2013 IEEE International Conference on Systems, Man, and Cybernetics

Cohort Intelligence: A Self Supervised Learning Behavior

Anand J. Kulkarni
Optimization and Agent Technology (OAT) Research Lab
Maharashtra Institute of Technology, 124 Paud Road
Pune 411038, India
ajkulkarni@oatresearch.org; kulk0003@ntu.edu.sg

Ishan P. Durugkar, Mrinal Kumar
Optimization and Agent Technology (OAT) Research Lab
Maharashtra Institute of Technology, 124 Paud Road
Pune 411038, India
ipdurugkar@oatresearch.org; mrkumar@oatresearch.org

***Abstract*—By virtue of the collective and interdependent behavior of its candidates, a swarm organizes itself to achieve a particular task. Similarly, inspired from the natural and social tendency of learning from one another, a novel concept of Cohort Intelligence (CI) is presented. The learning refers to a cohort candidate's effort to self supervise its behavior and further adapt to the behavior of other candidate which it intends to follow. This makes every candidate to improve/evolve its own and eventually the entire cohort behavior. The approach is validated by solving four test problems. The advantages and limitations are also discussed.**

Keywords- Cohort Intelligence, Self Supervised Learning, Nature-inspired Optimization

I. INTRODUCTION

In past few years a number of nature-/bio-inspired optimization techniques such as Evolutionary Algorithms (EAs), Swarm Intelligence (SI), etc. have been developed. The EA such as Genetic Algorithm (GA) works on the principle of Darwinian theory of survival of the fittest individual in the population. The population is evolved using the operators such as selection, crossover, mutation, etc. GA can often reach very close to the global optimal solution and necessitates local improvement techniques to incorporate into it [1, 2]. Similar to GA, the approach of Differential Evolution (DE) [3] is mutation driven which helps explore and further locally exploit the solution space to reach the global optimum. Although, easy to implement, there are several problem dependent parameters required to be tuned and may also require several associated trials to be performed. In addition, a distributed optimization technique inspired from societal or organizational behavior referred to as Probability Collectives (PC) decomposes entire system into subsystems and further optimizes them locally to achieve the global optimum. This approach requires several associated parameters to be tuned [4-6].

Inspired from social behavior of living organisms such as insects, fishes, etc. which can communicate with one another either directly or indirectly the paradigm of SI is a decentralized self organizing optimization approach. These algorithms work on the cooperating behavior of the organisms rather than competition amongst them. In SI, every individual evolves itself by sharing the information from others in the society. The techniques such as Particle Swarm Optimization (PSO) is inspired from the social behavior of bird flocking and school of fish searching for food [7]. The Ant Colony Optimization (ACO) works on the ants' social behavior of foraging food following a shortest path [8]. Similar to ACO, the Bee Algorithm (BA) also works on the social behavior of honey bees finding the food; however, the bee colony tends to optimize the use of number of members involved in particular predecided tasks [9]. The Firefly Algorithm (FA) is an emerging metaheuristic SI technique based on the idealized behavior of the flashing characteristics of fireflies [10, 11]. Generally, the swarm techniques are computationally intensive.

This paper proposes a novel methodology of Cohort Intelligence (CI) inspired from the candidates' self supervised learning behavior in a cohort. The cohort here refers to a group of candidates interacting and competing with one another to achieve some individual goal which is inherently common to all the candidates. When working in a cohort, every candidate tries to improve its own behavior by observing the behavior of every other candidate in that cohort. Every candidate may follow a certain behavior in the cohort which according to itself may result into improvement in its own behavior. As certain qualities make a particular behavior which, when a candidate follows, it actually tries to adapt to the associated qualities. This makes every candidate learn from one another and helps the overall cohort behavior to evolve. The cohort behavior could be considered saturated, if for considerable number of learning attempts the individual behavior of all the candidates does not improve considerably and candidates' behaviors become hard to distinguish. The cohort could be assumed to become successful when for a considerable number of times the cohort behavior saturates to the same behavior.

