A Fuzzy Clustering Technique for Adapting Tournament Selection

¹Abdelaziz El Moujahid, ¹Khalid Jebari, ^{1,2}Abdelaziz Bouroumi and ¹Aziz Ettouhami

¹LCS Laboratory, Faculty of Sciences, Mohammed V-Agdal University,
UM5A Rabat, Morocco

²LMI Laboratory, Ben M'Sik Faculty of Sciences,
Hassan II Mohammedia University (UH2M) Casablanca, Morocco

Abstract: Tournament selection is a popular selection scheme commonly used in genetic algorithms. This method depends largely on the size of the tournament which influences the quality of the final solution. In this study, a new technique is used to dynamically adjust the tournament size using the fuzzy c-means algorithm. The efficiency of the proposed technique is shown by using several benchmark test functions.

Key words: Tournament selection, tournament size, fuzzy c-means, unsupervised fuzzy clustering, genetic algorithms, evolutionary algorithms

---

INTRODUCTION

There are many selection schemes for Genetic Algorithms (GA). Several factors have contributed to the increased use of tournament selection as a selection mechanism for GA. Tournament selection is simple to code and efficient for both parallel and nonparallel architectures. In addition, tournament selection can also adjust the selection pressure to adapt to different application domains (Miller and Goldberg, 1995). Yet, the selection pressure varies with the size of the tournament, k . Hence, the importance of choosing this parameter as demonstrated by several studies (Goldberg and Deb, 1991; Muhlenbein and Voosen, 1993; Back, 1994; Blickle and Thiele, 1995a, b; Poli and Langdon, 2006).

Choosing good k values is particularly difficult in unsupervised contexts i.e., in situations where no prior knowledge is available on the nature of the problem. Large values of k leads to strong selection pressure and convergence of the GA to a local optima. Small k values on the other hand may prevent individuals with good fitness values from being candidates for the tournament selection (Poli and Langdon, 2006).

In this study, we present a new technique which allows determining k in a dynamic way. This technique is mainly based upon the Fuzzy C-Means algorithm (FCM) and uses the normalized partition entropy as a validation criterion. The value of k is the optimal number of clusters given by this unsupervised clustering procedure.

Tournament selection: The tournament selection was attributed to Wetzel in an unpublished research then it has been studied by Brindle in his doctoral thesis in 1981 (Goldberg and Deb, 1991). Afterwards, several researchers became interested in this method (Blickle and Thiele, 1995a, b; Miller and Goldberg, 1995; Filipovic *et al.*, 2000; Julstrom and Robinson, 2000; Sokolov and Whitley, 2005; Vajda *et al.*, 2008). The standard tournament selection can simply be described by the following pseudo-code:

Pseudo-code of standard tournament selection

- Randomly select k individuals from the population; $((k > 1, k$: Tournament size)
- Select the individual who has the best fitness value from selected individuals in step 1
- Repeat steps 1 and 2 N times (N is the population size)

The balance between exploitation and exploration is critical for the behavior of the GA. This balance can be adjusted by selection pressure of the selection operator (Blickle and Thiele, 1995a). Several techniques can be used in order to evaluate the selection pressure. The followings are among the mostly used in practice.

The takeover time (Goldberg and Deb, 1991) is the number of generations required for a single best individual to fill the entire population by using only the selection operator:

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

Where:

K = The tournament size

N = The population size

The loss of diversity (Blickle and Thiele, 1995a) is the proportion of individuals that are not selected during selection:

$$P_d = k^{\frac{-1}{k-1}} - k^{\frac{-k}{k-1}} \quad (2)$$

Selection intensity was used by Back (1995), Muhlenbein and Voosen (1993) and Blickle and Thiele (1995b):

$$I \approx \sqrt{2 \left(\ln(k) - \ln \left(\sqrt{4.14 * \ln(k)} \right) \right)} \quad (3)$$

Takeover time, loss of diversity and selection intensity depend on the size of the tournament hence, the importance of this parameter to adjust the selection pressure.

The difficulty lies in how to determine k. Many approaches have been proposed to determine k. Some well-known methods are mentioned here:

Fixed Tournament Size (FTS) (Goldberg and Deb, 1991):

Tournament size k is fixed according to the form $k = 2^i$, $i = 1, \dots, 5$.

Tournament selection based on weight: Julstrom and Robinson (2000) introduced a tournament selection based on weight. The probability that individual j is selected is:

$$P_j = \frac{w^{k-j} (1-w)}{1-w^k} \quad (4)$$

where, w denotes the weight of individual j:

$$k = \frac{N(1-r)}{1-r^N} \quad (5)$$

Where, r is the exponential normalization factor and N the population size.

Fine-grained tournament selection: Filipovic *et al.* (2000) 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.

