Applied Mathematical Sciences, Vol. 5, 2011, no. 58, 2863 - 2881

Unsupervised Fuzzy Tournament Selection

Khalid Jebari, Abdelaziz El moujahid, Amina Dik,
Abdelaziz Bouroumi and Aziz Ettouhami

Laboratoire Conception et Systèmes (Microélectronique et Informatique)
Faculty of Sciences, Mohamed V-Agdal University, UM5, Rabat , Morocco
{khalid.jebari, elmoujahid, a.dik2003}@gmail.com
{bouroumi, touhami}@fsr.ac.ma

Abdelaziz Bouroumi

Modeling and Simulation Laboratory, Ben Msik Faculty of Sciences
Hassan II Mohammedia-Casablanca University, UH2MC
Casablanca, Morocco
a.bouroumi@gmail.com

Abstract

Tournament selection has been widely used and studied in evolutionary algorithms. The size of tournament is a crucial parameter for this method. It influences on the algorithm convergence, the population diversity and the solution quality. This paper presents a new technique to adjust this parameter dynamically using fuzzy unsupervised learning. The efficiency of the proposed technique is shown by using several benchmark multimodal test functions.

Keywords: Tournament Selection, Tournament Size, Fuzzy Clustering, Unsupervised Learning, Genetic Algorithms, Evolutionary Algorithms, Fuzzy C-means

1 Introduction

Genetic algorithms (GA) have proven, in recent decades, their effectiveness to solve many complicated problems in the real world. However, their performance depends heavily on certain parameters such as: probability of crossover, probability of mutation, population size and selection pressure. The ultimate goal is the good choice of these parameters [10] to maintain a dynamic balance between exploration and exploitation. This balance can be adjusted by:

- probability of crossover and probability of mutation that explore other areas of research space;
- population size that influences the diversity of the population;
- selection pressure that controls the selection of individuals from the current population to produce a new population in the next generation [22].

Research was therefore oriented towards techniques to dynamically adjust these parameters to improve the quality of the solution. In this paper, we are interested in the tournament selection due to its wide use in various optimization problems using genetic algorithms [1].

The standard tournament selection method consists in [12]:

1. randomly selecting k individuals from the population; // ($k > 1$, k: Tournament size)
2. selecting the individual who has the best fitness value from selected individuals in step 1;
3. Repeating steps 1. and 2. N times // (N: Population size).

Tournament selection depends largely on the parameter k. Several studies [1, 5, 6, 12, 13, 19, 20, 21] have shown the importance of the choice of k. The difficulty lies in the fact how to determine k, taking into consideration the state of the GA evolution. Choosing a large value of k leads to a strong pressure selection and the GA may converge to a local optimum. If k is small, we have a problem with individuals who may have a better value of adaptation without being candidates for selection by tournament [21]. Hence the purpose of this paper is to determine k in a dynamic way by using a process of unsupervised fuzzy clustering, taking into consideration the state of the GA evolution. The value of k is the number of the clusters given by the cluster algorithm.

Section II is devoted to the selection tournament. Our method is described in Section III. The numerical results are given and discussed in Section IV. Finally, section V contains our concluding and remarks.

2 Tournament selection

The tournament selection was attributed to Wetzel in an unpublished work, then it has been studied by Brindle in his doctoral thesis in 1981 [12]. Subsequently, several researchers were interested in this method [5, 6, 11, 16, 18]. More recently we find [14, 15, 23, 24, 25].

We can describe this method by the algorithm in Fig. 1:

```
// N :population size
t_rand: array of integer containing the indices of
          individuals in the population
t_ind_selected : an array of individuals indices
                    who will be selected
l = 0
For (i=0; i < k; i++)
{
    Shuffle t_rand elements;
    For (j=0; j < N; j += k)
    {
        I1 = t_rand(j);
        For (m=1; m < k; m++)
        {
            I2 = t_rand(j+m);
            if (f(I1) < f(I2))  I1 =I2;
            // f(Ii): Fitness of individual Ii
        }
        t_ind_selected(l) = I1;
        l += 1;
    }
}
```

Figure 1: Tournament Selection

Several studies have looked at the behavior of the tournament selection and the concept of selection pressure. Goldberg introduced takeover time as a criterion for comparison [12]. It is defined as the number of generations required for a single best individual to fill the entire population by using only the selection operator. Mühlenbein and Schlierkamp in Breeder Genetic Algorithm (BGA) have used the term selection pressure to measure progress in the population [20]. Blickle studied several methods of selection [5]. He introduced several comparison criteria based on the fitness distribution : Reproduction rate, Loss of diversity and Selection intensity [6].