The remainder of this paper is organized as follows. The framework and detailed formulation of the CI methodology is presented in Section 2. In Section 3, the validation of the CI approach is presented by solving a four test problems along with a brief review of the other methods solving these problems. The evident features, advantages, some limitations of the CI approach along with the concluding remarks and a note on future directions are presented in Section 4 of the paper.

II. COHORT INTELLIGENCE

Consider a general unconstrained problem (in the minimization sense) as follows:

978-1-4799-0652-9/13 $31.00 © 2013 IEEE
DOI 10.1109/SMC.2013.241

$$
\begin{aligned}
&\text{Minimize} \quad f(\mathbf{x}) = f(x_1, \ldots x_i, \ldots, x_N) \\
&\text{Subject to} \quad \Psi_i^{lower} \le x_i \le \Psi_i^{upper} \quad , \quad i = 1, \ldots, N
\end{aligned}
\tag{1}
$$

As a general case, assume the objective function $f(\mathbf{x})$ as the behavior of an individual candidate in the cohort which it naturally tries to enrich by modifying the associated set of characteristics/attributes/qualities $\mathbf{x} = (x_1, \ldots x_i, \ldots, x_N)$.

Having considered a cohort with number of candidates C, every individual candidate c $(c = 1, \ldots, C)$ belongs a set of characteristics/attributes/qualities $\mathbf{x}^c = (x_1^c, \ldots x_i^c, \ldots, x_N^c)$ which makes the overall quality of its behavior $f(\mathbf{x}^c)$. The individual behavior of each candidate c is generally being observed by itself and every other candidate (c) in the cohort. This naturally urges every candidate c to follow the behavior better than its current behavior. More specifically, candidate c may follow $f^*(\mathbf{x}^{(c)})$ if it is better than $f^*(\mathbf{x}^c)$, i.e. $f^*(\mathbf{x}^{(c)}) < f^*(\mathbf{x}^c)$. Importantly, following a behavior $f(\mathbf{x})$ refers to following associated qualities $\mathbf{x} = (x_1, \ldots x_i, \ldots, x_N)$ with certain variations t associated with them. However, following better behavior and associated qualities is highly uncertain. This is because; there is certain probability involved by which it selects certain behavior to follow. In addition, a stage may come where the cohort behavior could become saturated. In other words, at a certain stage, there could be no improvement in the behavior of every individual candidate for a considerable number of learning attempts. Such situation is referred to as saturation stage. This makes every candidate to expand its search around the qualities associated with the current behavior being followed. The mathematical formulation of the CI methodology is explained below in detail with the algorithm flowchart in Fig. 1.

The procedure begins with the initialization of number of candidates C, sampling interval Ψ_i for each quality x_i, $i = 1, \ldots, N$, learning attempt counter $n = 1$, and the set up of sampling interval reduction factor $r \in [0,1]$, convergence parameter ε and number of variations t. The values of C, t and r are chosen based on preliminary trials of the algorithm.

Step 1 The probability of selecting the behavior $f^*(\mathbf{x}^c)$ of every associated candidate c $(c = 1, \ldots, C)$ is calculated as follows:

$$
p^c = \frac{1 / f^*(\mathbf{x}^c)}{\sum_{c=1}^{C} 1 / f^*(\mathbf{x}^c)} \quad , \quad (c = 1, \ldots, C) \tag{2}
$$

