A Novel Binary Particle Swarm Optimization Method Using Artificial Immune System

Farzaneh Afshinmanesh, Alireza Marandi, Ashkan Rahimi-Kian , *Member, IEEE*

Abstract **Particle Swarm Optimization, a nature-inspired evolutionary algorithm, has been successful in solving a wide range of real-value optimization problems. However, little attempts have been made to extend it to discrete problems. In this paper, a new binary particle swarm optimization method based on the theory of immunity in biology is proposed. In spite of the simplicity of the technique, simulation results show the improvement of the searching ability and increment in the convergence speed in comparison with the other binary PSO and genetic algorithm.**

Keywords — **Discrete Particle Swarm Optimization, Artificial Immune System, Negative Selection.**

I. Introduction

PARTICLE Swarm Optimization was originally developed by Eberhart and Kennedy in 1995 [2]. This evolutionary computation technique, based on the movement and intelligence of swarms, has been shown to be effective in optimizing difficult multidimensional problems in a variety of fields [11]. Additionally, it has been demonstrated in certain instances that PSO outperforms other methods of optimization like genetic algorithm [5]. This method comprises a very simple concept and paradigms can be implemented more easily with it.

In PSO, each individual (*particle*) traces a trajectory in the search space, constantly updating a *velocity* vector based on the best solutions found so far by that particle as well as others in the population (*swarm*) [12].

PSO has been introduced as an optimization technique in real-number spaces, where the trajectories are defined as changes in position on some dimensions. But in discrete PSO the particles operate on discrete search space, where the trajectories are defined as changes in the probability that a coordinate will take on a value from feasible discrete values [1].

Artificial immune systems propose a novel computational intelligence paradigm inspired by the immune system. Like neural networks and evolutionary algorithms, AIS are highly abstract models of their biological counterparts applied to solve problems in different domains. AIS have also been used in conjunction with other soft computing paradigms in order to create more powerful models and improve individual performances, supporting the claim that they compose a new and very useful soft computing approach [6].

This paper introduces a new binary PSO optimization algorithm. The AIS-based Particle Swarm Optimization is proposed after combining PSO with negative selection [15]. The results of computer simulations, optimizing several test functions, indicate the effectiveness of the proposed algorithm due to its better searching ability and convergence speed compared to other binary PSO and genetic algorithm.

II. Particle Swarm Optimization

The PSO algorithm is initialized with a population of individuals placed on the search space randomly and searching for optimal solution by updating individual generations. In each iteration the velocity and the position of each particle is updated according to its previous best position ($p_{best,n}$) and the best position of all particles ($g_{best,n}$). Each particle's velocity and position is adjusted by the following formulas [2, 3]:

$$V_n(t+1) = \omega \times V_n(t) + c_1 \times \varphi_1 \times \left(p_{best,n} \quad X_n(t)\right) + c_2 \times \varphi_2 \times \left(g_{best,n} \quad X_n(t)\right) \quad (1)$$

$$X_n(t+1) = X_n(t) + V_n(t+1) \quad (2)$$

n denotes the n-th particle in the swarm, t represents the iteration number, $V_n(t)$ is the velocity vector of the n-th particle, and $X_n(t)$ is the position vector. c_1, c_2 are the acceleration constants which are positive numbers, φ_1 and φ_2 are random numbers uniformly distributed between 0 and 1, and ω is called the inertia weight [5, 13, 14]. In addition to c_1 and c_2, implementation of the PSO algorithm also requires placing limit on the velocities ($V_{\max}$). If the velocity on one dimension exceeds the maximum, it will be set to $V_{\max}$. This parameter controls the convergence rate and can prevent the method from exploding [13, 14].

III. Binary PSO

It follows from the physical analogy of the swarm that it is best suited to handle real valued optimization problems. The technique tends to fall apart if a particle is flying between zero and one, and can only take on integer values. A clever technique for creating a discrete binary version of the PSO introduced by Kennedy and Eberhart [1] in 1997 uses the concept of velocity as a probability that a bit takes on a one or a zero.