There are several techniques to evaluate the selection pressure. We've collected the most used:

- Takeover time is [12]:

$$t = \frac{1}{\ln(k)}[\ln(N) + \ln(\ln(N))] \quad (1)$$

k: Tournament size , N: Population size

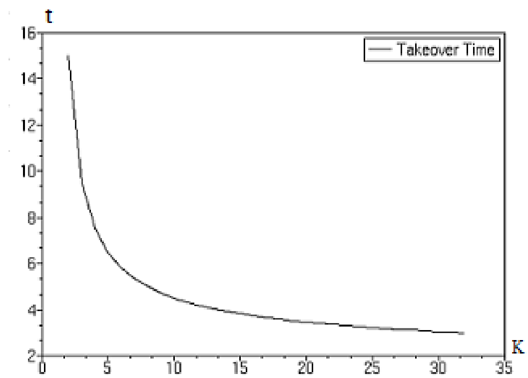

Figure 2: Takeover Time t(k) for Tournament Selection

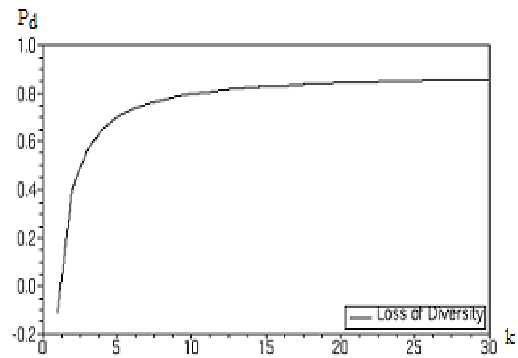

Figure 3: Loss of Diversity $P_d(k)$ for Tournament Selection

- Loss of diversity [5] is proportion of individuals not selected during selection phase:

$$P_d = k^{\frac{-1}{k-1}} - k^{\frac{-k}{k-1}} \tag{2}$$

- Motoki [19] recalculated the loss of diversity and demonstrated that:

$$P_d = \frac{1}{N}\sum_{j=1}^{N}\left(1 - \frac{j^k - (j-1)^k}{N^k}\right)^N \tag{3}$$

- Selection intensity used by Bäck [2], Mühlenbein [20] and Blickle [6] :

$$I \approx \sqrt{2\left(\ln(k) - \ln\left(\sqrt{4.14 * \ln(k)}\right)\right)} \tag{4}$$

Selection intensity, takeover time, reproduction rate and loss of diversity depend on the size of the tournament. This shows the importance of this

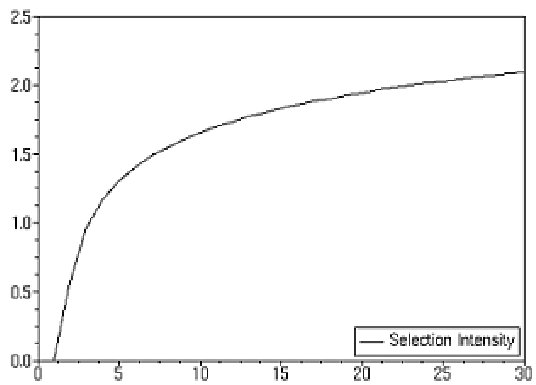

Figure 4: Selection Intensity I(k) for Tournament Selection

parameter to adjust the selection pressure. Miller and Goldberg stressed the importance of this parameter even in the presence of the noise effect [18].

We then present some techniques to find k:

- Goldberg and Deb [12] studied some values of k, having the form: $k = 2^i$(i=1,...,5). They started by $k = 2$ and showed that the tournament selection with size 2 is equivalent to rank selection.

- Julstrom and Robinson [16] introduced a tournament selection based on weight. The probability that individual j is selected is:

$$P_j = \frac{w^{k-j}(1-w)}{1-w^k} \tag{5}$$

  w: weight of individual j

$$k = \frac{N(1-r)}{1-r^N} \tag{6}$$

  r: exponential normalization factor

- Filipovi'c et al. [11] studied "Fine-grained tournament selection". They showed that the standard tournament selection does not allow precise setting of the balance between exploration and exploitation. In their method, the tournament size is not fixed, but takes a value among a set of values.

- Sokolov and Whitley [23] have developed another variant of tournament selection which they have named "Unbiased tournament selection" to solve the problem of individuals not sampled. They proposed a function to modify the ranking of individuals. But they did not discuss the problem of choosing the parameter k.

- P. Vajda et al. [24] proposed "Deterministic tournament-size control (DTC), the tournament size is a function of generation t, but their empirical formula depends on two parameters P_1 and P_2. The tournament size increases linearly from P_1 to P_2 for the first 1,000 generations, and binds to P_2 after.

$$k(t) = \begin{cases} \frac{t(p_2 - p_1)}{1000} + p_1 & \text{t} \in \{1,2,...1000\} \\ p_2 & \text{otherwise} \end{cases} \tag{7}$$

They chose: p_1, p_2 in {2,7,17}. They proposed also "Fuzzy Tournament Selection", where k fuzzified using the following equation:

$$k_t = (\mu + 0.5) * k_{t-1} \tag{8}$$

the parameter μ is calculated by the rules of fuzzy logic.

3 Proposed method

To dynamically choose k, our idea is to precede the selection phase by a classification stage that detects the presence of homogeneous groupings to determine the number of clusters.

