAs) are stochastic methods based on the concepts of genetics and Darwinian evolution. They stimulate the genetic process of natural populations that evolve according to the principle of *survival of the fittest* [1]. By imitating this process, genetic algorithms are able to propose solutions to a variety of hard real-world problems in many application domains including engineering, medicine, computational science, bioinformatics, logistics, forecasting, manufacturing, etc. [2].

In order to conceive a genetic solution to any optimisation problem we first need to represent each candidate solution to the problem, called individuals [3]. A fitness function is required for assigning a figure of merit to each candidate solutions. Hence, starting from an initial population of randomly generated individuals, GAs evolve this population, throughout iterations called generations. The individuals, during the reproductive phase, are selected from the population and recombined, producing offspring for the next generation. Parents are selected from the population using a scheme, which furthers better individuals. Having selected two parents, their chromosomes are recombined, typically using mechanisms of crossover and mutation. More formally, a standard GA can be described by the following pseudo-code:

// GA algorithm

- 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 {
  - $t \leftarrow t+1$
  - Select parents for offspring production
  - Apply crossover 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 (i) the way parents are selected for offspring production, (ii) selected parents are crossed, and (iii) individuals are mutated, among other parameters [4].

One of the most important parameters that may influence the performance of a GA is treated in this paper. It is the parent selection operator (PSO). A PSO 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]. Several parent selection operators, which can lead to different results, are proposed in the literature [5-8]. However, despite decades of research, there are no general guidelines or theoretical support concerning the way of selecting a good PSO for each application problem [6]. This can be a serious problem because an inadequate PSO can lead the GA in rapid convergence and inefficiency.

To illustrate this problem we will consider a Np-Hard problem often used in literature that is the Knapsack problem. The rest of the paper is organized as follows: section II gives a brief description of Knapsack problem, section III presents each studied operator; section IV is dedicated to the presentation of a heuristic proposed in order to help reducing the influence of PSO on the global performance of a GA. Experimental study is discussed in section V. Finally, section VI contains the concluding remarks.

II. Knapsack problems

Knapsack problems have been amply studied sine the Dantzig works, because of their immediate applications in industry and financial management. In its simplified form, Knapsack problem is a combinatorial problem which seeks a subset of items that the corresponding profit sum is maximized without exceeding the capacity of the knapsack. (KP) is defined as follows: given a set of items (n), each with a weight $W[i]$ and a profit $P[i]$, with $i=1,..,n$. The goal is to determine the number of each item to include in the knapsack so that the total weight is less than some given limit (C) and maximizes the profit sum.

$$\max \sum_{i=1}^{n} P_i X_i \qquad (1)$$

Subject to the constraints:

$$\sum_{i=1}^{n} W_i C_i \leq C \qquad (2)$$

Using the GAs for solving the knapsack problem, a chromosome can be represented in a string having the length equal to the number of items. Each gene from the chromosome denotes whether an item is in the knapsack "1" or not "0". Whereas–the fitness of each chromosome is calculated by summing up the benefits of the items included in the knapsack, but considering that the capacity of the knapsack is not exceeded. If the capacity of the chromosome is greater than the capacity of the knapsack then one of the bits in the chromosome whose value is "1" is inverted and the chromosome is checked again.

III. PARENT SELECTION OPERATORS FOR GAS

As mentioned before, six different SOP 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 operator.

A. *The Roulette Wheel Selection (RWS)*

The salient characteristic of this operator is the fact that it gives to each individual i of the current population a probability $p(i)$ of being selected, proportional to its fitness $f(i)$

$$p(i) = \frac{f(i)}{\sum_{j=1}^{n} f(j)} \qquad (3)$$

where n denotes the population size in terms of the number of individuals.

The RWS can be implemented according to the following pseudo-code

- Calculate the sum $S = \sum_{i=1}^{n} f(i)$
- For each individual $1 \leq i \leq n$ do {
  - Generate a random number $\alpha \in [0, S]$
  - $iSum = 0$; $j = 0$
  - Do {
    - $iSum \leftarrow iSum + f(j)$
    - $j \leftarrow j + 1$

    } while($iSum < \alpha$ 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.