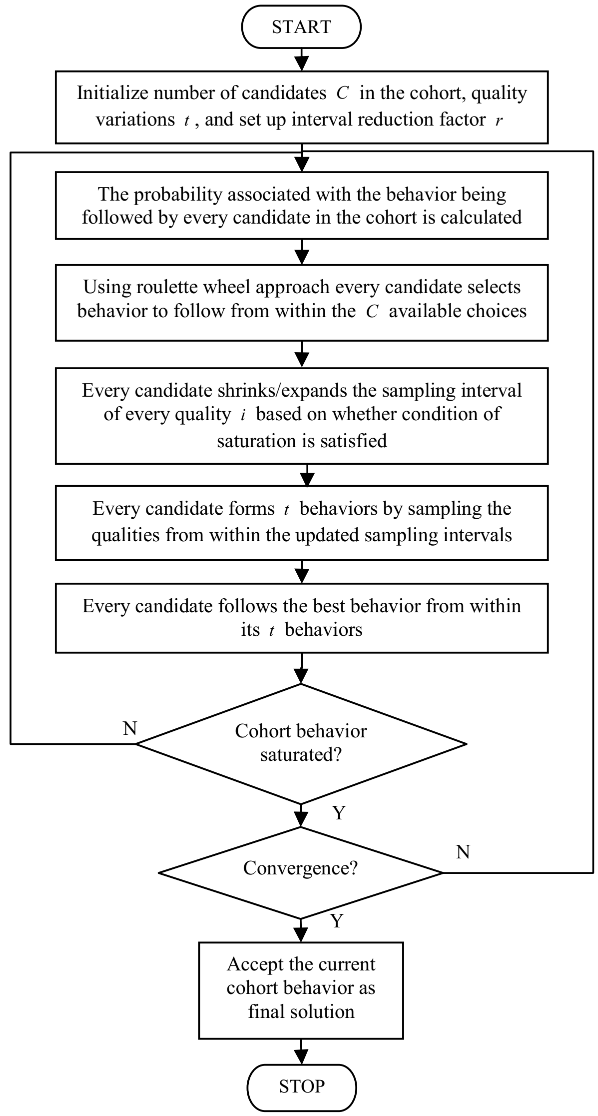


Figure 1. Cohort Intelligence (CI) Algorithm Flowchart

Step 2 Every candidate c $(c = 1, \ldots, C)$ generates a random number $rand \in [0,1]$ and using a roulette wheel approach decides to follow corresponding behavior $f^*(\mathbf{x}^{c[?]})$ and associated qualities $\mathbf{x}^{c[?]} = (x_1^{c[?]}, \ldots x_i^{c[?]}, \ldots, x_N^{c[?]})$. The superscript indicates that the behavior is selected by candidate c and not known in advance. The roulette wheel approach could be most appropriate as it provides chance to every behavior in the cohort to get selected purely based on its quality. In addition, it also may increase the chances of any candidate to select the better behavior as the associated probability stake p^c $(c = 1, \ldots, C)$ presented in Eq. (2) in the interval $[0,1]$ is directly proportional to the quality of the behavior $f^*\left(\mathbf{x}^c\right)$. In other words, better the solution, higher is the probability of being followed by the candidates in the cohort.

Step 3 Every candidate $c \quad (c=1,...,C)$ shrinks the sampling interval $\Psi_i^{c[?]}$, $i=1,...,N$ associated with every variable $x_i^{c[?]}$, $i=1,...,N$ to its local neighborhood. This is done as follows:

$$\Psi_i^{c[?]} \in \left[x_i^{c[?]} - \left(\|\Psi_i\|/2\right),\ x_i^{c[?]} + \left(\|\Psi_i\|/2\right) \right] \tag{3}$$

where $\Psi_i = \left(\|\Psi_i\|\right) \times r$.

Step 4 Each candidate $c \quad (c=1,...,C)$ samples t qualities from within the updated sampling interval $\Psi_i^{c[?]}$, $i=1,...,N$ associated with every quality $x_i^{c[?]}$, $i=1,...,N$ and computes a set of associated t behaviors, i.e. $\mathbf{F}^{c,t} = \left\{ f\left(\mathbf{x}^c\right)^1, ..., f\left(\mathbf{x}^c\right)^j, ..., f\left(\mathbf{x}^c\right)^t \right\}$, and selects the best behavior $f^*\left(\mathbf{x}^c\right)$ from within. This makes the cohort available with C updated behaviors represented as $\mathbf{F}^C = \left\{ f^*\left(\mathbf{x}^1\right), ..., f^*\left(\mathbf{x}^c\right), ..., f^*\left(\mathbf{x}^C\right) \right\}$.