This idea comes from the fact that if we know the number of clusters of the population, we have an idea of diversity of the population. So we will be able to say how we are going to increase the selection pressure, decrease it and adjust the proportion of individuals not sampled. This explains the use of the number of clusters detected as a value of k.

In addition, as lack of information on population, the fuzzy clustering offers a better possibility of modeling and management of overlapping clusters. Thus our approach is to introduce a fuzzy clustering based on two existing algorithms. The first is "unsupervised fuzzy learning" (UFL) [7], is a learning phase to automatically detect the number of clusters (c) present in the population (Fig. 5). The second is the fuzzy c-means (FCM) proposed by Bezdek [4] for an optimization phase that improves the distribution of individuals in c clusters. A third phase is a validation (VAL) using the normalized partition entropy defined as [4]:

$$h(U) = -\frac{1}{\log(c)} \frac{1}{n} \sum_{i=0}^{n} \sum_{j=0}^{c} [u_{ij} \log(u_{ij})] \tag{9}$$

U : *matrix of membership degrees*; c : *Number of clusters*

to evaluate the quality scores obtained in order to choose the best distribution.

In each generation of GA, we applied our selection technique that we named "Unsupervised Fuzzy Tournament Selection" (FUT). FUT uses UFL-FCM-VAL, described in Fig. 6 to find the number of clusters of the population used as the tournament size. The pseudo-code of FUT is presented in Fig. 7. The GA modified is given in Fig. 8.

I_i: individual, $1 \le i \le N$, with p is Individual dimension, N: population size

choose: a similarity threshold S_{th}
Initialization: c=1, $V_1=I_1$, V_1 cluster Center for the first Class
for meseasuring the similarity beteween I_i and I_j:

$$S(i,j) = 1 - \frac{d(I_i, I_j)}{\sqrt{p}};\ d(I_i, I_j)\ \textit{Euclidean distance measure} \tag{10}$$

For i=2 to n do
{
    If $(\max_{1 \le j \le c}(S(i,j)) < S_{th})$ {
        c = c + 1;
        $V_c = I_i$;
    }
    Else
        Update V_j by using

$$V_j = \frac{\sum_{k=1}^{n} (u_{jk})^m I_k}{\sum_{k=1}^{n} (u_{jk})^m};\ 1 \le j \le c;\ m = 2; \tag{11}$$

    u_{jk} membership degree of I_k to class V_j

$$u_{jk} = 1 - \frac{d(I_j, I_k)}{\sqrt{p}}; \tag{12}$$

    return c,u and V
}

Figure 5: UFL

```
S_th-step =0.01
C_optimum = 2   // C_optimum :Number of clusters for
                // a minimal entropy
h_min = 1;   // h_min : Minimal entropy
 S_th: similarity threshold
S_th-min = 0.1; S_th-max = 0.99;

For (S_th=S_th-min; S_th<S_max; S_th += S_th-step)
{
  Apply UFL for population of individuals
  Apply FCM for population of individuals
  Calculate h () Using U and V (Eq. (9))

   if (h_min>h() ) {h_min=h(); C_optimum = c}


}

  return C_optimum to use as tournament size k
```

Figure 6: UFL-FCM-VAL

4 Numerical Results and Discussions

We compared our technique (FUT) with :

- Standard tournament selection (FTS), we chose the values for k: 4,8,16.
- Deterministic tournament size (DTC): k is given by (8).

we present numerical results to the problem of genetic optimization on a set of 14 well-known test functions commonly used in genetic algorithms (see appendix A).

We calculated, for the three methods (FTS, DTC and FUT), relative error and takeover time.

We chose the following parameters:

- Random initialization with population size N= 100;
- the coding we used is the real code.
- For the crossover operator, we used Simulated Binary Crossover SBX [8].
- For the mutation, we considered the polynomial mutation [9]. To simulate other tests, we also used the Gaussian mutation [17] of adding Gaussian noise to each value of genes that form the chromosome (candidate solution) and the uniform mutation is to select random genes and mutate them, ie, changing their values in the area of the test function.

```
k = c // k: Tournament size
      // c: Number of clusters given by UFL-FCM-VAL (Fig. 6)
      // N population size
t_rand : array of integer containing the indices of
         individuals in the population
t_ind_selected : an array of individuals indices who
                 will be selected
l = 0

For( i=0; i<k; i++)
{
  Shuffle t_rand elements;
  For (j=0; j<N; j=j+k)
  {
    I1 = t_rand(j);
    For (m=1; m<k; m++)
    {
      I2=t_rand(j+m);
      if (f(I1)< f(I2)) I1 = I2
      // f(Ii): Fitness of individual Ii
    }
    t_ind_selected(l) = I1
    l = l + 1;
  }
}
```

Figure 7: FUT

- For the stopping, criterion we have chosen as the maximum number of generations $t_{max} = 100$. Another stop criterion is used: percentage of optimum in population (Eq. (13)).

