ICAI'10

The 2010 International Conference on Artificial Intelligence

Improving Interarea and Local Mode Oscillations with UPFC and TCSC Controlled by Optimized and Adaptive Fuzzy PI Controller

Heidarali Shayanfar, Reza Navabi, Reza Etemad

Design and Implementation of A Real-Time Intelligent Controller for Endodontics Motor Driven Nitinol Rotary Drill

Sayedali Mousavi, Mohsen Sadeghi, Majid Hashemipour

Center of Slice Area Average Defuzzifier for Digital Implementations of Fuzzy Systems

Antonio Hernandez Zavala, Oscar Camacho Nieto, Ildar Batyrshin

SESSION: GENETIC AND EVOLUTIONARY ALGORITHMS

Named Entity Recognition: A Genetic Approach for Vote Based Classifier Ensemble Selection

Asif Ekbal, Sriparna Saha, Christoph Garbe

Robust Design of STATCOM Controller with PSO and GA

Hossein Shayeghi, Heidarali Shayanfar, Amin Safari

Differential Evolution Based on Intelligent Mutation Scheme

Wen-feng Yang, Xing-lin Zhang, Feng Zhang, Ze-feng Ye

Generating Efficient Test Cases Using Genetic Algorithm

Anoj Kumar, K.K. Mishra, A.K. Misra

A New Hybrid Evolutionary Algorithm Based on ICA and GA: Recursive-ICA-GA

Vahid Khorani, Farzad Razavi, Ahsan Ghoncheh

SESSION: SWARM OPTIMIZATION + ANT/BEES COLONY SYSTEMS + SIMULATED ANNEALING

Automated Map Generalization: Application of Simulated Annealing to River Symbolization

Nigel Richards, Mark Ware, Nathan Thomas, Andrew Ware

Application of a Hybrid Evolutionary Algorithm for STLF

Hossein Shayeghi, Heidarali Shayanfar, Ghaffar Azimi

The Application of Cooperative Particle Swarm Optimization with Quadratic Interpolation in Optimal Lambert Two-impulse Transfers

Song Shenmin, Jing Xin

A Modified PSO Method Enhanced with Fuzzy Inference System for Solving the Planar Graph Coloring Problem

Mostafa Erfani, Afshin Ghanizadeh, Amir Atapour Abarghouei, Saman Sinaie, Siti Mariyam Shamsuddin

A Proposed Rank Based Information Dissemination Topology for the Cooperative and Non-Cooperative PSO

Mohammad Ullah, Azzam Asar, Affan Khan, Riaz Hasnain, Mudasser Wyne

Application of SSSC for LFC of a Deregulated Power System Using Particle Swarm Optimization

Hossein Shayeghi, Gholamreza Zareiegovar, A. R. Marami Iranaq, Heidarali Shayanfar

A new hybrid evolutionary algorithm based on ICA and GA: Recursive-ICA-GA

A. Vahid Khorani[1], B. Farzad Razavi[1], C. Ahsan Ghoncheh[1]
[1]Islamic Azad University, Qazvin Branch, Iran

Abstract - *In this paper a new method is proposed based on the combination of ICA (Imperial Competitive Algorithm) and GA (Genetic Algorithm) which improves the convergence speed and accuracy of the optimization results. The new algorithm, which is named R-ICA-GA (Recursive-ICA-GA), runs ICA and GA consecutively. It is shown that a fast decrease occurs while the proposed algorithm switches from ICA to GA. The main goal of the new proposed algorithm is to achieve a faster optimization technique by applying this fast decrease. Moreover, the simple combination of ICA and GA, which is named ICA-GA, is presented in this study. These two combination schemes of ICA and GA are used for comparing with other conventional algorithms. Finally, three fitness functions are used for comparing the suggested algorithms. The obtained results show that compared with the previous method, the proposed algorithms are at least 32% faster in optimization processes; also the variance convergence speed is smaller than the ICA and GA.*

1 Introduction

Evolutionary optimization algorithms are studied in different fields. Need to find global optimum of the fitness functions which are defined in different fields such as control, communication, mechanics, economy, Artificial Neural Networks (ANN) and chemistry lead to improve the evolutionary algorithms [1-4]. Also due to usages of these algorithms to solve the continuous and discontinuous problems, the research topics in this field has increased [5]. Classical, gradient based and analytical optimization methods are repeatedly trapped in the local optimums. After these methods, the Evolutionary Computation Algorithms (ECAs) have been proposed.

