Conflict Monitoring Optimization Heuristic Inspired by Brain Fear and Conflict Systems

Mojtaba Moattari

Signal Analysis Laboratory

Bioelectrical engineering school

Biomedical Engineering Department of Amirkabir University of Technology, Tehran, Iran.

Email: moatary@aut.ac.ir

Mohammad Hassan Moradi

Signal Analysis Laboratory

Bioelectrical engineering school

Biomedical Engineering Department of Amirkabir University of Technology, Tehran, Iran.

Email: mhmoradi@aut.ac.ir

ABSTRACT

This paper deals with a brain-inspired approach to design evolutionary optimization algorithm. The approach is named as Conflict Monitoring Optimization inspired by two related processes in the Brain, conflict monitoring and fear processing systems. First of all, the current issues of optimization metaheuristic are discussed and challenges are addressed. Afterwards, the paper discussed a brief review of researchers' works in danger processing (fear) system of the Brain in three different aspects. Then a model based on fear system model is derived and checked out. Next, the role of Anterior cingulate cortex in conflict monitoring of information is used as a seal of approval to the proposed algorithm. The finalized algorithm at last, is modified to take mutation parameters to intensify the evolutionary aspects of model. After proposing arbitrary subprograms, the proposed algorithm is checked out with 20 benchmark functions with dimension length of 3, 10 and 50. The evaluation result is compared with well-known metaheuristics for 50 different runs and its effectiveness on different function types is discussed afterwards.

Keywords: Unconstrained optimization, fear system, conflict monitoring system, Amygdala , system modeling.

Mathematics Subject Classification: 62J12, 62G99

2012 ACM Computing Classification System:
Theory of computation~Optimization with randomized search heuristics; Theory of computation~Randomized local search; Theory of computation~Non-parametric optimization; Theory of computation~Nonconvex optimization; Computing methodologies~Bio-inspired approaches.

1. INTRODUCTION

Soft computing is a collection of methodologies, which aims to exploit tolerance for imprecision, uncertainty and partial truth to achieve tractability, robustness and low solution cost. The domain of soft computing, especially its branch of metaheuristic optimizers keeps on spreading up. One major reason for higher-level procedure is the demand for different kinds of heuristics suiting for problems with large search space, non-uniform combination of convex regions and multimodal cost space. There are various optimization problems with no information about main cost function which taking place in power and industrial control systems optimization and stabilization (Roshandel and Moattari, 2015). These problems, in need of adaptive switching between diversification and intensification, have only got tackled in terms of intensification and no suitable framework for diversification has been proposed. Also, there are objective functions that are known but possess no differentiable functions, which lead to the uselessness of deterministic approaches like gradient methods (Caraffini et al., 2013). Another important application of metaheuristics is in the optimization of search spaces with multi-convex regions which faster optimization of these kinds of spaces requires a combination of deterministic and stochastic optimization methods (Noel, 2012).

The domain of metaheuristic optimizations has got vast and dense. In this field, nature-inspired optimizations have got so famous due to their power in imitating nature for optimizer algorithm design. Among them, genetic algorithm (GA) uses the idea of natural selection of fittest to survive and pass away the less powerful species in endurance. GA is well known for its powerful tools in solution combination in the genotype domain to accelerate exploration (Caraffini et al., 2013). But it is hardly a good exploitation method. Tabu Search methods, try to reduce search space by considering them as inhibited locations. The name of Tabu is a metaphor used in societies for considering a group of manners as illegal. The main use of this algorithm is in situations with a large search area needed to be pruned as much as possible (Caraffini et al., 2013). Simulated annealing (SA) ideates its procedure from heating and controlled cooling of a material to increase the size of its crystals and reduce their defects. The main problem with SA is its slow performance in functions having low number of local optimas. More than that, its schedule is too slow to deal with expensive cost functions, making algorithm convergence uncertain. Particle Swarm Optimization (PSO) mimics swarming, social flocking behavior of bird or fish schooling in itself. It not only tries to use objectives of individual, but to use social objectives of its society. PSO only uses a linear combination of sought solutions. In more recent works, there are also attempts to apply nature-inspired algorithms in hybrid settings to find solutions for engineering applications, showing the importance of such algorithms in industry (Vaščák, 2012), (Shams et al., 2017), (Vrkalovic et al., 2018), (Precup and David, 2019).

Each of the aforementioned metaheuristics suits specific sets of problems. But in many situations, it is difficult to find out which approach is better to implement. Yet there is no metaheuristic method capable of revising itself while revising search space locality, the way the Brain revises its solutions while reweighting problems importance. So, In order to have a more holistic algorithm capable of dealing with large domain of optimization problems, and also create a framework to deal with mentioned problems easier, a method is proposed inspired by the Brain problem-solving, conflict monitoring and danger processing functions. Due to the strong relationship of such three functions in the Brain, the algorithm is inspired by all of them together. In the next section, neuroscientists' works on these functions are reviewed in both physiological and behavioral aspects and afterwards, the logic behind this relationship is clarified. In section (3), a general optimization algorithm named CMO is proposed and its relationship to the models of fear processing and conflict modulation is clarified. Section (4) applied another revision to make algorithm work more robustly. In section (5), preferences for the generally proposed CMO is specified and a preferred version of CMO is shown in pseudo codes. Comparative analysis has been performed in section (6) and deficiencies and effectiveness of each function type are analyzed. Finally, the potential advantages of the proposed algorithm are included in the conclusion.

2. Related works

Reinforcement learning is the ability to take actions driven by rewards. Contrarily, Avoidance learning is the main organism’s response or behavior in order to avoid an unpleasant or stressful situations. This kind of learning is so crucial for animals helping them escape from unpleasant state (LeDoux, 1996), (LeDoux, 2018), (Johansen et al., 2011). To make that attainable, the Brain has to be able to condition itself to have certain responses in the future to certain stimulus. Conditions need memory to be recorded. Avoidance learning underlies two processes. One is classical conditioning, which leads to fear system activation to develop behavior. And the second process is operant conditioning, which is responsible for the animal’s insight in effectiveness of selected behavior. Although GA concept is used in the proposed method, as it is common, the author refrained from explaining it.