Deterministic Tournament-size Control (DTC)

(Vajda *et al.*, 2008): 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 - P_2 for the first 1000 generations and afterwards binds to P_2 . They chose: p_1, p_2 in $\{2, 7, 17\}$,

$$k(t) = \begin{cases} \frac{t(p_2 - p_1)}{1000} + p_1 & t \in \{1, 2, \dots, 1000\} \\ p_2 & \text{otherwise} \end{cases} \quad (6)$$

MATERIALS AND METHODS

Proposed method: To dynamically determine k, we precede the selection process by an unsupervised fuzzy clustering procedure whose main goal is to detect the presence of homogeneous groups within the population of candidate solutions. This procedure gives us an idea about the population diversity which can help adjust the balance between exploration and exploitation by increasing or decreasing the selection pressure. For this we use the number of detected clusters as a value of k.

The fuzzy clustering procedure which offers a better way of modeling and representing overlapping clusters, consists in two main steps. The first step is an optimization procedure which involves the FCM algorithm (Bezdek, 1981) for different values of the number of clusters, c. The second step is a module for validating results using the normalized partition entropy (Bezdek, 1981) defined as follows:

$$h(U) = -\frac{1}{\log(c)} \frac{1}{N} \sum_{i=0}^n \sum_{j=0}^c \left[u_{ij} \log(u_{ij}) \right] \quad (7)$$

Where, U denotes the matrix of membership degrees. By varying the parameter c within an interval of reasonable values and repeating FCM for each c value within this interval, several partitions can be produced for each generation. Choosing the best partition and therefore the number of clusters to use is finally done through a validation phase that assesses and compares the quality of different partitions. This classification phase has two outputs:

- The number of clusters
- A prototype for each cluster

k is chosen as the number of clusters found at the clustering phase and the prototype became a member of the next generation.

We named the method Fuzzy Adapting Tournament Selection (FATS).

Pseudo-code for FATS

```

Coptimum = 2 // Coptimum: Number of clusters for a minimal entropy
hmin = 1 ; // hmin: Minimal entropy
While (c ≤ 16 && hmin > 0.0001)
{
    Apply FCM for population of individuals
    Calculate h(U) // h: entropy (Eq. 7)
    // U: fuzzy membership matrix produced by FCM
    if (hmin > h(U)) { hmin = h(U) ; Coptimum = c }
    c++
}
k = Coptimum // k: Tournament size
Trand: array of integer containing the indices of individuals in the
population
Tind_selected: an array of individuals' indices who will be selected
l = 0
For (i = 0; i < k; i++)
{
    Shuffle trand elements;
    For (j = 0; j < N; j = j+k)/N population size
    {
        I1 = trand(j);
        For (m = 1; m < k ; m++)
        {
            I2 = trand(j+m);
            if ( f(I1) < f(I2) ) I1 = I2
            // f(I1): Fitness of individual I1
        }
        Tind_selected(l) = I1
        l = l+1;
    }
}

```

Benchmark functions: To illustrate the performance of the method we applied it to the problem of minimizing several test functions that are commonly used in practice as benchmark for genetic algorithms.

Ackley function:

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

- Number of variables: $n = 5$
- Search domain: $-30 \leq x_i \leq 30$
- Multimodal
- The global minimum: $x^* = (0, \dots, 0)$, $f(x^*) = 0$

Cosinus mixture function:

$$\min f(x) = \sum_{i=1}^n x_i^2 - 0.1 \sum_{i=1}^n \cos(5\pi x_i) \quad (9)$$

- Number of variables: $n = 40$
- Search domain: $-1 \leq x_i \leq 1$
- Unimodal
- The global minimum: $x^* = (0, \dots, 0)$, $f(x^*) = 0.1 * n$

Goldstein and price function:

$$\min f(x) = [1 + (x_0 + x_1 + 1)^2 (19 - 14x_0 + 3x_0^2 - 14x_1 + 6x_0x_1 + 3x_1^2)] [30 + (2x_0 - 3x_1)^2 (18 - 32x_0 + 12x_0^2 + 48x_1 - 36x_0x_1 + 27x_1^2)] \quad (10)$$

- Number of variables: $n = 2$
- Search domain: $-2 \leq x_i \leq 2$
- Multimodal
- The global minimum: $x^* = (0, -1)$, $f(x^*) = 3$

Griewank function:

$$\min f(x) = 1 + \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) \quad (11)$$

- Number of variables: $n = 5$
- Search domain: $-600 \leq x_i \leq 600$
- Multimodal
- The global minimum: $x^* = (0, \dots, 0)$, $f(x^*) = 0$

Rastrigin function:

$$\min f(x) = 10n + \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i)] \quad (12)$$

- Number of variables: $n = 10$
- Search domain: $-5.12 \leq x_i \leq 5.12$
- Multimodal
- The global minimum: $x^* = (0, \dots, 0)$, $f(x^*) = 0$