$$\frac{number\_of\_optimum}{N} * 100 < 95 \tag{13}$$

A summary of these parameters is in Table 1

Table 2 shows the results of an aspect considered in this paper is to evaluate the quality of the optimum provided by the three techniques. We proposed, in this work, to measure this quality by using relative error in percentage (PER) (Eq. (14))

$$PER = \frac{\Delta f}{f} = \frac{|f^* - f|}{f} \tag{14}$$

f^*, in Eq. (14), is the optimum provided by the algorithm for each technique, f the actual optimum, which is a priori known.

```
t=0      //t: Generation number
Random initialization of individuals P(t)
Evaluate P(t)
condition1= Eq.~ (13)
tmax=100 //Maximum of generations
While (condition1 AND t < tmax) Do
{
          t += 1
          Select P(t) from P(t-1) by FUT
          Apply Crossover for P(t)
          Apply mutation for P(t)
          //Population after crossover and mutation
          Create new population P'(t)

  P(t)=P'(t)
  Evaluate P(t)
  Calculate the proportion of optimum in population
}
Give the best individual(s)
```

Figure 8: GA modified

Population size	100
Crossover	SBX
Crossover probability	0.97
Mutation	polynomial
Mutation probability	0.03
Coding	Real
Maximum number of generations	100

Table 1: AG Parameters

Functions	Technique used			Best Technique
	FTS	DTC	FUT	
f1	7.8	0	0	FUT,DTC
f3	11	16	0	FUT
f4	19	15	4	FUT
f5	5	0	0	FUT,DTC
f6	0.5	0.8	0.5	FUT,FTS
f7	5	13	1	FUT
f8	5	1	0	FUT
f9	19	0	0	FUT ,DTC
f10	13	5	0	FUT
f11	5	6	1	FUT
f13	3	3	3	FUT,DTC,FTS
f14	4.2	7	0.7	FUT
f15	8	5	2	FUT
f16	11	9	1.4	FUT

Table 2: Relative Errors in Percentage of the Optima provided by the GA for each technique

We note that our technique provides significantly better results compared to FTS that is the standard selection method and to DTC.

Our second comparison is devoted to the takeover time. This criterion can be directly calculated by the Goldberg equation for the case of FTS. But we have experimentally determined takeover time for each technique.

Indeed, table 3 presents the number of generations needed for the GA converges, using only the selection operator, for each technique and test functions.

We note that the results found for FTS and DTC are independent of the function. We can conclude that these two techniques do not respect the functions characteristics that differ according to their distributions, and their nature unimodal or multimodal. While our technique has an acquisition time that depends on the nature of the test function.

Initially the population is diverse, so a high number of clusters is produced. therefore, the takeover time is low and the selection pressure is high. Over generations, if diversity is big, takeover time decreases and vice versa. In contrast to DTC, the takeover time which depends only on generation number, the values of takeover time are almost identical for the different test functions. While for the FTS, the takeover time is the same for each test functions and for each generation.

Fig. 9 shows the variation of takeover time for 3 functions. Ackley function (f1) is a multimodal function, cosine mixture function (f2) which has an unimodal function in the interval [-1, 1], but its optimum value depends on its

Fonction	FTS (4)	FTS (8)	FTS (16)	DTC	FUT
f1	7	5	3	12	6
f3	8	5	4	11	5
f4	8	5	4	12	3
f5	7	5	3	12	8
f6	7	5	3	12	8
f7	7	5	4	12	10
f8	7	5	4	12	7
f9	7	5	4	11	6
f10	7	5	4	12	8
f11	7	5	4	12	7
f13	8	5	4	12	10
f14	8	5	4	12	8
f15	8	5	4	12	8
f16	7	6	4	12	15

Table 3: Takeover Time for test functions

	Ackley Function			Cosinus Mixture Function			Griewank Function		
Genration	FTS(4)	DTC	FUT	FTS(4)	DTC	FUT	FTS(4)	DTC	FUT
1	0.5601	0.430702	0.802621	0.5601	0.430702	0.802621	0.5601	0.430702	0.802621
2	0.5601	0.430702	0.733084	0.5601	0.430702	0.733084	0.5601	0.430702	0.695625
3	0.5601	0.430702	0.695625	0.5601	0.430702	0.605625	0.5601	0.430702	0.605625
4	0.5601	0.430702	0.640789	0.5601	0.430702	0.5601	0.5601	0.430702	0.5601
5	0.5601	0.430702	0.640789	0.5601	0.430702	0.5601	0.5601	0.430702	0.5601
6	0.5601	0.430702	0.605625	0.5601	0.430702	0.502997	0.5601	0.430702	0.43072
7	0.5601	0.430702	0.5601	0.5601	0.430702	0.502997	0.5601	0.430702	0.43072

Table 4: Loss of Diversity for f1, f2, f4

size and Griewank function (f4), a unimodal function not convex. Selection pressure is the same whatever the test function for the FTS, it is a function of generation for the DTC, whereas the FUT depends on the nature of the function.