2.1 Fear system, physiological aspect of avoidance learning

It has been already stated that conditioning is necessary to respond in dangerous situations and that is impossible without memory. The quality of the human brain to learn from dangers for the future, is called plasticity (Botvinick, 2007). Plasticity in the learning phase, allows the neural interconnections to rewire themselves in order to make different neural responses leading to wholly different behaviors (LeDoux, 1996). Either the responses are unconditioned, which are the natural response of the body to new stimuli; or they are conditioned, which is for recalling unpleasant experiences (LeDoux, 2014). Ongoing conditioned response overrides natural responses due to painful experiences recorded already.

Change in behaviors is mainly caused by physiological changes during memory formations (LeDoux, 2014). As discussed before, memory is formed to avoid unpleasant situations. The mentioned process is mainly responsible for danger prevention in physiological aspect. Amygdala is believed to be the main region for processing fear conditions and also adding to them (LeDoux, 2014), (LeDoux, 1996).

Concluding from the physiological aspect, a proposed model for the fear process before and after a moment of conditioning can be as follows:

Certain series of neurons fire together due to their current wiring receiving unwanted stimuli -> current firing process stops -> rewiring neurons based on experiences -> returning to neuron firing.

2.2. The conflict monitoring system, Behavioral aspect of avoidance learning

In the Brain, conflict occurs when two different processes are not acting coherently (LeDoux, 2014). Also in psychology, conflict is the arousal of two or more strong motives that cannot be solved together

(Botvinick, 2007). Conflict monitoring is used in information processing for cognitive control and understanding. It is a more general view of fear process which is also investigated by another group of neuroscientists. Studying the works of behavioral neuroscientists can give insight into more powerful models for a proposed fear-inspired optimization algorithm (Botvinick, 2007). By this multilateral investigation, the proposed optimization method can be confirmed in two aspects.

Conflict can contain fear in itself, but no other way around necessarily. Conflict is a behavioral term. But fear is more a physiological term which is based in the avoidance learning process. There are various cases of conflict. One possible case is when different regions of brain give contradictory information about one existing object. In Stroop task, which is designed by Pardo et al., the word “red” with blue color is shown to the subject and he/she is inquired to respond the color of the word as fast as possible (Pardo, et al., 1990). Due to the contradiction between information, subjects for a short period get confused about the color which leads to an increase in activity of Anterior cingulate cortex (ACC), the main region for conflict resolution (Botvinick, 2007). Another type of conflict is when unexpected happenings occur. It happens when reality does not meet one’s physiological and psychological expectations. Physiological expectations are physical needs/instincts and psychological expectations are feelings driven by programs learned by Amygdala . This type of conflict can lead to fear. That is why conflict and fear, both, can be addressed in the avoidance learning process. To state generally, fear in terms of the alarming system occurs when a conflict takes place (Cacioppo, et al., 2006).

It is reported that ACC is responsible for managing incoming and outgoing information from other regions of the Brain. The moment conflict takes place, ACC removes it by controlling the current of information and filters through rewiring nerves to help decision making (Botvinick, 2007).

2.3. Gaussian Mixtures Model

As Gaussian Mixtures Model (GMM) is used as a condition in the proposed algorithm, its mechanism along with the formulation is described in this section. The objective of GMM is to estimate the probability density of given data. When using parametric estimators, a certain underlying distribution should be presumed, and sufficient data has to be available to estimate its parameters (Bishop, 2006).

We are given *N* input vectors in $\Re^d$. In order to estimate the input’s probability density, we will use a linear combination of *M* basis functions:

$$P(x) = \sum_{j=1}^{M} p(x / j)\, p(j) \tag{1}$$

Where $p(x)$ is the density at point x, $p(x/j)$ is called the j-th *component density*, and $p(j)$’s are the *mixing coefficients*. Here, p works as a priori to weight each Gaussian distribution in relation to the relative density of samples locality. Each value in p is a positive number less than 1. Sum of all ingredients in priori and likelihood is equal to 1. As the mixture is Gaussian, it is shown by the following formula:

$$p(x/j)=\frac{1}{(2\pi\sigma_j^{\,2})^{d/2}}\exp\{-\frac{\left\|x-\mu_j\right\|^2}{2\sigma_j^{\,2}}\} \tag{2}$$

By derivating w.r.t. average and standard deviation per each mixture entity, they are updated in a iterative algorithm called Expectation Maximization as follows:

$$\hat{\mu}_j=\frac{\sum_n p(j/x^n)\,x^n}{\sum_n p(j/x^n)} \tag{3}$$

$$\hat{\sigma}_j^{\,2}=\frac{1}{d}\frac{\sum_n p(j/x^n)\left\|x^n-\hat{\mu}_j\right\|^2}{\sum_n p(j/x^n)} \tag{4}$$

$$\hat{p}(j)=\frac{1}{N}\sum_n p(j/x^n) \tag{5}$$

In the next section, the role of GMM is described, and the effects it has is analysed on seeking solutions.

3. Conflict Monitoring Optimizer (CMO), A general framework

In an optimization problem, the main objective is to seek for global optimum with the least function evaluations. When there are no insights regarding the search space, knowledge about the most suited optimizer gets fuzziy and finding the preferred optimizer can lead to a waste of effort in function, excessive function evaluations and additional computational complexity. Lacking a general and easy to use framework causes us to suggest approaches to make optimizer change itself and learn from mistakes. Metrics for mistakes are determined by a predefined set of conditions. The conditions are abstractly regarded equivalent to dangers or conflicts in the proposed brain models. Dangerous conditions like:

Slow progression of fitness – sticking in local optima – repeated searching in abandoned regions – an urgent need to do a random searching – An urgent need for a deterministic search approach due to being in a locality- …

3.1. Designing optimizer capable of reprogramming itself

The main objective of the proposed framework is to switch optimization role from solution-seeking of desired cost to condition-reprioritizing of desired program. This process is mimicked from the way the Brain deals with fear-related programs. As long as the current approach leads to the salvation of conflicts, the satisfaction of expectations and the elimination of danger, it will not be changed. Otherwise, a reprograming takes place by changing behavior, rewiring neurons and etc. The area responsible for triggering conflict is the Amygdala which makes the body aware of conflicts and threats using natural alarms like emotions.