Rosenbrock function:

$$\min f(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2] \quad (13)$$

- Number of variables: $n = 2$
- Search domain: $-30 \leq x_i \leq 30$
- Unimodal
- The global minimum: $x^* = (1, 1)$, $f(x^*) = 0$

Schwefel function:

$$\min f(x) = 418.9829n - \sum_{i=1}^n x_i \sin(\sqrt{|x_i|}) \quad (14)$$

- Number of variables: $n = 2$
- Search domain: $-500 \leq x_i \leq 500$
- Multimodal
- The global minimum: $x^* = (420.97, 420.97)$, $f(x^*) = 0$

Easom function:

$$\min f(x) = - \left[\prod_{i=1}^n \cos(x_i) * \exp \left(- \sum_{i=1}^n (x_i - \pi)^2 \right) \right] \quad (15)$$

- Number of variables: $n = 3$
- Search domain: $-100 \leq x_i \leq 100$
- Unimodal
- The global minimum: $x^* = (\pi, \pi, \pi)$, $f(x^*) = -1$

Weierstrass function:

$$w_{b,s}(x) = \sum_{i=1}^{\infty} b^{i(s-2)} \sin(b^i x) \quad (16)$$

- Number of variables: $n = 1$
- Search domain: $-100 \leq x_i \leq 100$
- Multimodal
- The global minimum: $x^* = (\pi, \pi, \pi)$, $f(x^*) = -1$
- $b > 1$, $1 \leq s \leq 2$ in the case ($1 \leq i \leq 30$, $s = 1.7$, $b = 5$)

NUMERICAL RESULTS AND DISCUSSION

We compared the technique (FATS) with:

- Fixed Tournament Selection (FTS) with k in $\{4, 8, 16\}$
- Deterministic Tournament size (DTC) (Eq. 6)

The researchers calculate for each methods (FTS, DTC and FATS) relative error and takeover time. For GA, we chose the following parameters:

- Initialization is random with a population size $N = 100$
- The coding we used is the real code
- For the crossover operator we used Simulated Binary Crossover SBX (Deb and Agrawal, 1995)
- Crossover probability: 0.97
- For mutation we considered the polynomial mutation (Deb and Kumar, 1995)
- Mutation probability: 0.03
- 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 defined as follows

$$\frac{\text{Number of optimum}}{N} * 100 < 95 \quad (17)$$

Where, N = population size. To evaluate the quality of the optimum provided by FTS, DTC and FATS, we measure the quality of the Percentage Relative Error (PRE):

$$\text{PRE} = \frac{\Delta f}{f} = \frac{|f^* - f|}{f} \quad (18)$$

The f^* in Eq. 18 is the optimum provided by the algorithm for each technique, f the actual optimum which is a priori known. Table 1 shows the results: The researchers note that FATS provides significantly better results compared to FTS and DTC.

Takeover time is used for the second comparison. As Table 2 shows takeover time for FTS and DTC are independent of the test function. The two techniques do not reflect the characteristics of the test functions that differ according to their distributions and their unimodal or multimodal nature. This is explained by the fact that the takeover time for DTC depends only on generation number and the takeover time for FTS is the same for each test function and for each generation while FATS has an acquisition time that depends on the nature of the test function.

Indeed, during the first generations, the population is diversifying, so the number of clusters is high, the takeover time is small and the selection pressure is high. Over generations if diversity decreases, takeover time increased and vice versa.

The graph in Fig. 1 shows the variation of takeover time for Ackley function, cosine mixture function and Griewank function. Selection pressure is the same whatever the test function for the FTS, it is a function of generation for the DTC whereas the FATS depends on the nature of the function.

Table 1: Percentage relative errors

Functions	Technique used			Best technique
	FTS	DTC	FATS	
Ackley	7.8	0.0	0.0	FATS, DTC
Cosinus mixture	0.0	0.0	0.0	FTS, DTC, FATS
Goldstein and price	11.0	16.0	0.0	FATS
Griewank	19.0	15.0	4.0	FATS
Rastrigin	5.0	1.0	0.0	FATS
Rosenbrock	19.0	0.0	0.0	FATS, DTC
Schwefel	13.0	5.0	0.0	FATS
Easom	5.0	6.0	1.0	FATS
Weierstrass	11.0	9.0	1.4	FATS

Table 2: Takeover time for test functions

Fonction	FTS (4)	FTS (8)	FTS (16)	DTC	FATS
Ackley	7	5	3	12	6
Cosinus mixture	7	5	4	13	10
Goldstein and price	8	5	4	11	5
Griewank	8	5	4	12	3
Rastrigin	7	5	4	12	7
Rosenbrock	7	5	4	11	6
Schwefel	7	5	4	12	8
Easom	7	5	4	12	7
Weierstrass	7	6	4	12	15

Fig. 1: Takeover time for 3 techniques

Fig. 2: Proportion of the best individual for FTS, DTC and FATS

The proportion of individuals having the best fitness value for the three functions in Fig. 2 show that the variation in FATS is according to the nature of the function. In Fig. 2a, Cosine mixture which is a unimodal function, we have a FATS net diversity compared to FTS and DTC for the first generations. After, we have a high selection pressure resulting in the deflection angle of its curve. For Weierstrass (Fig. 2b) and Goldstein-Price (Fig. 2c), Multimodal functions, FATS has a lower selection pressure compared to the FTS and the DTC.