The loss of diversity (Eq. (3)) is a function of k and N. In our simulation N is fixed, k is variable for DTC and FUT and fixed for the FTS. For FUT, the value of k represents the number of clusters present in the population over many generations. So, as shown in table 4, we have a variation in the FUT column, resulting in high diversity at the beginning of the evolution, which decreases over generations.

We give an indication of the different k values over generations for three test functions:

f1=[16,10,8,6,6,5,4,3,2,2,2,2,3,3,2,2];

f2=[16,10,5,4,4,3,3,2,2,2,2,2,2,2,2,2];

f4=[16,8,5,4,3,2,2,2,2,2,2,2,2,2,2,2];

We note that the loss of diversity for the FTS is the same whatever functions. For the DTC, k varies depending on the number of generations. But

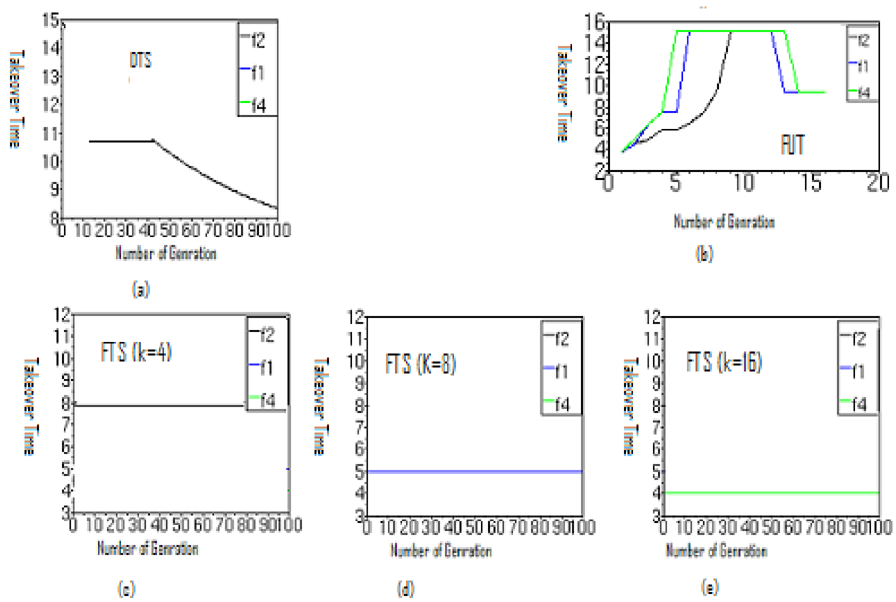

Figure 9: Takeover Time for 3 techniques

in the case of the three functions chosen, the GA has already converged even before the size of the tournament changed its values. While our technique respects the nature of the distribution and nature of the problem.

To understand the variation in FUT according to the nature of the function, we studied the proportion of individuals having the best fitness value for three functions: Sphere function (f13), a unimodal function, Weierstrass function (f16), a multimodal function of dimension 1 and Goldstein-Price Problem, a multimodal function of dimension 2.

For both types of functions studied, unimodal or multimodal, Bäck and Hoffmeister [3] have reported the following remarks :

- Multimodal function requires an explorative character of the search. To achieve this behaviour, a low pressure selection can be used in order to maintain a large genotypic diversity;
- Unimodal function requires a high pressure selection for forcing the search space into the gradient direction and exploiting the search space better.

Unfortunately, for a real word problem, we haven't any ideas about the fitness function properties. To solve this problem, FUT provides a powerful mechanism by respecting the function topology. This mechanism is clarified in experimental result of Fig. 10.

For the sphere function represented in Fig. 10a, which is an unimodal function, we note that, in the first generations for the GA process, FUT has