Step 5 The cohort behavior could be considered saturated, if there is no significant improvement in the behavior $f^*\left(\mathbf{x}^c\right)$ of every candidate $c \quad (c=1,...,C)$ in the cohort, and the difference between the individual behaviors is not very significant for successive considerable number of learning attempts, i.e. if

1. $\left\| \max\left(\mathbf{F}^C\right)^n - \max\left(\mathbf{F}^C\right)^{n-1} \right\| \le \varepsilon$, and
2. $\left\| \min\left(\mathbf{F}^C\right)^n - \min\left(\mathbf{F}^C\right)^{n-1} \right\| \le \varepsilon$, and
3. $\left\| \max\left(\mathbf{F}^C\right)^n - \min\left(\mathbf{F}^C\right)^n \right\| \le \varepsilon$, every candidate c $(c=1,...,C)$ expands the sampling interval $\Psi_i^{c[?]}$, $i=1,...,N$ associated with every quality $x_i^{c[?]}$, $i=1,...,N$ to its original one $\Psi_i^{lower} \le x_i \le \Psi_i^{upper}$, $i=1,...,N$.

Step 6 If either of the two criteria listed below is valid, accept any of the C behaviors from current set of behaviors in the cohort as the final objective function value $f^*\left(\mathbf{x}\right)$ as the final solution and stop, else continue to Step 1.

(a) If maximum number of attempts exceeded.

(b) If cohort saturates to the same behavior (satisfying the conditions in Step 5) for $\tau_{\max}$ times.

III. Numerical Experiments and Results

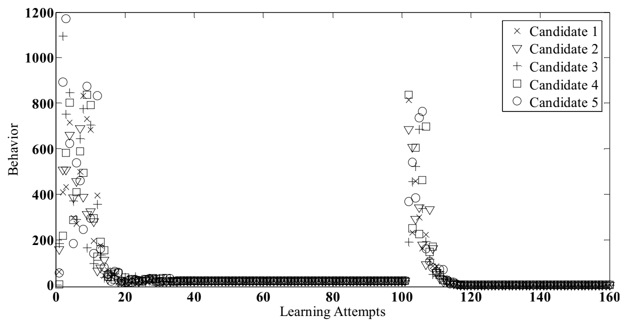
The proposed Cohort Intelligence (CI) algorithm was validated by solving four 10 dimensional test problems such as Rosenbrock function, Sphere function, Ackley function, and Griewank function. The optimum of the Rosenbrock function is located in a deep and narrow parabolic valley with a flat bottom. The gradient based methods may have to spend a large number of iterations to reach the global minimum [12]. The sphere function is unimodal and strongly convex function and the Ackley function is highly multimodal with unique global minimum [13]. The Griewank function has a very large number of local minima, and exponentially increase with the number of dimensions. According to [14], locating global minimum may become extremely difficult with the increase in number of dimensions. These problems have been well studied in the literature and used to compare the performance of various optimization algorithms including Sequential Quadratic Programming (SQP) [15], Chaos-PSO (CPSO) [16], Robust Hybrid PSO (RHPSO) [13] and Linearly Decreasing Weight PSO (LDWPSO) [16]. The SQP method makes the quadratic approximation of a Lagrangean, which is more suitable for solving nonlinear programming problems. The CPSO searches for better solution in the neighborhood of every particle's current location, the global best solution as well as its local best solution balancing the exploration and exploitation. The RHPSO combines PSO with PWLCM for increasing the diversity of search and further employs SQP to efficiently search in the close neighborhood of local minimum. In the LDWPSO the weight associated with the velocity of the particles decreases linearly leading to faster convergence; however, it is essentially governed by the maximum number of iterations for which the algorithm is set to run as well as maximum and minimum allowed weight values.

The proposed CI algorithm was coded in MATLAB 7.7.0 (R2008b) and simulations were run on a Windows platform using Intel Core i5-2400, 3.10GHz processor speed with 3.16 GB RAM. Every problem was solved 20 times. The best, mean and worst solutions, associated standard deviation (SD), computational time, number of function evaluations (FE) along with the associated set of parameters such as number of candidates C, the sampling interval reduction factor r and number of variations t, chosen along with the number of function evaluations (FE), computational time and standard deviation (SD) are presented in Table 1.