With increasing the processing power of computers, the evolutionary algorithms are more considered. In 1935 the Genetic Algorithm (GA) was introduced based on the Darvin theory and natural selection by Holland and his students [6]. GAs operate on a population of potential solutions by applying the principle of the survival of the fittest to produce successively superior approximations to a solution. At each generation of the GA, a new set of approximations is reproduced by the process of selecting individuals according to the fitness and some other operators form natural genetics [7]. Also, bird flocks and fish school's behavior were studied by Reynolds and Heppner in 1987 and 1990 to define new optimization algorithms. Particle Swarm Optimization (PSO) proposed and developed by Kennedy and Elberhart in 1996 [8]. This algorithm is an evolutionary computational model which is based on swarm intelligence [7]. Esmail Atashpaz proposed the ICA algorithm inspired by the imperialistic competition in 2007 [9]. Similar to the other evolutionary algorithms that start with an initial population, ICA begins with initial empires. Any individual of an empire is called a country. There are two types of countries; colony and imperialist state that collectively form empires. Imperialistic competitions among these empires form the basis of the ICA. During this competition, weak empires collapse and powerful ones take possession of their colonies. Imperialistic competitions converge to a state in which there exists only one empire and its colonies are in the same position and have the same cost as the imperialist [1].

Some researchers tried to improve the performance of the evolutionary algorithms by modifying the previous algorithms [7, 10]. Methods which are proposed to improve the evolutionary algorithms reduce a series of disadvantages from them. Another approach is combining the previous evolutionary algorithms to improve their performance [11, 12]. One of the important characteristics for all the optimization algorithms is optimization time and algorithm speed to get to the optimized answer. Also, importance of these characteristics can be revealed by looking at the real time usages of the evolutionary optimization algorithms [13]. In addition of the convergence time and algorithms speed, variance of the convergence speed must be noticed to evaluate the performance of the evolutionary algorithms.

This paper proposes a novel hybrid approach to combine the ICA and GA. The proposed hybrid algorithm has a better convergence speed in comparison to its basic algorithms. In this paper, the comparison between the suggested algorithms and other basic algorithms is done by comparing the algorithms average convergence speed and their variance convergence speed.

Farzad Razavi is with the Islamic Azad University, Qazvin Branch, Iran. (corresponding author to provide phone: +98912 1773241; email: farzad.razavi@qazviniau.ac.ir).

2 Basic Algorithms

2.1 ICA

ICA optimizes the objective function via imperialistic competition idea. This algorithm uses the assimilation policy which the imperialistic countries have reached after the 19th century. Based on this policy the imperialists try to improve the economy, culture and political situations of their colonies. This policy makes the colony's enthusiasm toward the imperialists. In this theory, an imperialist with its colonies is called an empire. In this approach, the power of an empire depends on the power of its imperialist and its colonies. By imperialistic competitions the imperialists which are weaker lose their colonies. These colonies would join a more powerful empire for greater supports. After a while, the weaker empires will lose all their colonies and their imperialists will transform to the colonies of the other empires; at the end, all the weak empires will be collapsed and only one powerful empire will be left.

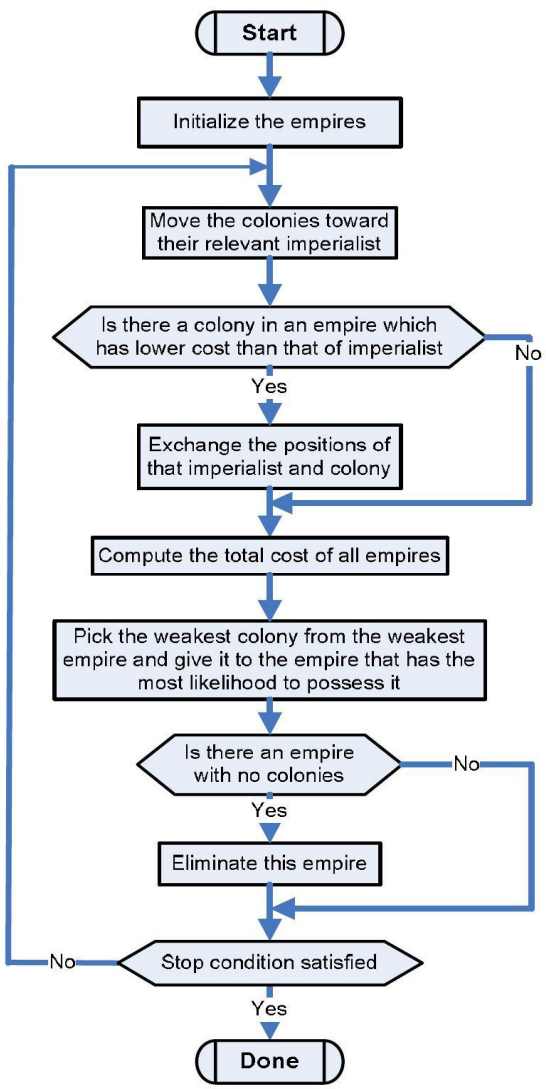


Fig. 1. Flowchart of ICA

Fig. 1 depicts the flowcharts of the ICA [9]. For coding this algorithm, an initial population is created randomly. Each member of this population is a vector of random numbers which is called a country. The cost of objective function will be calculated for each of these countries and shows their power. Countries which have more optimized costs select as the imperialists and other countries select as colonies and would divide between the imperialists. Imperialists with their relevant colonies are named empires. Cost of each empire determines its power. Equation (1) shows how the power of empires can be calculated.