A drawback of [1] is that the distance (x_{id}) update equation has a non-standard form and more difficult to implement than the standard one (e.g. Eq. (1)). In [1], By com-

Farzaneh Afshinmanesh, Alireza Marandi and Ashkan Rahimi-Kian are with the Control and Intelligent Processing Center of Excellence, School of ECE, University of Tehran, Tehran, Iran. (email: {f.afshinmanesh, a.marandi}@ece.ut.ac.ir, arkian@ut.ac.ir)

1-4244-0049-X/05/$20.00 ©2005 IEEE

paring the new velocity, v_{id}, with a random number the new value for x_{id} becomes 0 or 1. The other problem is the non-monotonic shape of the changing probability function, $P(\Delta)$, of a bit (from 0 to 1 or vice versa). $P(\Delta)$ in [1] has a concave shape that for some bigger v_{id} values the changing probability will decrease. This seems to be an unusual $P(\Delta)$, because we expect a higher changing probability as the velocity increases. Moreover, the sigmoid function used to limit the v_{id} values between 0 and 1, makes the problem non-linear.

IV. Proposed Binary PSO

An individual in the proposed method is a bit string which starts its trip from a random point in the search space and tries to become nearer to the global best position and previous best position of itself.

The process of generating a new position for a selected individual in the swarm is also depicted in figure 2. Differences between particle position and two desired positions (globally best position and particle previous best position) can be presented by two arrays of m bits, in which, each bits shows that this bit is different from the desired one or not, if yes that bit is one, else it is zero (Hamming distance). This comparison between two arrays is like an "xor" operator which is shown with "$\oplus$" in the following equations. After the calculation of difference vectors we should first add an exploration ability to our system, this ability is added by generating two different random vectors for two difference vectors and doing "and" processes which is shown by "$\otimes$" in the equations. These random binary arrays perform the random numbers task in real-valued PSO. The velocity vector that shows which bits should be changed is produced by applying "or" operator between two arrays produced in the previous step; then by applying "xor" between velocity vector and selected particle position vector, new position will be computed.

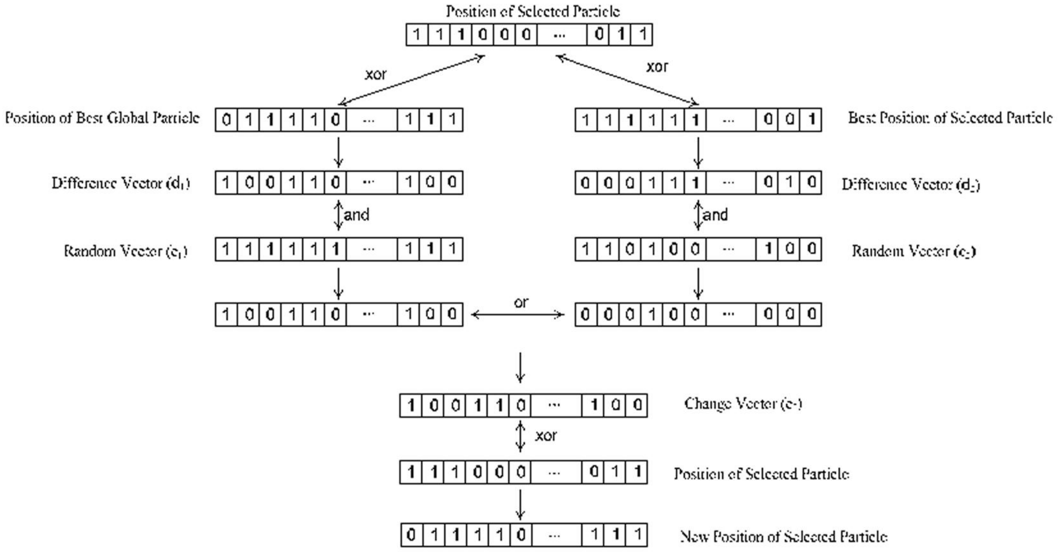


Fig. 1. New individual generation in proposed Binary PSO.

This procedure can also be represented by the following equations:

$$d_{1,n}(t) = p_{best,n} \oplus X_n(t) \tag{3}$$

$$d_{2,n}(t) = g_{best,n} \oplus X_n(t) \tag{4}$$

$$c_1 = rand(1,m), c_2 = rand(1,m) \tag{5,6}$$

$$V_n(t+1) = c_1 \otimes d_{2,n}(t) + c_2 \otimes d_{2,n}(t) \tag{7}$$

$$X_n(t+1) = X_n(t) \oplus V_n(t+1) \tag{8}$$

As it is clear, the proposed binary PSO has all major characteristics of PSO. Only the neighborhood in this method contains all particles, inertia weight is zero and the velocity limit has not been applied.

- Using Artificial Immune System

For adding one of the most important parameters of PSO ($V_{\max}$) to our technique, we used the theory of negative selection. The negative selection is drawn from this question: "How does the immune system behave when it is confronted with a self antigen?" The answer to this question is rather complex, controversial and involves different mechanisms for B-cells and T-cells. Due to the focus of this paper, the discussion will be restricted to the *thymic negative selection of T-cells* which is illustrated in figure 2. This is the process responsible for eliminating all T-cells whose receptors recognize and bind with self antigens presented in the thymus [14]; the immune system with its stochastic cell production produces more cells to be able to get out of thymus.