The key idea is to design such optimization process that keeps on running a trustful program as long as series of prioritized conditions are satisfied. To be more precise, series of conditions as indicators of a good optimizer should be defined. Conditions that measure a specific threat in the realm of optimizations. Possible threats for optimization is as long as expectations are met, there is no need to change the behavior of optimizer, tune or reorganize the default running program or alter that.

3.2. CMO algorithm, Definition

Optimization algorithm O(S,C,P,i,C_s,A) is inspired from a CMO system, whenever, using program p from set of programs P, in each iteration lead to improvements in solutions S with costs C; And when conflict occurs in (objective of) one condition C_s is condition set of C, run its corresponding reaction A_S(P,i).

3.3. CMO algorithm, a general framework

The general pseudocode of algorithm is as follows.

```
Input:
    Load groups of programs, conditions, reactions
    Setup the default program as current_program
Output:
    List of solutions and their costs, altered state of each programs hyperparameters
Process:
    Do until termination of criteria
        Do while threat_flag off
                Use the current_program and get solutions, costs, and program_state
                Evaluate all related conditions. Find the first most conflicting condition which is triggered
                If any triggered conditions found, set threat_flag on
        Tune/change/add program based on the condition's reaction algorithm
```

Each condition must have its corresponding reaction. But the program will be chosen to run which the triggered conflict intensity is largest amongst all.

3.4. Benefits for conditioning of an optimization algorithm

There are major benefits all of which to the best of our knowledge's, have not been addressed in previous researches. The following benefits are addressed to have that characteristic:

- A combination of metaheuristics and hyper-heuristics,

- A competitive framework for multiple programs in the main algorithm itself,

- Increasing flexibility to be molded more easily by programmers' side,

- Possibility of changing the algorithm's conditions by the main algorithm itself,

- Easier combination of well-known algorithms together

- The framework is based on the way the Brain functions.

- More flexible implementation of Tabu based approaches in various optimizers.

4. Finalized algorithm scheme of CMO

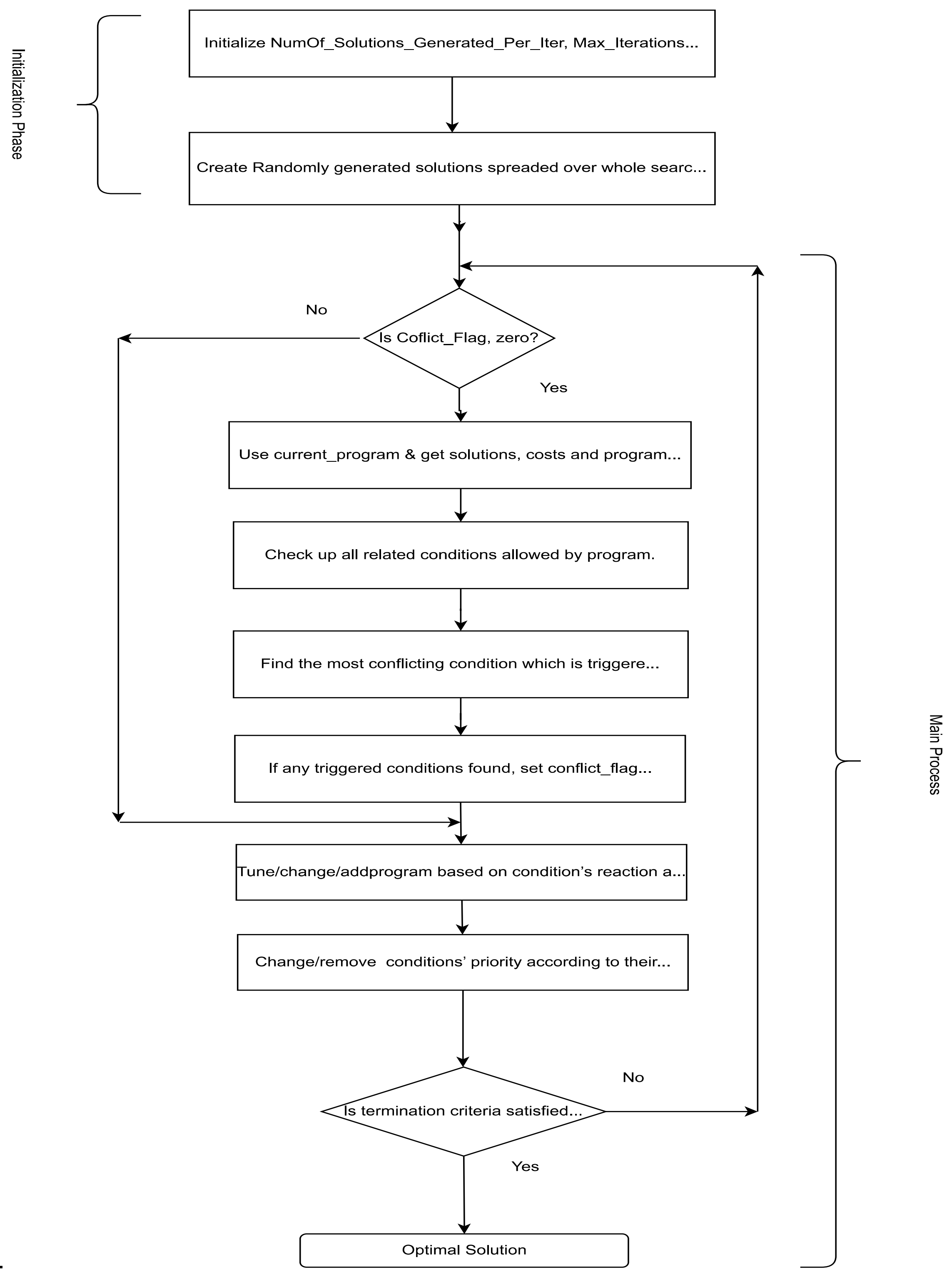
People, when dealing with conflicts, can respond in two possible ways. They, either insist on satisfying expectations by changing behavior, or giving up the objective. The first approach is what has already been modeled in the conflict modulation phase and is evident in the currently proposed algorithm. But the second approach takes place only when there is insight into impossibility in fulfilment of objectives. When brain realizes some dangers are inevitable, the fear system stops functioning on the threat. In the experiment of Pavlovian fear conditioning on rat mice designed by Ledoux et.al, the conditioned stimuli no longer activate the Amygdala circuits in comparison with unconditioned stimuli (LeDoux, 2014). From the experiment, it can be concluded that brain learns to reduce reactions to inevitable unpleasant sensations. In the human beings, this process takes place by attending to current status (Lehmann, et.al., 2012).