$$T.C._n = Cost(imp._n) + \xi\ mean\{cost(colonies_n)\} \quad (1)$$

Where $T.C._n$ is the total cost of the nth empire and ξ is a positive number which is considered to be less than 1 and $colonies_n$ presents colonies of nth empire. The colonies move toward their imperialists, in each iteration. Fig. 2 depicts how colonies move toward their relevant imperialists. In this figure, x is a random variable with uniform distribution between 0 and $\beta \times d$. Equation (2) shows how x calculates.

$$x \sim U(0\,, \beta \times d) \quad (2)$$

Where β is a number greater than 1 and d is the distance between colony and imperialist. To search different points around the imperialist, a random amount of deviation is added to the direction of movement which is showed by θ in Fig. 2. In this figure, θ is a random number with uniform distribution between $-\gamma$ and γ. Equation (3) shows how θ calculates.

$$\theta \sim U(-\gamma\,, \gamma) \quad (3)$$

Where γ is a number that adjusts the deviation from the original direction.

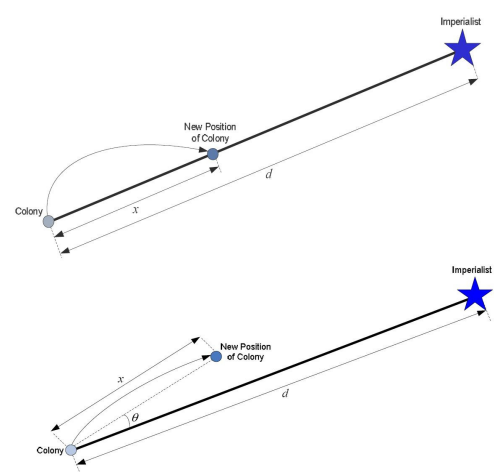
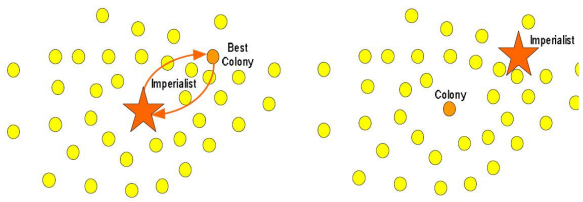


Fig. 2. Moving colonies toward their relevant imperialists

In this method, colonies will exchange their positions by their imperialists when they earn a cost more optimized in

comparison to them. Fig. 3 depicts how the positions can be exchanged.



a. Exchanging the positions b. The empire after exchange

Fig. 3. Exchanging positions of the imperialist and a colony

Next step is imperialistic competition. All empires try to take possession of colonies of the other empires and control them. So, the most powerful empire must take possession of a colony from the weakest empire. Fig. 4 depicts how it can be modeled.

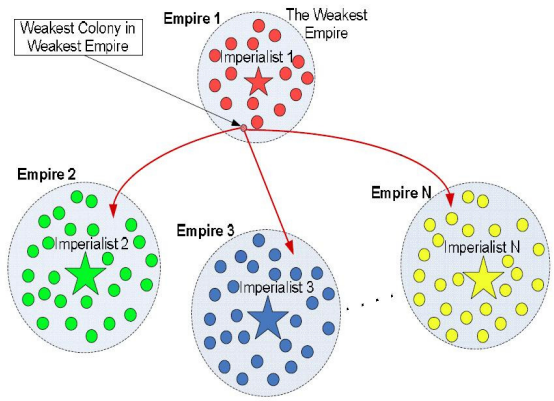


Fig. 4. Imperialistic competition. The most powerful empire will possess the weakest colony of the weakest empire.

After a few iterations, powerless empires will collapse in the imperialistic competition and their colonies will be divided among other empires. At the end, all the empires except the most powerful one will collapse and the colonies will be under the control of this unique empire. In this new world all the colonies have the same position and same costs and they are controlled by an imperialist with the same position and cost as themselves. In this condition the imperialistic competition ends and the algorithm stops. Position and cost of remained imperialist respectively show the optimized variables of the problem and optimized result of the problem [1].

