FICA: fuzzy imperialist competitive algorithm

Saeid ARISH, Ali AMIRI‡, Khadije NOORI

(Department of Computer Engineering, University of Zanjan, Zanjan, Iran)
E-mail: saeed.aiPROJECT@gmail.com; a_amiri@znu.ac.ir; kh.noori90@gmail.com
Received Apr. 11, 2013; Revision accepted Dec. 6, 2013; Crosschecked Apr. 11, 2014

Abstract: Despite the success of the imperialist competitive algorithm (ICA) in solving optimization problems, it still suffers from frequently falling into local minima and low convergence speed. In this paper, a fuzzy version of this algorithm is proposed to address these issues. In contrast to the standard version of ICA, in the proposed algorithm, powerful countries are chosen as imperialists in each step; according to a fuzzy membership function, other countries become colonies of all the empires. In absorption policy, based on the fuzzy membership function, colonies move toward the resulting vector of all imperialists. In this algorithm, no empire will be eliminated; instead, during the execution of the algorithm, empires move toward one point. Other steps of the algorithm are similar to the standard ICA. In experiments, the proposed algorithm has been used to solve the real-world optimization problems presented for IEEE-CEC 2011 evolutionary algorithm competition. Results of experiments confirm the performance of the algorithm.

Key words: Optimization problem, Imperialist competitive algorithm (ICA), Fuzzy ICA.
doi: 10.1631/jzus.C1300088
Document code: A
CLC number: TP301.6

1 Introduction

A wide variety of problems in different sciences can be formulated as an optimization problem. Consequently, finding an appropriate solution for optimization problems has great importance in all sciences, especially the engineering fields.

In general, optimization problem solutions can be divided into two categories: classical and heuristic. Some instances of classical methods include analytical optimization, the Lagrangian method (Andreani et al., 2009), the Nelder-Mead method (Nelder and Mead, 1965), and linear and nonlinear optimization (Golban and Nedevschi, 2011). Most of the classical methods are based on gradient zero-crossing of the fitness function (Bertsekas and Gafni, 1983). In complex functions, these methods may fall into local minima. Also, sometimes the fitness function may not admit derivatives. These drawbacks limit the utilization of the classical methods.

Heuristic methods are used in solving common problems of different sciences. These methods attempt to discover a solution without a deep understanding of the problem structure. In other words, the problem seems like a black box for the algorithm. These methods usually act randomly using statistics gained by the samples of the search space or they are based on a model from some natural phenomena or physical processes. These methods include the genetic algorithm (GA) (Guo et al., 2010), simulated annealing (Davis, 1987), particle swarm optimization (PSO) (Kennedy and Eberhart, 1995), ant colony optimization (Brownlee, 2011), hill climbing (Brownlee, 2011), the Monte-Carlo method (Fishman, 1996), the cuckoo algorithm (Rajabioun, 2011), and so on.

ICA, one of the heuristic methods for solving optimization problems, has achieved higher convergence speed in solving problems with many independent variables, compared to PSO and GA (Gargari and Lucas, 2007; Kaveh and Talatahari, 2010). In recent years, several studies have been done to improve the performance of ICA in solving optimization problems. Kaveh and Talatahari (2010) improved ICA
by adding a random ‘perpendicular vector’ between the imperialist and the colony in the absorption step. Talatahari et al. (2012) improved the convergence speed of ICA using chaotic functions instead of random functions in the absorption step. Despite all these successful studies in improving ICA, some challenging issues still exist, including (1) falling into local optima, (2) low convergence speed in some problems, and (3) extremely high computational complexity for a large number of independent variables of the function.

To deal with the above challenges, fuzzy ICA (FICA) is proposed. Using this algorithm, each colony is considered a colony of all imperialists with a membership degree. The colonies move in the direction of the resulting vector from the weighted summation of the imperialists’ vectors, proportional to the membership function. The weights of these vectors depend on the power of the imperialists. In other words, the imperialist that has more power will absorb colonies with a higher membership degree and has a greater effect on the moving direction of the colonies. This makes all colonies move coordinately toward the current most optimal direction and increases their focus on searching in one direction as the most optimal direction (instead of focusing on several directions in ICA). Therefore, the convergence speed increases and the optimal solution is reached sooner. Since moving toward one direction increases the possibility of falling into local optima, in each time period of the fuzzy method, the colony and the imperialist among the countries are selected from the very first. Moreover, the bias of the algorithm is reduced to the initial talented points because all initial points or initial countries have been selected randomly, so the probability that their talented points are local minima (especially in complex functions with a lot of local minima) is higher than the probability that the points are selected as talented points in further steps.

In FICA, no imperialist is eliminated as time increases; all imperialists move toward an optimum imperialist and finally converge at one imperialist, which indicates the optimum solution.

The proposed algorithm is assessed on solving the real world optimization problems presented for the IEEE-CEC 2011 evolutionary algorithm competition. Experimental results confirm the performance of the algorithm.

2 ICA and its extensions

In this section, the original ICA and some of its extensions are studied. It is assumed that an arbitrary optimization function with \( n \) independent variables is given and its output represents the amount of cost. The goal is to find a point in an \( n \)-dimensional space such that by giving it to the optimization function, the minimum cost will be achieved.

2.1 Original ICA

In ICA (Gargari and Lucas, 2007), to solve the assumed optimization problem, \( N \) countries are considered. Each country is represented by a vector which indicates a spot in the \( n \)-dimensional solution space. Among these spots, those having the minimum cost according to the fitness function are considered the imperialists and the rest are colonies.

For each imperialist, first the normalized cost is calculated:

\[
\text{NOC}_n = \max_{j=1,2,\ldots,N_{\text{imp}}} (c_j) - c_n, \quad (1)
\]

where \( c_i \) is the cost of the \( i \)th imperialist and \( C_n \) is the normalized cost of the \( n \)th imperialist. After this step, colonies should be selected. The imperialist that has more power attracts more colonies. Eq. (2) describes this operation:

\[
\text{NC}_n = \frac{\text{NOC}_n}{\sum_{j=1}^{N_{\text{imp}}} \text{NOC}_j} \cdot N_{\text{col}}, \quad (2)
\]

where \( \text{NC}_n \) represents the number of colonies for the \( n \)th imperialist, \( N_{\text{imp}} \) shows the number of imperialists, and \( N_{\text{col}} \) is the number of all colonies.

Next is the absorption step, in which each imperialist tries to absorb its own colonies. Colonies move in the direction of their imperialist’s vectors with a random deviation. The following formula describes this movement:

\[
\text{col}(t + 1) = \text{col}(t) + x, \quad (3)
\]

where \( U(0, \beta d) \) is a function that generates a random variable with a uniform distribution in \([0, \beta d]\). \( V_1 \) is the vector between the imperialist and the colony, \( d \) is the distance between the imperialist and the colony,