B. *The Stochastic Universal Sampling (SUS)*

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

- Calculate the mean $\overline{f} = 1/n \sum_{i=1}^{n} f(i)$
- Generate a random number $\alpha \in [0,1]$
- $Sum = f(1)$; $delta = \alpha \times \overline{f}$; $j = 0$
- Do {
  - If ($delta < Sum$) {
    - select the jth individual
    - $delta = delta + Sum$

    }

    else {
    - $j = j + 1$
    - $Sum = Sum + f(j)$

    }

  } while ($j < n$)

C. *The Linear Rank Selection (LRS)*

LRS 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

$$p(i) = \frac{rank(i)}{n \times (n-1)} \qquad (4)$$

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

- Calculate the sum $v = \dfrac{1}{n - 2.001}$
- For each individual $1 \leq i \leq n$ do {
  - Generate a random number $\alpha \in [0, v]$
  - For each $1 \leq j \leq n$ do {
    - If ($p(j) \leq \alpha$) {
      - Select the j^{th} individual
      - Break

      }

    }

  }

D. *The Exponential Rank Selection (ERS)*

The ERS 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

$$p(i) = 1.0 * \exp\left(\frac{-rang(i)}{c}\right) \qquad (5.a)$$

with

$$c = \frac{(n*2*(n-1))}{(6*(n-1)+n)} \tag{5.b}$$

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

- For each individual $1 \le i \le n$ do {
  - Generate a random number $\alpha \in \left[\frac{1}{9}c, \frac{2}{c}\right]$
  - For each $1 \le j \le n$ do {
    - If ($p(j) \le \alpha$) {
      - Select the *j*th individual
      - Break

      } // end if

    } // end for j

  }// end for i

E. The Tournament Selection (TOS)

Tournament selection is a variant of rank-based selection operators. 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

$$p(i) = \begin{cases} \dfrac{C_{n-1}^{k-1}}{C_n^k} \text{ if } i \in [1, n-k-1] \\ 0 \text{ if } i \in [n-k, n] \end{cases} \tag{6}$$

Technically speaking, the implementation of TOS can be performed according to the 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 {
    - $i_1 = t(j)$
    - For $m = 1$ to n do {
      - $i_2 = t(j+m)$
      - If ($f(i_1) > f(i_2)$) the select i_1 else select i_2

      }// end for m
    - $j = j + k$

    } // end for j

  }// end for i

Another way to implement the same technique is described by the following pseudo-code

- For $i = 1$ to n do {
  - Generate a random number $i_1 \in [1, n]$
  - For $j = 1$ to n do {
    - Generate a second random number $i_2 \in [1, n]$ with $i_2 \ne i_1$
    - If ($f(i_1) > f(i_2)$) the select i_1 else select i_2

    } // end for j

  }// end for i

F. The Truncation Selection (TRS)

The truncation selection 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:

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

IV. Description of the Proposed Method

After implementing the six selection operators 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 operator to another. This poses the problem of selecting the adequate operator 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 [9]. 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 operator are applied at each generation, but in a competitive way meaning that only results provided by the operator with the best performance are actually taken into account. To assess and compare the performance of candidate operators 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 value of the Hamming distances between the best individual i^* and all other individuals of $P(t)$, i.e.

$$< H >= \frac{1}{n} \sum_{i=1}^{n} H(i, i^*) \tag{7.a}$$

where $H(i,i^*)$ is the Hamming distance between individuals, or chromosomes, i and i^*, defined by

$$H(i,i^*) = \sum_{k=1}^{l} i(k) \oplus i^*(k) \tag{7.b}$$

with l denoting the length of chromosomes in terms of number of bits and $\oplus$ the "exclusive or" logical operator.
However, as the genetic diversity should, in principle, decrease with generations, the actual criterion for measuring the quality of diversity should be a decreasing function of $< H >$. For this reason, in this work, we used the criterion

$$C_1 = \exp\left(-\frac{< H >}{t}\right) \tag{8}$$

where t denotes the number of generations or iterations.
And as a measure of the quality of the solution at each generation we used the criterion

$$C_2 = \frac{f^*}{f_{\max}^2 + f_{\min}^2} \tag{9}$$

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

$$C = \frac{1}{t} C_1 + \frac{t-1}{t} C_2 \tag{10}$$

V. Numerical Results and Discussions

In this section, we implement the knapsack problem by varying the number of items (n=100, n=250, n=500) and generating a randomly uniform distribution of the weights and the profits.

$$W_i = N(1,10) \text{ and } P_i = N(1,10) \tag{11}$$

The capacity of the knapsack is equal to

$$C = 0.5 \times \sum_{i=1}^{n} W_i \tag{12}$$

Table I shows for each POS 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 POS, particularly in the case of the deceptive example, f_2.