2.2 GA

By noticing the Darwin theory about evolution, GA is created. Fig. 5 shows the flowchart of this algorithm [6]. For a specific problem, the GA codes a solution as an individual chromosome. It then defines an initial population of those individuals that represent a part of the solution space of the problem. The search space therefore, is defined as the solution space in which each feasible solution is represented by a distinct chromosome. Before the search starts, a set of chromosomes is randomly chosen from the search space to form the initial population. Next, through computations the individuals are selected in a competitive manner, based on their fitness as measured by a specific objective function. The genetic search operators such as selection, mutation and crossover are then applied one after another to obtain a new generation of chromosomes in which the expected quality over all the chromosomes is better than that of the previous generation. This process is repeated until the termination criterion is met, and the best chromosome of the last generation is reported as the final solution [7].

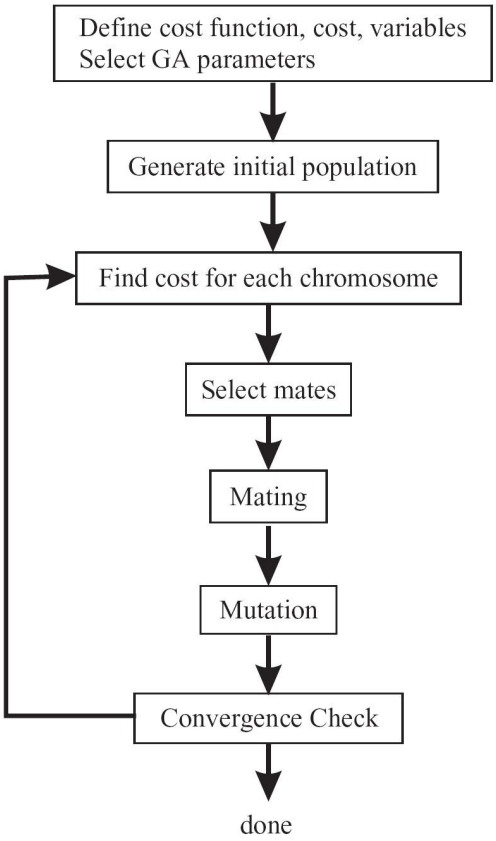


Fig. 5. Flowchart of GA

3 Problem statement

To improve the evolutionary optimization algorithms, it is needed to study on their weaknesses. So it can be surveyed as follow:

1) Simple GA doesn't have enough intelligence to recognize the most important regions of optimization area. This algorithm always searches the whole region of optimization area to find the optimized answer and doesn't have more concentration on any special region which is talent full to contain the global optimum. This weakness causes the convergence curve decreases softly. So the algorithm will optimize the problem slowly in the first generations compared to the ICA [14]. So a method for

compensating this weakness is needed.

ICA at the first iterations of optimization, instead of GA, concentrates on the talent full regions of the optimization area which have more chance to contain global optimum. But in this method other regions aren't concentrated enough. This shows that the ICA in opposite of GA will lead to a good convergence speed in the first iterations but in the final iterations it doesn't show a good performance. So a method is needed which forces the ICA to attend on all the regions uniformly. For solving this problem the ICA-GA method is proposed in 4.1.

2) By using the ICA method, in the last iterations of optimization, all of the colonies gather besides the imperialists. So by each iteration, just a few colonies have big replacements because of the invasions by the other empires. This causes that, although the imperialists are near the global optimum, the algorithm wouldn't be able to find the global optimum. So a method is needed in order to move these colonies to other regions. For solving this problem R-ICA-GA is proposed in 4.2.

4 Proposed combination methods

4.1 ICA and GA simple combination method (ICA-GA)

With intelligent combination of the two evolutionary algorithms of ICA and GA, these algorithms can be used next to each other for covering their disabilities. The suggested method is based on using the ICA in the first iterations of optimization. This leads that the ability of ICA in the first iterations of optimization be used. At the second part of optimization the developed population calculated by ICA is given to the GA. So the ability of GA in finding the global optimum without forgetting the other regions will be used in this section of algorithm. Fig. 6 shows flowchart of the algorithm.