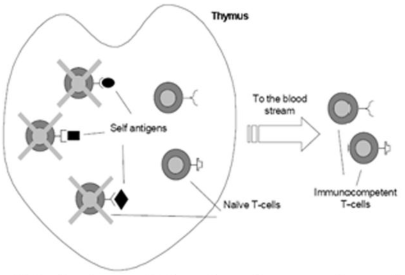


Fig. 2. Simplified view of the thymic negative selection. The naïve T-cells that recognize self antigens are purged from the repertoire. In contrast, those that do not recognize any self anti-

gen become immunocompetent cells and are released to the blood stream.

In this problem the antigens are velocity vectors $(V_n(t))$ which are produced with algorithm introduced in previous section. For indicating self and non-self antigens, the following equations are used:

$$L(V_n(t)) = \sum_{i-1}^{m} v_i(t) \qquad (9)$$

$$\begin{cases} if\ L(V_n(t)) \leq V_{\max} \rightarrow V_{\max} is\ non\ \ self \\ f\ L(V_n(t)) > V_{\max} \rightarrow V_{\max} is\ self \end{cases} \qquad (10,11)$$

As it is clear the self antigens are not allowed to be used in next steps and as it is illustrated in figure 3, for a self antigen (bad velocity vector) a manipulation should be done like the stochastic production of cells. The manipulation is a random process to reduce the length of the velocity vector ($L(V_n(t))$). The pseudo code for this process is as bellow:

```
1- C– number of 1s in V
2- If C>Vmax
3-        I–int(randome(1,C)
4-        Clear Ith 1 in V
5-        Goto 1
6- end
```

The schematic diagram of the whole method is depicted in figure 3. As it is discussed the proposed method has some benefits over other algorithms for binary optimization problems. One of the considerable features of this method, owing to the PSO, is its simplicity [2, 13, 14]. This method has utilized significant properties of PSO in binary space. Regarding its velocity definition, it is preferable to other proposed methods [1, 9, 10, 11, 12]. Moreover, by adding $V_{\max}$, it is possible to control the convergence speed and stability [13, 14].

V. Simulation Results

To show the efficiency of the algorithm, proposed binary PSO is tested on two experiment settings. First, the method is applied to two test functions (first and second De Jong's functions [9]) which were also used by Kennedy [1] and the results are compared with the first binary PSO in [1]. In this part the test functions' codes are provided through internet by William Spears[2]. (The functions are used after multiplication by -1, because the algorithm was designed for minimum optimizer.)

The result of De Jong's first function tested by the method is depicted in figure 4. As it is shown, the global optimum is achieved after 12 generations. Comparison between results obtained by proposed method with results of Kennedy [1] shows that at first the convergence rate of this method is significantly higher; also Kennedy's method has not reached the global optimum and it is claimed, in his paper, because of binary encoding, but with the same coding the global answer for the De Jong's first function is:

100000000010000000001000000000

And the fitness is 78.6. The coding [9] made the problem a hard optimization problem.

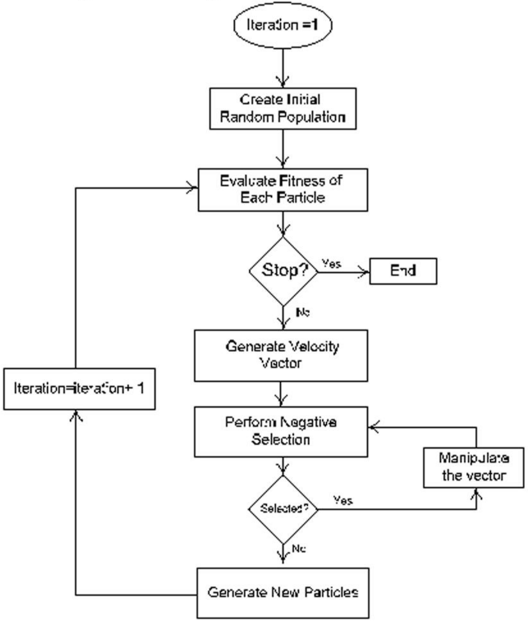


Fig. 3. Binary PSO process, using vaccination.

The second De Jong's test function with the same binary coding is also used [9]. This function as mentioned in [1] is a very hard problem for optimization; again the Kennedy method was not able to reach the global optimum point and was trapped in bad shaped local optimums [1].