TABLE I. Number of Generations Needed for Convergence

Test Functions	Parent Selection Operators					
	RWS	SUS	LRS	ERS	TOS	TRS
"50"	40	34	48	16	7	5
"100"	29	13	20	20	14	18
"250"	43	16	18	18	10	8
"500"	27	9	14	14	3	3

TABLE II. Percentage of Selection Pressure

Items	Parent Selection Operators					
	RWS	SUS	LRS	ERS	TOS	TRS
"50"	25	30	40	45	80	100
"100"	12.3	12.2	22.8	22.9	100	100
"250"	42.3	42	25	55.4	85	85
"500"	10	4.8	4.8	4.8	90	95

Table II shows the percentage of selection pressure for each studied PSO and each function. The selection pressure, sp, of a given PSO 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 PSO. We can remark that proportional methods maintain the genetic diversity more than elitist ones.

Table III 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 Knapsack problems and for each selection method, including the dynamic selection method proposed in this work.

TABLE III. The Optima Provided by the GA for 7 Selection Protocols

Items	Parent Selection Operators						
	RWS	SUS	LRS	ERS	TOS	TRS	DS
"50"	220	230	250	255	260	260	**280**
"100"	470	500	520	520	550	550	**560**
"250"	1060	1200	1240	1300	1350	1360	**1400**
"500"	2200	2250	2440	2450	2500	2550	**2600**

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

VI. Conclusion

In this paper, six well-known selection operators for GAs are studied, implemented and their performance analysed and compared using a knapsack problems. These operators can be categorised into two categories: proportional and elitist. The first category uses a probability of selection proportional to the fitness of each individual. Operators 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, operators 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 conducted in a new dynamic selection procedure whose outlines are presented in this paper. The main idea behind DS is the use of more than one selection operator in a competitive way together with two criterions which allow choosing the best operator to adopt at each generation.

The proposed technique was successfully applied to the optimisation problem, which encourages farther developments of this idea. It is very interesting to study experimentally the different techniques of the operator genetic crossover and use the technique proposed in this paper to deduce an appropriate scheme that allows to exploit the benefits of some well known crossover methods selected among the most commonly used in the literature. The same idea can be applied to the mutation operator.

REFERENCES

[1] D. E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley, New York, NY, 1989.
[2] Z. Michalewicz, Genetic Algorithms + Data Structures=Evolution Programs, 3rd ed., Springer, 1996.
[3] C. Reeves and J. Rowe, *Genetic Algorithms: Principles and Perspectives*, Kluwer Academic Publishers, Boston, MA, USA, 2003.
[4] K. DeJong, "Parameter settings in EAs: a 30 year perspective." In Parameter Setting in Evolutionary Algorithms, pp. 1-18, Springer, 2007.
[5] A. Tuson and P. Ross, "Adapting operator settings in genetic algorithms," Evolutionary Computation, Vol. 6, pp. 161-184, 1998.
[6] M. F. Azeem, "A novel parent selection operator in GA for tuning of scaling factors of FKBC," In Proceedings of the World Congress of Computational Intelligence, IEEE International Conference on Fuzzy Systems, Vancouver, BC, Canada, pp. 1742-1747, 2006.
[7] C. Cheng and J. Zuo, "Dynamic and recombination using self selecting crossover method for genetic algorithms," In proceedings of the 5th Int. Conf. Natural Computation, Jinan, Shandong, China, Vol. 1, pp. 586-596, 2008.
[8] P. Vajda, A.E. Aiben and W. Hordijk, "Parameter Control Methods for Selection Operators in Genetic Algorithms", in *Parameter Problem Solving from Nature*, pp. 620-630, Springer , Berlin/Heidelberg, 2008.
[9] K. Jebari, A. Bouroumi, A. Touhami, " An Experimental Study of Parent Selection Operators for Genetic Algorithms" In Proceedings of The 3rd International Conference on Software, Knowledge Information Management and Applications (SKIMA 09), pp. 56-61, Fes, Morocco, 2009.