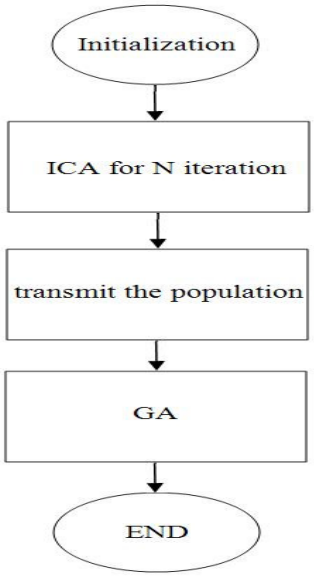


Fig. 6. Flowchart of ICA-GA

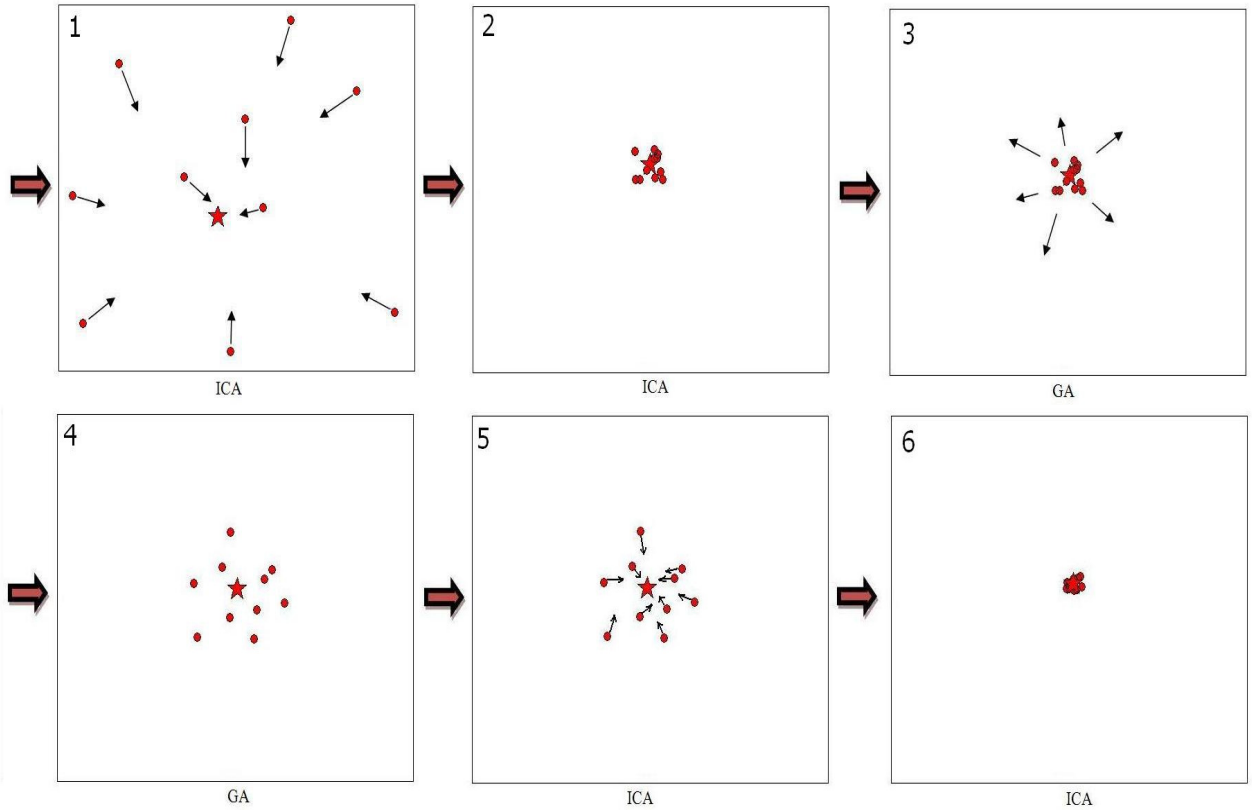


Fig. 7. Behavior of R-ICA-GA to getting close to the global optimum

4.2 ICA and GA recursive combination method (R-ICA-GA)

This algorithm proposes to combine two basic algorithms based on recursive repetition of them. GA always prevents increasing the concentration on talent full regions of optimization area. On the other hand, ICA always sends the colonies toward the imperialists and gathers a big percentage of population near the imperialists. The global optimum usually is near the local optimums in most of the problems. So the populations which are gathered near to the imperialists of ICA (which may be near the global optimum), will expand and twitch, by recursive combination of these two algorithms. In fact, the suggested algorithm behaves based on ICA, but it has used the help of GA to expand the population of colonies which are gathered near the imperialists. Fig. 7 shows the behavior of colonies in the proposed algorithm.

As shown in Fig. 7 the proposed algorithm, uses ICA to move the population toward the imperialists, with the difference that the population movement is done in a billowy way. This billowy movement leads to a fast decrease in the convergence curve in the times which algorithm switches from ICA to GA and vice versa. This fast decrease shows the more optimized regions next to the imperialists are found. Fig. 8 shows the flowchart of the R-ICA-GA algorithm. N_1, N_2 and N_3 depend on the problem can be chosen. For example N_1 is chosen 50 for the benchmark G_3(given in the appendix) in section 5; also N_2 and N_3 are chosen 10.