The situation for an optimization algorithm can also be the same as danger and conflict processing role in the human beings. It is possible for a search space to need specific sets of conditions among conditions that are already defined in the algorithm. For the algorithm, it is necessary to be smart enough to realize which conditions work more efficiently in what context.

By letting the programs run in the optimizer, insight about each condition is gathered and their effectiveness is analyzed. By setting conflict intensity for each condition, the more effective one condition is, the less its conflict intensity will be. By comparing the conflict intensity of each condition before and after triggering, it is possible to change the priority of them based on their effectiveness. So by adding the following line to the general pseudo-code, the algorithm will be more self-organized.

- Change/remove conditions' priority according to their effectiveness

The final framework is shown in the flowchart in Figure 1.


Initialization Phase
Initialize NumOf_Solutions_Generated_Per_Iter, Max_Iterations...
Create Randomly generated solutions spreaded over whole searc...
No
Is Coflict_Flag, zero?
Yes
Use current_program & get solutions, costs and program...
Check up all related conditions allowed by program.
Find the most conflicting condition which is triggere...
If any triggered conditions found, set conflict_flag...
Tune/change/addprogram based on condition's reaction a...
Change/remove conditions' priority according to their...
No
Is termination criteria satisfied...
Yes
Optimal Solution
Main Process

.

Figure 1- Proposed CMO flowchart

The analysis are taken place with 4 well-known benchmark functions with dimension length of 10, 50 in tables 1 and 2. The evaluation result is compared with well-known metaheuristics for 50 different runs. The values in each table are derived with the following parameter settings:

- 10 iterations with 100 individuals which lead to totally 1000 function evaluations
- 10 clusters each entails 10 solutions which leads to total 100 individuals
- Condition orders 1,3,4,2
- Mutate the parameters of the very conditions

5. Designed programs, conditioned and reactions for CMO

In order to show how preferences are specified in the proposed algorithm, lists of programs, conditions and reactions are designed and proposed. There is no claim or guaranty about the efficiency of selected specifications. Because in this paper, this was not the case of the investigation. Therefore, this work will be improved by a more comprehensive tuning.

It is important to note that every condition has only one corresponding reaction. But the correspondence of programs to conditions may not be one to one.

5.1. Selected programs

The first program is a set of clusters with Gaussian distribution in which solutions are created with the predefined cluster mean and covariance matrix. Mean and covariance matrix of every cluster is already randomly generated. Before popping up of the program, all solutions get updated in a global structure. The main reason for the selection of such algorithm for the program (1) is its power in intensification using covariance update process, which is embedded in one of the conditions' reaction.

The second program is an optimizer having only one algorithm, selected for the random searching power it manifests. This program is useful for idle progress status, ascertaining that it will not get caught into local optima while having new clusters injected to program (1) with mean provided by GA solutions.

5.2. Choosing a reasonable set of conditions for proposed optimization framework

Condition (1) is only checked while program (1) is underrun. This condition checks whether or not the conflict intensity of the condition is lower than a threshold. The conflict intensity designed for this condition is the ratio of regressed solutions counts in the next iteration to the current iteration. This is considered as a metric that evaluates the program in iteration and when there is a slow progression, trigger in the corresponding cluster as a conflict to run its corresponding reaction and change the cluster mean or variance.

Condition (2) is related to the program (1). If clusters progress get idle or clusters get closer to each other, it triggers conflict in the condition to run its corresponding reaction.

Condition (3) is run only in program (2). When at least five repeats of GA are passed, conflict handle gets enabled and its corresponding reaction gets triggered in response to the conflict.

Condition (4) is run only under program (1). When no better solution is found, trigger as conflict.

5.3. Selected reactions

Reaction (1) updates the mean and covariance matrix of clusters that got triggered as conflict.

Reaction (2) removes conflicting clusters and reset their mean and covariance matrix, anew.

Reaction (3) chooses selected GA solutions as an average for clusters in the program (1). Then, it switches the default program to 1.

Reaction (4) saves the best finalized solution. If reaction (3) is run more than five times, reaction (4) sets termination_of_criteria, 1 and exit the whole algorithm.

5.4. The logic behind the selection

GA is selected for better diversification, while Gaussian mixture based solution updating works the best in nearly convex areas of solution space and makes the intensification phase of algorithm work more plausible. By inserting these two programs, the conflict which may trigger by each condition function, determines which program takes control of the process. The CMO makes these two programs compete with each other and continue taking control as long as they improve the fitness. If improvement stops, the algorithm gives control to the other program. As some of the programs are mainly designed for intensification and the other are for diversification, the overall approach balances exploration and exploitation. The specifications are chosen such that most of the benefits discussed in Section (3) part C get satisfied.

5.5. Hyper-parameters tuning of the optimizers

In order to select suitable tuned hyper-parameters, algorithms have been run under plenty of settings and those settings with the best solutions among the others are chosen as the tuned hyper-parameters. The resulted hyper-parameters' values are shown in Table 1. Per each optimizer in the evaluation phase, hyper-parameters have been chosen among the following sets. For the proposed algorithm, each items in vector of Conditions_Conflict_Intensity is selected from from {10, 20, 30}, conflict_update_rate and Maximum_count_of_retained_solutions are selected from {10, 15, 20, 25}, and other parameters have been selected from the commonly used values in literature. The same procedure has been followed in the case of other algorithms.

6. Comparative Analysis

Optimizer preferences are mentioned in *Table 1*. The proposed algorithm converges whenever the error is fewer than 1e-40 or the number of iteration exceede its maximum, i.e. 100.