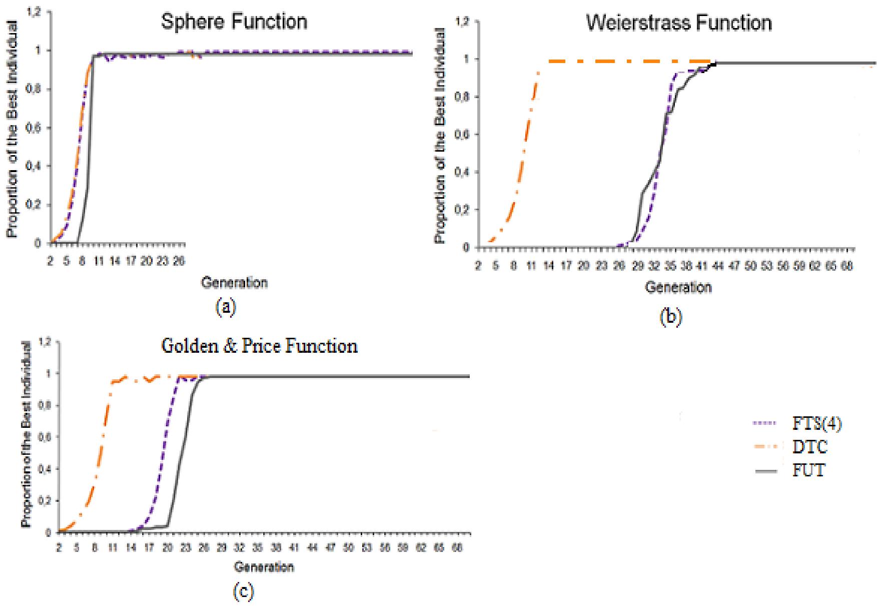

Figure 10: Proportion of the best individual for FTS, DTD and FUT

	Ackley Function			Cosinus Mixture Function			Griewank Function		
Generation	FTS(4)	DTC	FUT	FTS(4)	DTC	FUT	FTS(4)	DTC	FUT
1	0.018	0.134	0	0.018	0.134	0	0.018	0.134	0
2	0.018	0.134	0	0.018	0.134	0	0.018	0.134	0
3	0.018	0.134	0.0	0.018	0.134	0.006	0.018	0.134	0.018
4	0.018	0.134	0.006	0.018	0.134	0.018	0.018	0.134	0.018
5	0.018	0.134	0.006	0.018	0.134	0.018	0.018	0.134	0.134
6	0.018	0.134	0.0067	0.018	0.134	0.134	0.018	0.134	0.134
7	0.018	0.134	0.018	0.018	0.134	0.017	0.018	0.134	0.134
8	0.018	0.134	0.018	0.018	0.134	0.134			
9	0.018	0.134	0.134	0.018	0.134	0.134			

Table 5: Probability of individuals not sampled for the functions: f1, f2, f4

a significant diversity compared to the others techniques. Afterwards, during the GA evolution, the deflection angle of FUT curve shows that FUT has a strong selection pressure. For Both multimodal functions in Fig. 10b and Fig. 10c, we note that FUT has a lower selection pressure compared to FTS and DTC.

Now we considered the problem of individuals not sampled by tournament selection. The probability that an individual is not sampled is given by Eq. (7)

For our technique the probability of individuals not sampled is almost null at the beginning of the algorithm, and grows over the generations.

Our technique could make a balance between the selection pressure and the probability of individuals not sampled. When the tournament size is high, the selection pressure is high too. On the other hand, if the tournament size is high, the probability of the individuals not sampled is low. This balance cannot be regulated by the parameter setting with the number of the generations such as with the DTC, but with a study based on the clustering of the population such as FUT.

5 Conclusion

In this paper we showed the importance of tournament size. We have proposed a technique to determine this parameter which is based on unsupervised fuzzy clustering. We have shown that our technique can not only solve the problem of tournament size but also to fill the gap in the selection tournament which is the percentage of individuals not sampled. We used several benchmark functions to show that our technique adjusts dynamically, across generations, the pressure of selection, the diversity of the population and can solve the problem of individuals not sampled. So our technique was able to strike a balance between exploitation and exploration. This technique can also be implemented in genetic programming for solving difficult problems that require a method to dynamically adjust the selection pressure.

In future work, we propose a similar technique to dynamically adjust other parameters of genetic algorithms, namely the probability of mutation and crossover probability.