5 Experimental studies

In this part, the proposed algorithms are tested with three bench mark problems. two benchmarks of G1 and G2 (given in the appendix) are the simple optimization problems [9] and G3 (also given in the appendix) is a difficult optimization problem. The two first benchmarks are given for comparing the different algorithms' performance. The third benchmark is optimized for showing the power of the proposed algorithm in complicated problems. Four algorithms of GA, ICA, ICA-GA and R-ICA-GA are used for testing and comparing. In the two algorithms ICA-GA and R-ICA-GA, the exact code is used which has been written for two algorithms of GA and ICA. This shows that the new combination has been improved in comparison to the basic optimization algorithms and if the basic algorithms' codes are written more precisely in order to reach to the answer faster, the performance of the proposed hybrid algorithms will be better too. Results of the tests are studied in two categories. First a short study on the result of applying proposed algorithms on the simple optimization problems is done and after that analyzing the power of these algorithms in difficult and complicated problems is given.

1) Fig. 9 and Fig. 10 show the test results of the GA, ICA, ICA-GA and R-ICA-GA on the two benchmarks of G1 and G2. These results are obtained after 20 times running of each algorithm and choosing the closest sample to the average of convergence speed of these 20 runs. It is depicted in the figures that the ICA-GA algorithm behaves like the ICA in the beginning of optimization and it behaves like GA in the second stage of optimization. The R-ICA-GA algorithm has a very good performance in comparison to the other three algorithms.

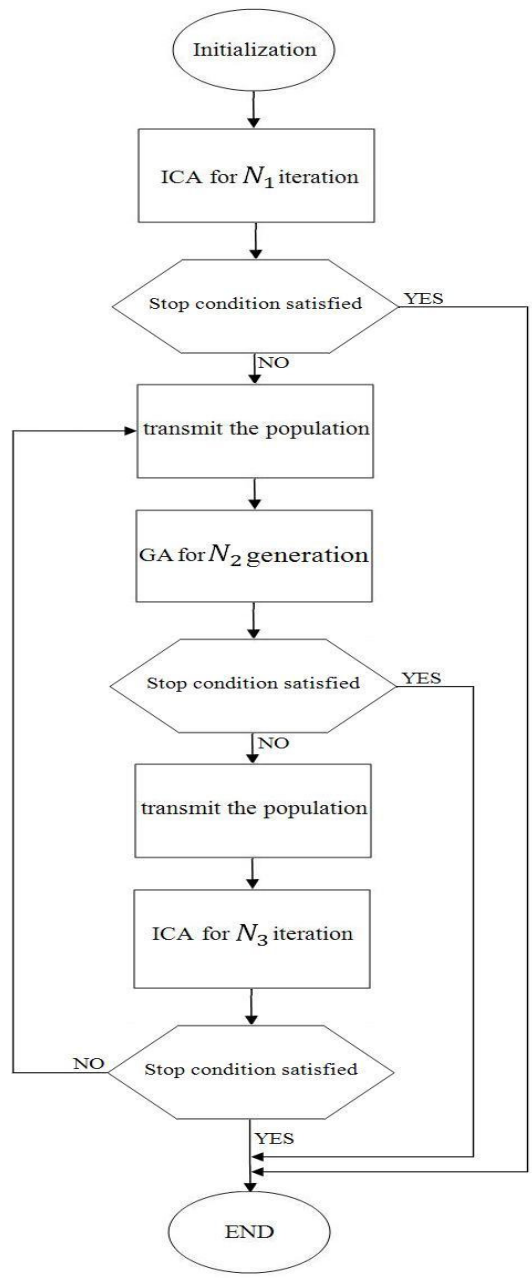


Fig. 8. Flowchart of R-ICA-GA

2) TABLE I shows the test results of the GA, ICA, ICA-GA and R-ICA-GA algorithms on the G3. Each algorithm has been run 20 times and the average convergence speed and its variance is seen in this table. As seen in TABLE I the

average convergence speed of ICA-GA is improved 46% in comparison to the GA and 32% in comparison to ICA. The average convergence speed of R-ICA-GA is improved 90% in comparison to GA and 88% in comparison to ICA. It is seen that the R-ICA-GA algorithm is more powerful in comparing to the other algorithms in the aspect of convergence speed and also it has a good convergence speed variance too.