Table 1- Optimizers' preferences

CLPSO	ICA (Atashpaz-Gargari and Lucas, 2007)	CICA(Bahrami et al., 2010)	GA	SA	BA (Yang & Gandomi, 2012)	PSO	PSO-W	PSO-w-local (Hu, 2012)	CMO*
Inertia=[0.7, 0.9]	Num. of Countries: 60	Number of Countries: 60	Mutation: 1%	T_{min}= 1	V_i: randomly uniform	Inertia = [0.7,0.9]	Inertia = [0.7,0.9]	Inertia= [0.7,0.9]	**Initial Number of solutions**: 20 **Programs order:** first GA then GMM **Conditions Conflict Intensity**=[30 20 10 10] (30 for GA and 20 for GMM. In phase of change conditions' priority according to conflict intensity) **Conflict Update rate**: 10 (if last condition failed in raising fitness, its conflict intensity gets increased to trigger harder afterwards)
Speed limit: 0.1/8* [X_{min} , X_{max}]	Imperialists =10	Imperialists=10	Crossover: 0.8	T_{max}= 10^{-4}	F_i=0.5	Speed limit: 0.1/8* [X_{min} , X_{max}]	Speed limit: 0.1/8* [X_{min} , X_{max}]	Speed limit: 0.1/8* [X_{min} , X_{max}]	**Maximum count of retained solutions:** 20 **GA program preference**:**Mutation**: 1% **Crossover**: .8 **Replacement:** 50%
-	Possessed colonies = 120	Possessed colonies = 120	Replacement: 50%	Repeats per state: 1,2,4,…. 4096	-	-	-	-	**Gaussian mixtures preference:** **Cluster count**=10 **initial clusters mean=** uniformly randomly generated **Mixture mean and cov. matrix updated based on**: fitness value
-	-	-	-	-	-	-	-	-	**Cost conversion to fitness:** linear, if min(costs)>0, fitness=2*max(costs)-costs if min(costs)<0, fitness=costs-2*min(costs)

This article can be cited as M. Moattari and M. H. Moradi, Conflict Monitoring Optimization Heuristic Inspired by Brain Fear and Conflict Systems, International Journal of Artificial Intelligence, vol. 18, no. 1, pp. 45-62, 2020.
Copyright©2020 by CESER Publications

Table 2- Benchmark functions used to compare optimizers. These functions have been selected from CEC2010 benchmarks in order of type. Types are unimodal functions from F1 to F6 and multimodal ones ranged from F7 to F11.

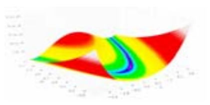
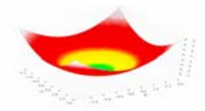
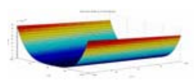
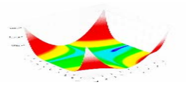
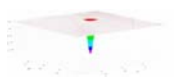
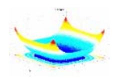
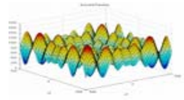
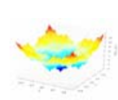
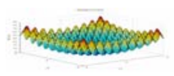
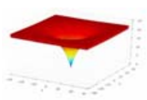
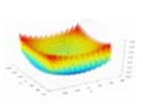
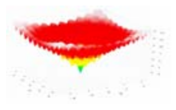
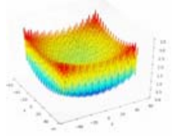
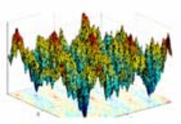
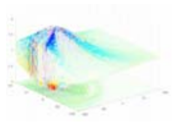
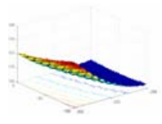
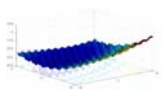
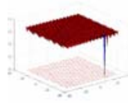
Function	Range	Desired Optima	Formula	Surface Plot
F1)Rosen brock	$x_i \in [-100,100], i = 1,\dots,d$	$\text{Min} = \begin{cases} n = 2, f(1,1) = 0 \\ n = 3, f(1,1,1) = 0 \\ n > 3, f(1_1,1_2,\dots,1_n) = 0 \end{cases}$	$f_1(x) = \sum_{i=1}^{N-1} 100(x_{i+1} x_i^2)^2 + (x_i)^2$	
F2)Sphere	$x_i \in [-5.12,5.12]$	$f(x_1,\dots,x_n) = f(0,\dots,0) = 0$	$f_2(x) = \sum_{i=1}^{n} x_i^2$	
F3)Dixon Price	$x_i \in [-10,10], i = 1,2$	$f(x*) = 0, \quad x_i = 2^{-\frac{2^i-2}{2^i}}$ $, i = 1,\dots,d$	$f_3(x) = (x_1 - 1)^2 + \sum_{i=2}^{d} i(2x_i^2 - x_{i-1})^2$	
F4)Beale	$x_i \in [-4.5,4.5], i = 1,2$	$f(3,0.5) = 0$	$f_4(x,y) = (1.5 - x + xy)^2 + (2.25x + xy^2)^2$	
F5)Easom	$x_i \in [-100,100], i = 1,\dots,d$	$f(0,\dots,0) = 0$	$f_5(x) = -\cos(x_1)\cos(x_2)e^{-(x_1-\pi)^2-(x_2-\pi)^2}$	
F6)Quartic	$x_i \in [-1.28,1.28], i = 1,\dots,d$	$f(0,\dots,0) = 0$	$f_{6(x)} = \sum_{i=1}^{n} ix_i^4$	
F7)Schwefel	x_i $\in [-0.5,0.5], i = 1,\dots,d$	$f(420.9687,\dots,420.9687) = 0$	$f_7(x) = 418.9829d - \sum_{i=1}^{d} x_i \sin\left(\sqrt{\lvert x_i \rvert}\right)$	
F8)Weierstrass	$x_i \in [-0.5,0.5], i = 1,\dots,d$	$f(x_{opt_i}) = f(o) = x_{bias} = -0.5$	$f_{8(x)} = \sum_{i=1}^{D} \left(\sum_{k=0}^{k_{max}} a^k \cos\left(2\pi b^{k(x_i+0.5)}\right) \right) - D. \sum_{k=0}^{k_{max}} a^k \cos(2\pi b^k . 0.5)$	
F9)Rastrigin	$x_i \in [-5.12,5.12], i = 1,\dots,d$	$f(0,\dots,0) = 0$ $a = 0.5, b = 3, K_{max} = 20$	$f_{9(x)} = \sum_{i=1}^{N} (x_i^2 10\cos(2\pi ix) + 10)$	
F10)Ackley	$x_i \in [-32.768,32.768], i = 1,$	$f(0,\dots,0) = 0$ $a = 20, b = 0.2 \text{ and } c = 2\pi$	$f_{10(x)} = -ae^{-b\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2}} e^{\frac{1}{d}\sum_{i=1}^{d} \cos(cx_i)} + a + e$	
F11)Griewank	$x_i \in [-600,600], i = 1,\dots,d,$	$f(0,\dots,0) = 0$	$f_{11(x)} = \sum_{i=1}^{D} \frac{x_i^2}{4000} - \prod_{i=1}^{D} \cos\left(\frac{x_i}{\sqrt{i}}\right)$	

Table 3- Benchmark functions used to assess optimizer's stability. F12 to F16 are rotated functions and F17 to F20 are shifted rotated functions which are both for stability testing purposes. In upcoming results, Types are shown in their own color to be discerned more easily.