Appendix A: Test Functions

Name	Description	Dim	Characteristics
f1	Ackley Problem $minf(x) = -20 \cdot e^{\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2)}\right)} - e^{\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right)} + 20 + e$ $-30 \leq x_i \leq 30, x^* = (0,0,...,0), f(x^*) = 0$	n=5	Multimodal High dimension Secable, Regular
f2	Cosinus mixture Problem $minf(x) = \sum_{i=1}^{n} x_i^2 - 0.1\sum_{i=1}^{n} cos(5\pi x_i) with -1 \leq x_i \leq 1, x^* = (0,0,...,0), f(x^*) = 0.1 * n$	n=10	Unimodal, Local solution= Global solution
f3	Goldstein-Price Problem $minf(x) = [1 + (x_0 + x_1 + 1)^2(19 - 14x_0 + 3x_0^2 - 14x_1 + 6x_0x_1 + 3x_1^2)][30 + (2x_0 - 3x_1)^2(18 - 32x_0 + 12x_0^2 + 48x_1 - 36x_0x_1 + 27x_1^2)]$ $-2 \leq x_i \leq 2, x^* = (0,-1)f(x^*) = 3$	n=2	Several Local Minima
f4	Griewank Problem $minf(x) = 1 + \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}})$ $-600 \leq x_i \leq 600, x^* = (0,0,...,0), f(x^*) = 0$	n=5	Unimodal; High Dimension Non convex
f5	Levy et Montalvo Problem 1 $minf(x) = \frac{\pi}{n}\left(10\sin^2(\pi y_1) + \sum_{i=1}^{n-1}(y_i - 1)^2[1 + 10\sin^2(\pi y_{i+1})] + (y_n - 1)^2\right)$ $y_i = 1 + \frac{1}{4}(x_i + 1), -5 \leq x_i \leq 5, x^* = (-1,-1,...,-1), f(x^*) = 0$	n=5	Multimodal
f6	Levy et Montalvo Problem 2 $minf(x) = 0.1\left(\sin^2(3\pi x_1) + \sum_{i=1}^{n-1}(x_i - 1)^2\left[1 + \sin^2(3\pi x_{i+1})\right] + (x_n - 1)^2\left[1 + \sin^2(2\pi x_n)\right]\right)$ $-5 \leq x_i \leq 5, x^* = (1,1,...,1), f(x^*) = 0$	n=5	Multimodal
f7	Paviani Problem $minf(x) = \sum_{i=1}^{10}\left[(\ln(x_i - 2))^2 + (\ln(10 - x_i))^2\right] + \left(\prod_{i=1}^{10} x_i\right)^{0.2}$ $2 \leq x_i \leq 10, x^* \approx (9.350266, 9.350266, ..., 9.350266), f(x^*) = -45.778470$	n=10	With several local minimas and one global minima
f8	Rastrigin Problem $minf(x) = 10n + \sum_{i=1}^{n}\left[x_i^2 - 10\cos(2\pi x_i)\right]$ $-5.12 \leq x_i \leq 5.12, x^* = (0,0,..,0), f(x^*) = 0$	n=10	multimodal complex with a very large number of regularly distributed Optima
f9	Rosenbrock problem $minf(x) = \sum_{i=1}^{n-1}\left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2\right]$ $-30 \leq x_i \leq 30, x^* = (1,1,...,1), f(x^*) = 0$	n=2	Unimodal Non convex
f10	Schewefel problem 1 $minf(x) = 418.9829n - \sum_{i=1}^{n} x_i \sin\left(\sqrt{\|x_i\|}\right)$ $-500 \leq x_i \leq 500, x^* = (420.97, 420.97, ..., 420.97), f(x^*) = 0$	n=2	Multimodal
f11	Easom problem $minf(x) = -\left[\prod_{i=1}^{n}\cos(x_i) * \left(\exp\left(-\sum_{i=1}^{n}(x_i - \pi)^2\right)\right)\right]$ $-100 \leq x_i \leq 100, x^* = (\pi, \pi, ..., \pi), f(x^*) = -1$	n=3	Unimodal
f12	Ellipsoidal problem $minf(x) = \sum_{i=1}^{n}(x_i - i)^2, -n \leq x_i \leq n, x^* = (1,2,...,n), f(x^*) = 0$	n=10	Unimodal

Name	Description	Dim	characteristics
f13	Sphere Problem $minf(x) = \sum_{i=1}^{n} x_i^2$ $-5.12 \leq x_i \leq 5.12, x^* = (0, 0, ..., 0), f(x^*) = 0$	n=5	Unimodal local solution= global solution
f14	Generalized penalized problem 1 $minf(x) = \frac{\pi}{n}\left(10\sin^2(\pi y_i) + \sum_{i=1}^{n-1}(y_i - 1)^2[1 + 10\sin^2(\pi y_{i+1})] + (y_n - 1)^2\right)$ $+ \sum_{i=1}^{n} u(x_i, 10, 100, 4)$ $y_i = 1 + \frac{1}{4}(x_i + 1), -5 \leq x_i \leq 5, x^* = (-1, -1, ..., -1), f(x^*) = 0$	n=5	Multimodal
f15	Generalized penalized problem 2 $minf(x) = 0.1\left(\sin^2(3\pi x_1) + \sum_{i=1}^{n}(x_i - 1)^2[1 + \sin^2(3\pi x_i)] + (x_n - 1)^2[1 + \sin^2(2\pi x_n)]\right)$ $+ \sum_{i=1}^{n} u(x_i, 10, 100, 4)$ $-5 \leq x_i \leq 5, x^* = (1, 1, ..., 1), f(x^*) = 0$ with $u(x, a, k, m) = \begin{cases} k * (x - a)^m & \text{if } x > a \\ -k * (x - a)^m & \text{if } x < -a \\ 0 & \text{otherwise} \end{cases}$	n=5	Multimodal
f16	Weierstrass function $w_{b,s}(x) = \sum_{i=1}^{\infty} b^{i(s-2)}\sin(b^i x)$ $b > 1, 1 \leq s \leq 2$, in our case $(1 \leq i \leq 30, s = 1.7, b = 5)$	n=1	Multimodal Continue non differentiable

References

[1] T. Bäck. Selective pressure in evolutionary algorithms: a characterization of selection mechanisms. In *Evolutionary Computation, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the First IEEE Conference on* (June 1994), pp. 57 –62 vol.1.

[2] T. Bäck. Generalized convergence models for tournament- and (mu, lambda)-selection. In *ICGA* (1995), L. J. Eshelman, Ed., Morgan Kaufmann, pp. 2–8.

[3] T. Bäck and F. Hoffmeister. Extended selection mechanisms in genetic algorithms. In *Proceedings of the Fourth International Conference on Genetic Algorithms* (1991), Morgan Kaufmann, pp. 92–99.