Fig. 11 shows the convergence speed curve of the GA, ICA, ICA-GA and R-ICA-GA algorithms on G3 beside each other. In this sample the time of switching from ICA to GA in ICA-GA (*N* from Fig. 6) is selected 150 iterations. By zooming on the switching region, a fast curve decrease will be seen after switching from ICA to GA in Fig. 11 . The reason is described in 4.2. With using this fast decreasing continuously it will be seen that the R-ICA-GA algorithm has reached to the optimum answer with a much less iterations in comparison to the tree other algorithms. The first switching time from ICA to GA in the R-ICA-GA is chose to 50 iterations and after that each algorithm runs for 10 iterations recursively. This process continues until the global optimum is found.

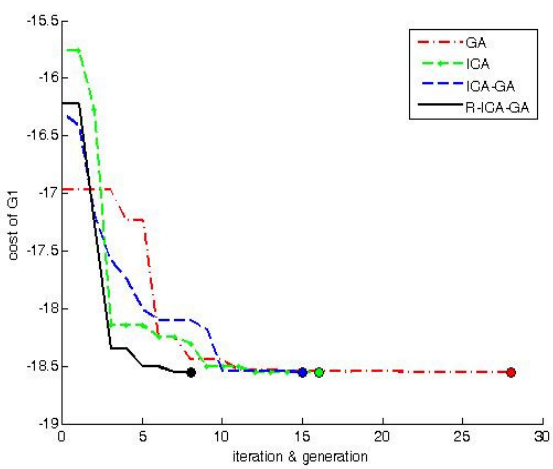


Fig. 9. Optimization results on G1

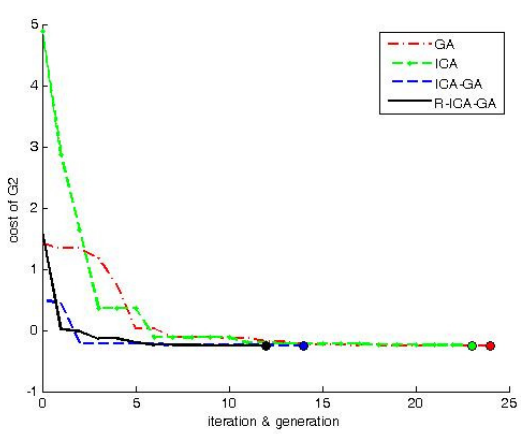


Fig. 10 Optimization results on G2

TABLE I
TEST RESULTS

Method	Mean Convergence Speed (iterations & generations)	Convergence Variance (iterations & generations)
GA	1806	270
ICA	1447	651
ICA-GA	975	390
R-ICA-GA	164	49

Results of 40 times running optimization algorithms on benchmark of G3

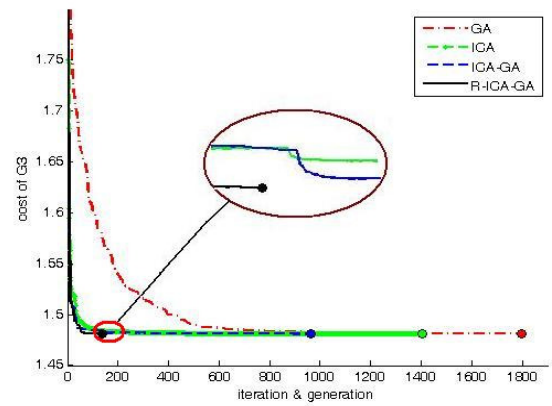


Fig. 11. Optimization results on G3

6 Conclusion

In this paper an evolutionary algorithm was proposed based on ICA and GA. The proposed algorithm was performed by repeating ICA and GA recursively and was named R-ICA-GA. It was presented that the proposed algorithm has a better performance, in simple and complicated problems, compared with its basic algorithms. Also a hybrid algorithm is made from the simple combination of ICA and GA which is named ICA-GA. This paper showed that a fast decrease in the convergence speed curve will be occurred while switching from ICA to GA. This fast decrease was used for making a powerful hybrid algorithm. Also the GA, ICA, ICA-GA and R-ICA-GA were compared with each other on three optimization problems.