Function	Range	Desired Optima	Formula	Surface Plot
F12)Rotated Ackley	$x_i \in [-32.768, 32.768],$	$f(0,\ldots,0) = 0$	$f_{12(x)} = -20.e^{-0.2\sqrt{\frac{1}{D}\times\sum_{i=1}^{D} z_i^2}}\, e^{\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi z_i)} + 20 + e + f_{opt}$	
F13)Rotated Rastrigin	$x_i \in [-5.12, 5.12], i = 1,$ $z = R(0.0512.(x - x\_(o$	$f(0,\ldots,0) = 0$	$f_{13}(x) = \sum_{i=1}^{D} (z_i^2 10\cos(2\pi z_i) + 10)$	
F14)Rotated Schwefel	$x_i \in [-500, 500], i = 1,$	$f(x) = 0, x = \frac{1}{6} \times R^{-1} \times [4$	$f_{14(x)} = 418.9829d - \sum_{i=1}^{d} z_i \sin(\sqrt{\lvert z_i \rvert}$	
F15)Rotated Griewank	$x_i \in [-600, 600], i = 1, ,$	$f(0,\ldots,0) = 0$	$f_{15(x)} = \sum_{i=1}^{D} \frac{z_i^2}{4000} - \prod_{i=1}^{D} \cos\left(\frac{z_i}{\sqrt{i}}\right) +$	
F16)Rotated Weierstrass	$x_i \in [-0.5, 0.5]$ $a = 0.5, b = 3, K\_max =$	$f(0,\ldots,0) = 0, sourcefun$	$\sum_{i=1}^{D}\left(\sum_{k=0}^{k_{max}} a^k \cos(2\pi b^{k(z_i+0.5)})\right) - D.\sum_{k=0}^{k_{max}} a^k \cos(2\pi b^k.0.5) + f$	
F17)Rotate Shift Expand Scaffer	$x_i \in [-100, 100]$	$f(0,\ldots,0) = 0, sourcefu$	$\sum_{i=1}^{D-1} p(z_i, z_{i+1}) + p(z_D, z_1)$ $p(u,y) = \frac{\sin(\sqrt{u^2 + y^2}) - 0.5}{(1 + 0.001.(u^2 + y^2))^2} + 0.5$	
F18)Rotate Shift Griewank	$x_i \in [-600, 600], i = 1,$	$f(0,\ldots,0) = 0, sourcefun$	$\sum_{i=1}^{D} \frac{z_i^2}{4000} - \prod_{i=1}^{D} \cos\left(\frac{z_i}{\sqrt{i}}\right)$	
F19)Rotate Shift Rastrigin	$x_i \in [-5.12, 5.12], i = 1,$	$f(0,\ldots,0) = 0, sourcefun$	z=R(6.(x-x_opt)) $f_{19(x)} = \sum_{i=1}^{D} (z_i^2 10\cos(2\pi z_i) + 10) + f_{(bias)_{19}}$	
F20)Rotate Shift Ackly	$x\_i \in [-32.768, 32.768],$	$f(0,\ldots,0) = 0, sourcefun$	$f_{20(x)} = -20e^{-0.2\sqrt{\frac{1}{D}\times\sum_{i=1}^{D} z_i^2}}\, e^{\frac{1}{D}\times\sum_{i=1}^{D}\cos(2\pi z_i)}$	

6.1. Crucial Multimodal optimization problems

Due to the nature of the proposed optimizer, the main focus of evaluation is on the multimodal benchmark functions like Griewank, Weirstrass and Shwefel and shifted/rotated variations of them. These functions, mentioned and described in rows F11, F7 and F8 of Table 2, are highly multimodal and only one of their peaks is the global optima. Do to managing power of the proposed optimizer to switch between diversifying algorithms like GA and intensifying algorithms like line search, the algorithm provides satisfactory evaluation results in the mentioned functions.

6.2. Evaluation Results on the provided benchmarks

In order to discover the performance of the algorithm, it is compared with other well-known metaheuristics in terms of average error of global optima location from its true position. Average has been taken place among 50 trials. A P-test has been used to assess whether or not each baseline optimizer's mean results of optimum error are in the same distribution as our proposed CMO results. Null hypothesis is regarded as the indifference of CMO with respective evaluating optimizer per benchmark function. None of the P-value results are higher than 0.05 for 95% of the trials per each experiment. Results show the significance of experiments in rejecting null hypothesis verified by the fact that p-values are less than 0.05. It is noteworthy that the codes for the proposed algorithm with the benchmark functions has been provided in the link https://github.com/moatary/CMO. The standard for distributions randomness and uniform densities selection are based on the literature standards.

Due to the novelty of idea in automating hybridization of algorithms, the paper will not have strong competitive purposes for comparison of proposed method with other algorithms. Preferably, objective of this section is to prove this algorithm can potentially compete with other methods and eventually is worth improvement. Although the specification of parameters of the proposed framework deeply depends on the algorithms and programs coded inside, CMO needs a more robust parameter tuning. Moreover, rules are not perfectly assessed and most of effort was focused on the design of the main framework. Experimental analysis of CMO was approached using four different groups of benchmark functions; i.e., Unimodal, multimodal, shifted and shifted rotated sets. Famous metaheuristic optimizations are compared with this method and consistent results were screened using p-value. Moreover, a comparison is made between CMO and its containing programs. For the CMO algorithm to get run, it needs programs and rules to be put into it. Assessment of CMO is also attained by comparing the performance of the program in and out of CMO. CMO is fed by two main programs and multiple rules. Programs are a simple GA and an expectation maximization.