[4] J. Bezdek. *Pattern recognition with fuzzy objective function algorithms.* Plenum Press, New York, 1981.

[5] T. Blickle and L. Thiele. A comparison of selection schemes used in genetic algorithms. TIK-Report 11, TIK Institut fur Technische Informatik und Kommunikationsnetze, Computer Engineering and Networks Laboratory, ETH, Swiss Federal Institute of Technology, Gloriastrasse 35, 8092 Zurich, Switzerland, Dec. 1995.

[6] T. Blickle and L. Thiele. A mathematical analysis of tournament selection. In *Proceedings of the Sixth International Conference on Ge-*

netic Algorithms (San Francisco, CA, 1995), L. Eshelman, Ed., Morgan Kaufmann, pp. 9–16.

[7] A. Bouroumi, M. Limouri, and A. Essaïd. Unsupervised fuzzy learning and cluster seeking. *Intelligent Data Analysis 4* (September 2000), 241–253.

[8] K. Deb and R. Agrawal. Simulated binary crossover for continuous search space. *Complex systems 9*, 2 (1995), 115–148.

[9] K. Deb and A. Kumar. Real-coded Genetic Algorhithms with Simulated Binary Crossover: Studies on Multimodel and Multiobjective Problems. *Complex Systems 9*, 6 (1995), 431–454.

[10] A. E. Eiben, R. Hinterding, and Z. Michalewicz. Parameter control in evolutionary algorithms. *IEEE Transations on Evolutionary Computation 3*, 2 (July 1999), 124–141.

[11] V. Filipović, J. Kratica, D. Tošić, and I. Ljubić. Fine grained tournament selection for the simple plant location problem. In *Proceedings of the 5th Online World Conference on Soft Computing Methods in Industrial Applications-WSC5* (2000), pp. 152–158.

[12] D. E. Goldberg and K. Deb. A comparative analysis of selection schemes used in genetic algorithms. In *Foundations of Genetic Algorithms* (San Mateo, 1991), G. J. E. Rawlins, Ed., Morgan Kaufmann, pp. 69–93.

[13] P. Hancock. A comparison of selection mechanisms. In *Handbook of Evolutionary Computation*, T. Bäck, D. B. Fogel, and Z. Michalewicz, Eds. IOP Publishing and Oxford University Press., Bristol, UK, 1997.

[14] K. Hingee and M. Hutter. Equivalence of probabilistic tournament and polynomial ranking selection. In *Evolutionary Computation, 2008. CEC 2008. (IEEE World Congress on Computational Intelligence). IEEE Congress on* (2008), pp. 564 –571.

[15] R. Huber and T. Schell. Mixed size tournament selection. *Soft Computing - A Fusion of Foundations, Methodologies and Applications 6* (2002), 449–455. 10.1007/s00500-001-0160-8.

[16] B. Julstrom and D. Robinson. Simulating exponential normalization with weighted k-tournaments. In *Evolutionary Computation, 2000. Proceedings of the 2000 Congress on* (2000), vol. 1, pp. 227 –231 vol.1.

[17] Z. Michalewicz. *Genetic algorithms + data structures = evolution programs (3rd ed.).* Springer-Verlag, London, UK, 1996.

[18] B. L. Miller and D. E. Goldberg. Genetic algorithms, tournament selection, and the effects of noise. *Urbana 51* (Dec. 13 1995), 61801.

[19] T. Motoki. Calculating the expected loss of diversity of selection schemes. *Evolutionary Computation 10*, 4 (2002), 397–422.

[20] H. Mühlenbein and D. Schlierkamp-Voosen. Predictive models for the breeder genetic algorithm, I: Continuous parameter optimization. *Evolutionary Computation 1*, 1 (1993), 25–49.

[21] R. Poli and W. B. Langdon. Backward-chaining evolutionary algorithms. *Artificial Intelligence 170*, 11 (2006), 953 – 982.

[22] C. R. Reeves and J. E. Rowe. *Genetic Algorithms: Principles and Perspectives: A Guide to GA Theory*. Kluwer Academic Publishers, Norwell, MA, USA, 2002.

[23] A. Sokolov and D. Whitley. Unbiased tournament selection. In *Proceedings of the 2005 conference on Genetic and evolutionary computation* (New York, NY, USA, 2005), GECCO '05, ACM, pp. 1131–1138.

[24] P. Vajda, Á. E. Eiben, and W. Hordijk. Parameter control methods for selection operators in genetic algorithms. In *PPSN* (2008), vol. 5199 of *Lecture Notes in Computer Science*, Springer, pp. 620–630.

[25] H. Xie, M. Zhang, and P. Andreae. Another investigation on tournament selection: modelling and visualisation. In *GECCO '07: Proceedings of the 9th annual conference on Genetic and evolutionary computation* (London, 7-11 July 2007), vol. 2, ACM Press, pp. 1468–1475.

Received: October, 2010