7 Appendix

Problem G_1

$$f = x.\sin(4x) + 1.1y.\sin(2y)$$

$$0 < x, y < 10$$

$$minimum : f(9.039, 8.668) = -18.5547$$

Problem G_2

$$f = (x^2 + y^2)^{0.25} \times \sin\{30[(x + 0.5)^2 + y^2]^{0.1}\} + |x| +$$

$$-\infty < x, y < +\infty$$

$$minimum : f(-0.202, 0) = -0.2474$$

Problem G_3

$$\begin{aligned} f = {} & [x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 + x_6^2 + x_7^2 + x_8^2 + x_9^2 + x_{10}^2] \\ & + [(2.813x_9 - 3.63x_8 - 0.4)^2 + (3.24x_7 - 3.63x_8 - 0.4)^2 \\ & + (2.979x_7 - 3.489x_2 - 0.4)^2 + (3.187x_1 - 3.489x_2 - 0.4)^2 \\ & + (4.62x_2 - 4.11x_3 - 0.4)^2 + (5.242x_3 - 4.47x_4 - 0.4)^2 \\ & + (6.96x_4 - 6.13x_5 - 0.4)^2 + (2.812x_5 - 3.633x_6 - 0.4)^2 \\ & + (5.422x_6 - 4.735x_1 - 0.4)^2 + (6.531x_{10} - 5.8x_9 - 0.4)^2 \\ & + (3.0145x_5 - 3.513x_7 - 0.4)^2] \end{aligned}$$

$$0.1 \le x_i < 1 \quad for \quad i = 1, 2, \ldots, 10$$

$$\begin{aligned} minimum : {} & f(0.1173, 0.1061, 0.1506, 0.1861, 0.1966, 0.1316, 0.1664, 0.1, 0.216, 0.2473) \\ & = 1.4812 \end{aligned}$$

8 Acknowledgment

Authors gratefully acknowledge Mr. Esmaeil Atashpaz, who has invented ICA algorithm and presented the related useful documents on his website. His support and help is truly appreciated.

9 References

[1] Atashpaz Gargari, E., et al., *A novel approach for PID controller design in MIMO distillation column process*, in *Proceeding of IEEE CEC 2008, within IEEE WCCI 2008*. 2008: Hong Kong. p. 1929-1934.

[2] Mitilineos, S.A., K.S. Mougiakos, and S.C.A. Thomopoulos, *Design and Optimization of ESPAR Antennas via Impedance Measurements and a Genetic Algorithm.* Antennas and Propagation Magazine, IEEE, 2009. 51(2): p. 188-123.

[3] Ling, X., et al., *Estimation Mechanics Parameters of Rock Mass Based on Improved Genetic Algorithm*, in *2009 the 2nd IEEE International Conference on Computer Science and Information Technology (ICCSIT 2009)*. 2009.

[4] Wimalendra, R.S., et al., *Determination of Maximum Possible Fuel Economy of HEV for Known Drive Cycle: Genetic Algorithm Based Approach* in *Information and Automation for Sustainability, 2008. ICIAFS 2008. 4th International Conference on*. 2008. p. 289-294.

[5] Razavi, F., et al., *A new comprehensive genetic algorithm method for optimal overcurrent relays coordination.* Electric Power Systems Research, 2008. 78(4): p. 713-720.

[6] Haupt, R.L. and S.E. Haupt, *Practical Genetic Algorithms*. Second Edition ed. 2004, New Jersey: John Wiley & Sons.

[7] Shi, X.H., et al., *An improved GA and a novel PSO-GA-based hybrid algorithm.* Information Processing Letters, 2005. 93(5): p. 255-261.

[8] PARSOPOULOS, K.E. and M.N. VRAHATIS, *Recent approaches to global optimization problems through Particle Swarm Optimization.* Natural Computing, 2002. 1(2-3): p. 235-306.

[9] Atashpaz Gargari, E. and L. Caro, *Imperialist Competitve Algorithm: An Algorithm for Optimization Inspired by Imperialistic Competition*, in *2007 IEEE Congress on Evolutionary Computation (CEC 2007)*. 2007.

[10] Zhang, Q.-L., X. Li, and Q.-A. Tran, *A Modified Particle Swarm Optimization Algorithm*, in *Machine Learning and Cybernetics, 2005. Proceedings of 2005 International Conference on*. 2005: Shanghai, China p. 2993 - 2995.

[11] Shi, X.H., et al., *Hybrid evolutionary algorithms based on PSO and GA*, in *Evolutionary Computation*. 2003. p. 2393-2399.

[12] Ai-ling, C., Y. Gen-ke, and W. Zhi-ming, *Hybrid discrete particle swarm optimization algorithm for capacitated vehicle routing problem* Journal of Zhejiang University - Science A, 2006. 7: p. 607-614.

[13] RAUCH, W. and P. HARREMOES, *GENETIC ALGORITHM S I N REAL TIME CONTROL APPLIED TO MINIMIZE TRANSIENT POLLUTION FROM URBAN WASTEWATER SYSTEMS.* Water Research 1999. 33(5): p. 1265-1277

[14] Liu, Y., L. Ma, and J. Zhang, *Reactive power optimization by GA/SA/TS combined algorithms* International Journal of Electrical Power & Energy Systems, 2002. 24(9): p. 765-769.