The hardness of the function might be explained by the existence of strict local optima in regions which are distant from the best known optimum. For instance, the local optimum:

010111111101111000000111

returns a value of 3905.929688, while the vector:

110111101001110111011111

returns 3905.929443, and

111000011001011001000001

returns 3905.924561. And the global optima is

101111101000101111101000

This returns 3905.93, which is reached after few iterations as depicted in figure 5.

The second part of the simulation, as shown in figure 6, shows the setting result of the proposed algorithm (PSO2) in comparison with the traditional binary PSO [1] (PSO1) and Genetic Algorithm, applied to 4 famous test functions: De Jong's function, Rastrigin's function, Griewangk's function, and Sum of different Powers function [16]. For creating a hard problem, the population and dimension for these functions are set to 30, and the results are the best answers among 10 tries with each algorithm.

VI. Conclusion

A novel binary PSO based on negative selection is presented. The introduced technique is compared with other proposed binary PSO methods and conceptual similarity to conventional PSO is discussed. The benchmark functions have been used for testing the efficiency of the technique. For comparison, simulations are conducted for other binary PSO, GA and proposed PSO. The results show the better convergence rate and local optimum elimination.

A simple and high performance binary PSO representation is a way to *Particle Swarm Programming* which is the

[2] C function code provided by William Spears at http://www.aic.nrl.navy.mil/~spears/functs.dejong.html

future research of the authors. Also using concepts of PSO in AIS can be a potential research direction.

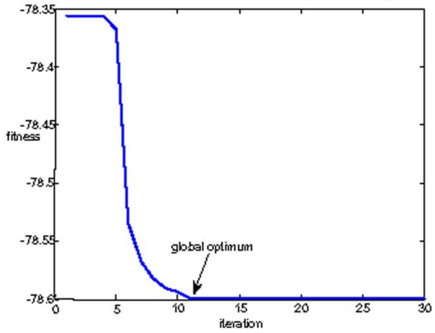


Fig. 4. Proposed Method Applied to De Jong's function 1

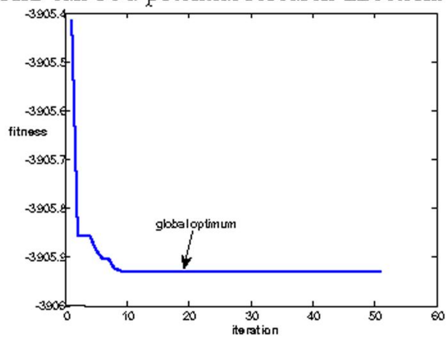
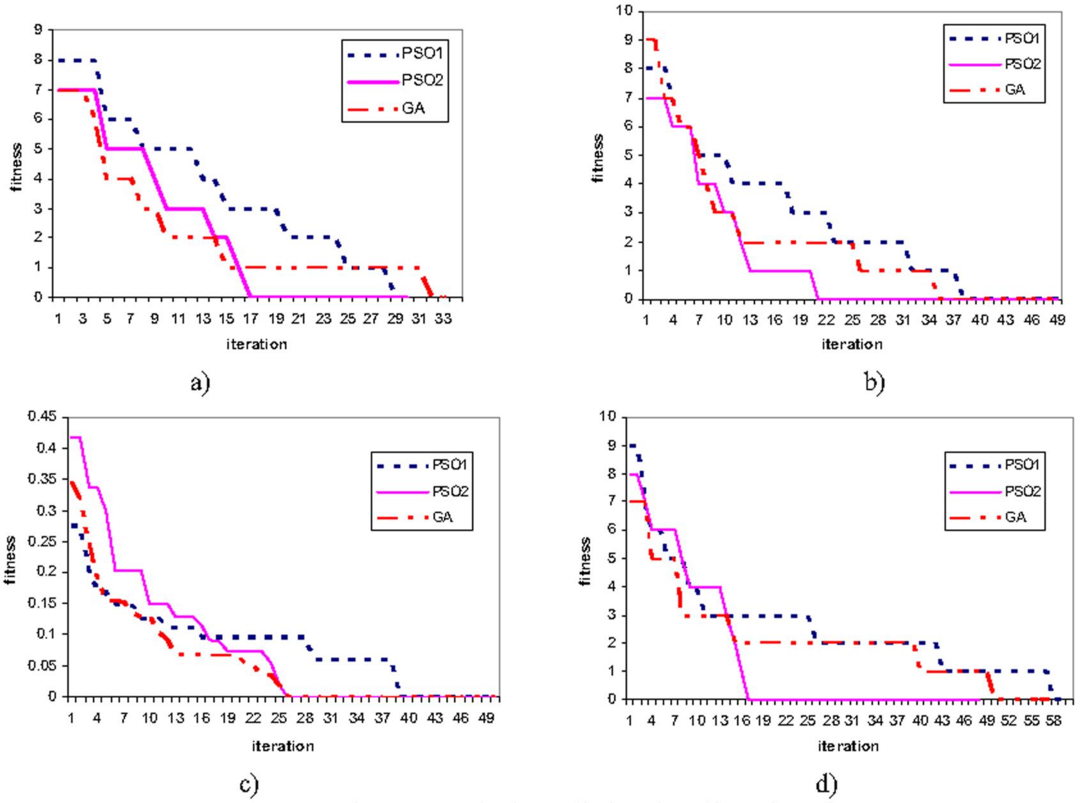