As evidenced by Table 4, although results for multimodal functions did not outperform algorithms like CLPSO, but they show that overall rank of CMO almost close to the best rank in table, which is CLPSO in this case. CLPSO enjoys score of 256 while CMO ranked 2nd among 9 optimizers with score of 247. The score is sum of the overall algorithms outperformance in all 20 functions mean optimization error results. After all, result with the weak tuning of hyper-parameters show that proposed brain-inspired CMO framework can lead to more powerful result by choosing better conditions and programs and better tuning of hyper-parameters.

Table 4- Mean error functions optima from their true values. All functions with NFE=30000 had 3-dimensional input vectors. Dimensions for

N	Benchmark	Type	NFE	CLPSO	ICA	CICA	GA	SA	BA	PSO	PSO-W	PSO-w-local	CMO
F1	**Rosenbrock**	**Unimodal**	180000	2.46E+00 ± 1.70E+00	2.00E-01 ± 3.60E-01	**2.00E-02 ± 2.00E-02**	1.63E+01 ± 7.60E+00	**2.09E-04 ± 2.67E-04**	1.97E+01 ± 1.42E+01	2.14E+00 ± 1.81E+00	3.08E+00 ± 7.69E-01	3.92E+00 ± 1.19E+00	1.42E+03 ± 7.98E-01
F2	**Sphere**	**Unimodal**	180000	5.15E-29 ± 2.16E-28	2.87E+01 ± 2.07E+00	2.50E-07 ± 1.10E-01	3.92E-05 ± 4.86E-04	4.14E-03 ± 4.87E-03	7.85E-21 ± 3.74E-23	**1.23E-30 ± 3.41E-29**	9.78E-30 ± 2.50E-29	**1.23E-30 ± 6.17E-30**	1.24E-02 ± 1.79E-04
F3	**Dixon Price**	**Unimodal**	500000	**1.32E-05 ± 6.95E-07**	9.11E+00 ± 3.73E-01	6.28E-02 ± 5.07E-03	6.67E-01 ± 5.32E-02	1.71E+03 ± 6.92E+01	6.67E-01 ± 5.07E-02	1.00E+00 ± 5.05E-02	1.90E-02 ± 1.17E-03	**3.40E-06 ± 2.30E-06**	2.23E-02 ± 2.62E-04
F4	**Beale**	**Unimodal**	500000	9.04E-02 ± 9.79E-03	5.05E+02 ± 2.53E+00	9.40E-01 ± 6.32E-02	**2.68E-11 ± 2.83E-12**	4.60E-06 ± 4.27E-07	7.62E-01 ± 4.91E-02	1.42E+01 ± 1.28E+00	1.01E-01 ± 6.72E-03	9.93E-02 ± 1.59E-01	**1.85E-06 ± 1.95E-08**
F5	**Easom**	**Unimodal**	500000	6.52E-08 ± 9.39E-09	6.26E-11 ± 7.71E-12	3.27E-06 ± 4.29E-07	-8.11E-05 ± -7.41E-06	**1.23E-30 ± 9.66E-29**	**1.23E-30 ± 3.65E-29**	2.68E-09 ± 3.34E-10	7.27E-07 ± 7.92E-08	7.05E-08 ± 3.40E-08	2.32E-02 ± 3.16E-04
F6	**Quartic**	**Unimodal**	500000	1.23E-30 ± 7.66E-24	**1.23E-30 ± 1.25E-19**	1.23E-30 ± 3.82E-29	1.23E-30 ± 1.70E-30	7.81E-12 ± 8.07E-13	**1.23E-30 ± 4.28E-29**	1.23E-30 ± 2.41E-29	1.23E-30 ± 8.61E-29	1.23E-30 ± 9.43E-30	2.54E-02 ± 1.28E-04
F7	**Schwefel***	**Multimodal**	180000	**1.23E-30 ± 5.54E-29**	5.99E-01 ± 6.66E-02	1.98E+04 ± 7.94E+01	1.08E+01 ± 2.08E-01	5.04E-04 ± 7.02E-05	2.56E+01 ± 2.36E+00	9.82E-02 ± 7.21E-03	3.20E+02 ± 1.63E+00	3.26E+02 ± 1.32E+02	**2.73E-05 ± 3.63E-06**
F8	**Weierstrass**	**Multimodal**	180000	**1.23E-30 ± 6.27E-29**	5.51E-02 ± 2.09E-01	1.29E-04 ± 6.60E-04	4.35E-05 ± 2.77E-05	3.31E-03 ± 2.74E-03	5.33E-12 ± 5.99E-08	**1.23E-30 ± 9.06E-29**	1.30E-04 ± 3.30E-04	1.41E-06 ± 6.31E-06	4.33E-14 ± 5.36E-15
F9	**Rastrigin**	**Multimodal**	500000	**1.23E-30 ± 4.86E-29**	1.66E-06 ± 9.12E-06	**9.34E-09 ± 3.42E-08**	4.75E-10 ± 2.14E-08	9.79E-05 ± 1.89E-03	7.96E+00 ± 8.61E+00	9.95E-01 ± 6.09E-01	6.76E+02 ± 5.43E+01	3.88E+00 ± 2.30E+00	6.75E-02 ± 3.87E-04
F10	**Ackley**	**Multimodal**	500000	4.32E-14 ± 2.55E-14	7.11E-05 ± 8.20E-06	1.02E-07 ± 1.23E-07	1.47E-05 ± 8.97E-08	5.04E-04 ± 4.23E-04	2.63E-12 ± 2.49E-12	**1.23E-30 ± 7.33E-29**	6.32E-11 ± 1.73E-15	**6.04E-15 ± 1.67E-15**	3.04E-08 ± 1.52E-10
F11	**Griewank***	**Multimodal**	500000	4.56E-03 ± 4.81E-03	1.03E-10 ± 8.14E-10	**3.47E-14 ± 5.07E-15**	1.56E+01 ± 2.08E+01	1.34E-03 ± 1.95E-04	1.36E-09 ± 2.71E-06	1.14E-01 ± 4.96E-02	2.43E-01 ± 3.07E-02	7.80E-02 ± 3.79E-02	**4.00E-12 ± 3.11E-13**
F12	**Rotated Ackley**	**Robustness Evaluator**	30000	**3.56E-05 ± 4.35E-06**	3.39E+02 ± 1.96E+00	1.92E+00 ± 2.41E-01	1.55E+01 ± 1.54E+00	2.08E-04 ± 1.62E-05	1.99E+01 ± 4.78E-01	1.73E+01 ± 3.43E-01	2.80E-01 ± 2.11E-02	**6.39E-15 ± 3.18E-15**	2.23E-02 ± 2.98E-04
F13	**Rotated Rastrigin**	**Robustness Evaluator**	30000	5.97E+00 ± 2.93E-01	2.48E+02 ± 4.94E+00	7.51E+01 ± 6.17E+00	1.54E+05 ± 4.48E+02	**4.14E+00 ± 3.42E-01**	4.48E+00 ± 1.89E-01	1.62E+01 ± 5.45E-01	9.90E+00 ± 4.87E-01	9.25E+00 ± 2.74E+00	**1.70E-05 ± 9.12E-08**
F14	**Rotated Schwefel**	**Robustness Evaluator**	30000	1.14E+02 ± 1.30E+00	2.32E+06 ± 8.09E+02	1.07E+04 ± 1.46E+01	3.62E+03 ± 1.33E+02	**5.36E-02 ± 6.76E-03**	3.84E+01 ± 2.20E+00	5.35E+03 ± 8.45E+01	5.69E+02 ± 3.66E+00	4.72E+02 ± 3.07E+02	**5.38E-04 ± 6.82E-06**
F15	**Rotated Griewank**	**Robustness Evaluator**	180000	4.50E-02 ± 4.50E-03	2.56E-02 ± 1.79E-03	2.68E+01 ± 6.97E-01	5.61E-06 ± 6.03E-07	**3.36E-06 ± 3.54E-07**	1.11E-15 ± 1.61E-16	1.44E-02 ± 8.86E-04	2.54E+00 ± 2.13E-01	8.04E-02 ± 4.46E-02	**3.26E-05 ± 3.58E-07**
F16	**Rotated Weierstrass**	**Robustness Evaluator**	180000	3.72E-10 ± 5.23E-11	**6.83E-07 ± 3.55E-08**	6.04E+01 ± 4.14E+00	1.22E-04 ± 1.56E-05	**3.65E-07 ± 2.99E-08**	9.54E-07 ± 7.08E-08	9.32E-06 ± 7.58E-07	3.74E+00 ± 2.13E-01	2.14E-01 ± 3.65E-01	1.23E-06 ± 7.10E-09
F17	**Rotate Shift Expand Scaffer**	**Robustness Evaluator**	30000	1.76E+00 ± 1.65E-01	4.82E+01 ± 2.18E+00	4.12E+01 ± 6.27E-01	4.88E+00 ± 1.80E-01	**4.91E-05 ± 6.41E-06**	4.51E+00 ± 3.95E-01	4.90E+00 ± 1.72E-01	2.42E+00 ± 1.92E-01	1.92E+00 ± 7.77E-01	**4.90E-07 ± 4.60E-09**
F18	**Rotate Shift Griewank**	**Robustness Evaluator**	30000	3.55E+01 ± 7.02E-01	2.74E+03 ± 3.80E+01	6.52E+02 ± 3.74E+00	4.34E+02 ± 5.00E+01	**5.51E-03 ± 4.58E-04**	2.19E-01 ± 1.18E-02	5.38E+01 ± 6.59E+00	4.26E+01 ± 8.80E-01	3.92E+01 ± 7.64E+00	**8.63E-06 ± 8.06E-08**
F19	**Rotate Shift Rastrigrin**	**Robustness Evaluator**	30000	6.01E+01 ± 1.78E+00	2.66E+04 ± 2.37E+01	2.41E+04 ± 1.81E+02	6.58E+02 ± 6.30E+01	**9.32E-03 ± 1.08E-03**	9.48E+02 ± 7.76E+00	6.92E+02 ± 2.57E+01	3.11E+02 ± 1.64E+01	1.68E+02 ± 2.63E+02	**7.91E-02 ± 9.73E-04**
F20	**Rotate Shift Ackly**	**Robustness Evaluator**	180000	7.20E+00 ± 4.15E-01	2.81E+02 ± 2.92E+00	1.24E+02 ± 2.11E+00	2.02E+01 ± 1.19E+00	**2.15E-04 ± 2.46E-05**	2.00E+01 ± 2.34E+00	2.07E+01 ± 2.76E+00	9.01E+00 ± 4.08E-01	7.23E+00 ± 4.99E+00	**2.02E-06 ± 1.39E-08**