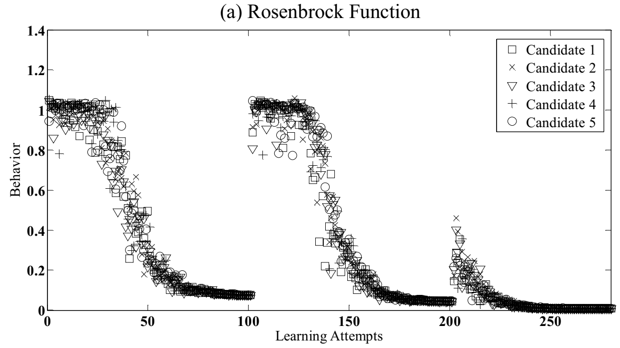
The solution convergence plots for Rosenbrock function and Griewank function are presented in Fig. 2, which exhibit the self supervised learning behavior of every candidate in the cohort. Initially, the behavior of every individual candidate in the cohort can be easily distinguished. As every candidate adapts the qualities of other candidates to improve its own behavior, the cohort saturates to a certain improved behavior. In addition, the saturation makes every candidate to explore further its own qualities and again start learning from one another.

TABLE I. PERFORMANCE OF THE PROPOSED COHORT INTELLGENCE (CI)

Problem	RHPSO [13]	CPSO [16]	LDWPSO [16]	SQP [15]	**Proposed CI**				
	Best *Mean* *Worst*	*Best* *Mean* *Worst*	*Best* *Mean* *Worst*	*Best* *Mean* *Worst*	***Best*** ***Mean*** ***Worst***	*C , t , r*	***FE***	***SD***	***Time (sec)***
Sphere	1.5000E-323 3.5078E-245 5.0380E-248	1.4356E-81 3.4213E-12 1.7103E-10	1.5387E-06 1.2102E-04 1.1486E-03	3.5657E-28 2.5749E-27 8.8173E-27	**2.0000E-15** **2.4900E-06** **1.7780E-05**	**5, 15, 0.80**	**18750**	**4.5800E-03**	**1.55**
Rosenbrock	1.5606E-08 1.2061E-07 3.0398E-07	1.1856E-08 9.3949E-03 9.0066E-02	2.8453E_03 3.1101E+00 1.1050E+01	7.5595E-12 1.4352E+00 3.9866E+00	**0.0000E+00** **0.0000E+00** **0.0000E+00**	**5, 15, 0.80**	**9750**	**0.0000E+00**	**5.20**
Ackley	0.0000E+00 0.0000E+00 0.0000E+00	8.8178E-16 1.5952E-08 6.3330E-07	1.3078E-04 5.9934E-03 2.5325E-02	1.5245E+01 1.9090E+01 1.9959E+01	**1.2322E-07** **2.0911E-07** **2.6499E-07**	**5, 15, 0.85**	**11250**	**4.3200E-08**	**1.50**
Griewank	0.0000E+00 0.0000E+00 0.0000E+00	0.0000E+00 2.1287E-10 6.4174E-09	1.6949E-02 1.7072E-01 7.2835E-01	2.8879E-09 3.5357E-01 3.6312E+00	**7.3960E-03** **1.7100E-02** **4.9183E-02**	**5, 15, 0.997**	**18750**	**8.8300E-03**	**2.00**



(a) Rosenbrock Function



(b) Griewank function

Figure 2. Convergence plot for the test functions

IV. CONCLUSIONS AND FUTURE WORK

The CI methodology is successfully proposed and the self supervising nature of the cohort candidates is successfully demonstrated along with the learning and improving qualities which further improved their individual behavior. The methodology is successfully validated by solving four test problems. The results highlighted that the approach is sufficiently robust with reasonable computational cost. The approach is found to be very competitive and even better than some of the contemporary approaches.

In addition to the advantages few limitations are also observed. The computational performance was essentially governed by the parameter such as sampling interval reduction factor r and needs to develop a self adaptive scheme for its fine tuning. The algorithm could be improved to make it solve constrained as well as combinatorial problems to make it further solve real world problems. Moreover, authors intend to modify the CI approach to solve practically important fault tolerant systems in distributed way [17].