This result corresponds to the model defined in (Back and Hoffmeister, 1991): Unimodal function require a high pressure of selection to force the search space into the gradient direction. By cons, a multimodal function requires an explorative character of the research so, a low pressure of selection is used so as not to push the GA to converge to a local optimum.

CONCLUSION

In this study, we proposed a new technique for dynamically determining the tournament size using a fuzzy clustering method. The basic idea behind this approach

consists of adjusting the selection pressure according to the population diversity in order to strike a balance between exploitation and exploration: FATS adjust appropriately the selection pressure according to the search result by dynamically using different tournament size values at different stages of the GA taking into account the nature of the problem.

In addition, the comparison between the results obtained by FATS, FTS and DTC, confirms the efficiency of the proposed technique in dealing with functions of different nature. In future study, the researchers propose other technique to dynamically adjust other parameters of genetic algorithms.

REFERENCES

- Back, T. and F. Hoffmeister, 1991. Extended selection mechanisms in genetic algorithms. Proceedings of the 4th International Conference on Genetic Algorithms and their Applications, (GAA'91), Morgan Kaufmann Publishers Inc., San Diego, CA., USA., pp: 92-99.
- Back, T., 1994. Selective pressure in evolutionary algorithms: A characterization of selection mechanisms. Proceedings of the 1st IEEE Conference on Evolutionary Computation, June 27-29, Orlando, FL., USA., pp: 57-62.
- Back, T., 1995. Generalized convergence models for tournament and (μ, λ) -selection. Proceedings of the 6th International Conference on Genetic Algorithms, July 15-19, Morgan Kaufmann Publishers Inc., San Francisco, CA., USA., pp: 2-8.
- Bezdek, J.C., 1981. Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, New York, USA., pp: 256.
- Blickle, T. and L. Thiele, 1995b. A comparison of selection schemes used in genetic algorithms. TIK Report No. 11, TIK Institute of Technology Computer Science and Communication Networks, Computer Engineering and Networks Laboratory, ETH, Swiss Federal Institute of Technology, Zurich, Switzerland.
- Blickle, T. and L. Thiele, 1995a. A mathematical analysis of tournament selection. Proceedings of the 6th International Conference on Genetic Algorithms, July 15-19, Pittsburgh, PA., USA., pp: 9-16.
- Deb, K. and A. Kumar, 1995. Real-coded genetic algorithms with simulated binary crossover: Studies on multimodel and multiobjective problems. *Complex Syst.*, 9: 431-454.
- Deb, K. and R.B. Agrawal, 1995. Simulated binary crossover for continuous search space. *Complex Syst.*, 9: 115-148.
- Goldberg, D.E. and K. Deb, 1991. A Comparative Analysis of Selection Schemes Used in Genetic Algorithms. In: Foundations of Genetic Algorithms, Rawlins, G.J.E. (Ed.). Morgan Kaufmann Publishers, Inc., San Francisco, CA., USA., ISBN-13: 978-1-558601703, pp: 69-93.
- Julstrom, B.A. and D.H. Robinson, 2000. Simulating exponential normalization with weighted k-tournaments. Proceedings of the Congress on Evolutionary Computation, July 16-19, IEEE Press, La Jolla, CA., USA., pp: 227-231.
- Miller, R.L. and D.E. Goldberg, 1995. Genetic algorithms, tournament selection and the effects of noise. *Complex Syst.*, 9: 193-212.
- Muhlenbein, H. and D.S. Voosen, 1993. Predictive models for the breeder genetic algorithm. I. Continuous parameter optimization. *Evolut. Comput.*, 1: 25-49.
- Poli, R. and W.B. Langdon, 2006. Backward-chaining evolutionary algorithms. *Artif. Intell.*, 170: 953-982.
- Sokolov, A. and D. Whitley, 2005. Unbiased tournament selection. Proceedings of the Conference on Genetic and Evolutionary Computation, June 25-29, Washington DC., USA., pp: 1131-1138.
- Vajda, P., A.E. Eiben and W. Hordijk, 2008. Parameter control methods for selection operators in genetic algorithms. *Parallel Prob. Solving Nat.*, 5199: 620-630.