Fig. 5. Proposed Method Applied to De Jong's function 2



a)

b)

c)

d)

Fig.6. Different methods applied to a)f1, b)f2, c)f3, d)f4

REFERENCES

[1] J. Kennedy, R. Eberhurt, "A discrete binary version of the particle swarm algorithm," *in Proc. 1997 Conf. Systems, Man, Cybernetics, Piscataway, NJ, 1997.*

[2] J. Kennedy, R. Eberhurt, "Particle Swarm Optimization," *in Proc. IEEE International Conf. on Neural Networks, vol. 4, Nov. Dec. 1997.*

[3] J. Kennedy, "The Particle Swarm: Social Adaption of Knowledge," *in Proc. IEEE International Conf. on Evolutionary Computation, April 1997.*

[4] Dasgupta, D., Attoh-Okine, N.,"Immunity-based systems: a survey", *in Proc. IEEE International Conf. on Systems, Man, and Cybernetics, Orlando, Florida, Oct. 1997.*

[5] J. Kennedy and W. M. Spears, "Matching algorithms to problems: an experimental test of the particle swarm and some genetic algorithms on multi modal problem generator," *in Proc. IEEE Int. Conf. Evolutionary Computation, 1998.*

[6] de Castro, L. N. & Timmis, J. I., "Artificial Immune System as a Novel Soft Computing Paradigm", *soft computing journal 2003, 7(7), July.*

[7] De Jong, K. A. (1975). "An analysis of the behavior of a class of genetic adaptive systems.",*Doctoral dissertation, University of Michigan.*

[8] Xi-Huai Wang; Jun-Jun Li; "Hybrid particle swarm optimization with simulated annealing", *in Proc. of 2004 International Conference on Machine Learning and Cybernetics, 26-29 Aug. 2004 vol.4*

[9] Schoofs, L.; Naudts, B.; "Swarm Intelligence on the Binary Constraint Satisfaction problem", *in Proc. of the 2002 Congress on Evolutionary Computation, Volume 2, 12-17 May 2002*

[10] Wei Pang; Kang-ping Wang; Chun-guang Zhou; Long-jiang Dong; "Fuzzy Discrete Particle Swarm Optimization for Solving Traveling Salesman Problem" *in Proc. The Fourth International Conference on Computer and Information Technology, 14-16 Sept. 2004*

[11] R. C. Eberhart and Y. Shi, "Evolving artificial neural networks," *in Proc. Int. Conf. Neural Networks and Brain*, Beijing, P.R.C., 1998.

[12] Al-kazemi, B.; Mohan, C.K.; "Training feedforward neural networks using multi-phase particle swarm optimization",*in Proc. of the 9th International Conference on Neural Information Processing, 18-22 Nov. 2002 pp:2615 - 2619 vol.5.*

[13] Yong-Ling Zheng; Long-Hua Ma; Li-Yan Zhang; Ji-Xin Qian; "*On the convergence analysis and parameter selection in particle swarm optimization",in Proc. Of International Conference on Machine Learning and Cybernetics, 2-5 Nov. 2003 pp:1802 - 1807 Vol.3*

[14] Clerc, M.; Kennedy, J.; "The particle swarm - explosion, stability, and convergence in a multidimensional complex space" *IEEE Transactions on Evolutionary Computation, Volume 6, Issue 1, Feb. 2002 Page(s):58 73*

[15] Nossal, G. J. V., "Negative Selection of Lymphocytes", *cell 1994, 76, pp:229-239*

[16] H. Pohlheim, "Genetic Algorithm Toolbox Test Functions", *Department of Automatic Control and Systems Engineering University of Sheffield.*