ACKNOWLEDGMENT

The authors would like to acknowledge Rutvi Thanki a student researcher at OAT Research Lab for her help with some part of the algorithm coding and analysis of results.

REFERENCES

[1] K. Deb, “An Efficient Constraint Handling Method for Genetic Algorithms”, Computer Methods in Applied Mechanics in Engineering, 186, pp. 311-338, 2000.

[2] T. Ray, K. Tai, K.C. Seow, “Multiobjective Design Optimization by an Evolutionary Algorithm”. Engineering Optimization, 33(4), pp. 399-424, 2001.

[3] R. Storn, K. Price, “Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces”, Journal of Global Optimization, 11, pp. 341–359, 1997.

[4] A.J. Kulkarni, K. Tai, “Probability Collectives: A Multi-Agent Approach for Solving Combinatorial Optimization Problems”, Applied Soft Computing, 10(3), pp. 759-771, 2010.

[5] A.J. Kulkarni, N.S. Patankar, A. Sandupatla and K. Tai, “A Modified Feasibility-based Rule for Solving Constrained Optimization Problem using Probability Collectives”, Proc. of IEEE 12th International Conferrence of Hybrid Intelligence Systems, pp. 213-218, 2012.

[6] A.J. Kulkarni, I.R.Kale, K.Tai and S.K. Azad, “Discrete Optimization of Truss Structure using Probability Collectives”, Proc. of IEEE 12th International Conference of Hybrid Intelligence System, pp. 225-230, 2012.

[7] J. Kennedy, R. Eberhart, “Particle Swarm Optimization”. Proc. of IEEE International Conference on Neural Networks. pp. 1942–1948, 1995.

[8] M. Dorigo, M. Birattari, T. Stitzle, “Ant Colony Optimization: Arificial Ants as a Computational Intelligence Technique”, IEEE Computational Intelligence Magazine, November, pp. 28-39, 2006.

[9] D.T. Pham, A. Ghanbarzadeh, E. Koc, S. Otri, S. Rahim, M. Zaidi, "The Bees Algorithm", Technical Note, Manufacturing Engineering Centre, Cardiff University, UK, 2005.

[10] A.M. Deshpande, G.M. Phatnani, A.J. Kulkarni, "Constraint Handling in Firefly Algorithm" Proc. of IEEE International Conference on Cybernetics, pp. 186-190, 2013.

[11] X.S. Yang, "Firefly algorithms for multimodal optimization', in: Stochastic Algorithms: Foundations and Applications", SAGA, Lecture Notes in Computer Sciences, Vol. 5792, pp. 169-178, 2009.

[12] A.J. Kulkarni, K. Tai, "Probability Collectives: A Decentralized, Distributed Optimization for Multi-Agent Systems", Applications of Soft Computing, Mehnen, J., Koeppen, M., Saad, A., Tiwari, A. (Eds.), Springer, pp. 441-450, 2009.

[13] W. Xu, Z. Geng, Q. Zhu, X. Gu, "A Piecewise Linear Chaotic Map and Sequential Quadratic Programming based Robust Hybrid Particle Swarm Pptimization", Information Sciences, 218, pp. 85–102, 2013.

[14] M. Locatelli, "A Note on the Griewank Test Function", Journal of Global Optimization 25, pp. 169–174, 2003.

[15] K. Schittkowski, "NLQPL: A FORTRAN-subroutine solving constrained nonlinear programming problems", Annals of Operations Research 5, pp. 485–500, 1985.

[16] L. Liu, W.M. Zhong, F. Qian, "An improved chaos-particle swarm optimization algorithm" Journal of East China University of Science and Technology (Natural Science Edition), 36, pp. 267–272, 2010.

[17] A.J. Kulkarni and K. Tai, "A Probability Collectives Approach for Multi-Agent Distributed and Cooperative Optimization with Tolerance for Agent Failure", In: Agent Based Optimization, I. Czarnowski et al., Eds., Springer-Verlag, 2012, pp. 175-201.