180000 and 500000 cases were 10 and 30 respectively. Two best results have been shown in bold.

Results on most of the shifted rotated functions are plausible. Functions like Griewank and Schwefel lead to better results among other algorithms. The CMO successfully outperforms other algorithms in the mentioned functions and it was ranked at most two comparing to other optimizers. These functions are of most complex ones over the others. Algorithm outperformance in 7 out of 9 shifted/rotated benchmark functions shows high robustness of the proposed CMO. These results in Table 3 show that, in case of aforementioned metaheuristics, being combination of a GA

and a GMM (described by first program) for solution updating, the intelligent decision among two programs is not taking place, leading to higher mean error with respect to CMO itself. As previously discussed, such intelligent decision in CMO happens using set of rules that exert changes to the optimization process when it gets triggered.

Table 5 shows the number of algorithms each algorithm outperformed in each of shifted or rotated functions. Values for CMO in this table is not less any of other functions which shows that in term of robustness, CMO does not have anything less than other algorithms.

Table 5- Count of algorithm outperformances

Benchmark	NFE	CLPSO	ICA	CICA	GA	SA	BA	PSO	PSO-W	PSO-W-LOCAL	CMO
Sphere	180000	6	0	4	3	2	5	**9**	7	**8**	1
Rosenbrock	180000	5	7	**8**	2	9	1	6	4	3	0
Ackley	500000	7	1	3	2	0	6	**9**	5	**8**	4
Griewank	500000	4	7	**9**	0	5	6	2	1	3	**8**
Weierstrass	180000	**8**	0	3	4	1	6	**9**	2	5	7
Rastrigin	500000	**9**	6	7	8	5	1	3	0	2	4
Schwefel	180000	**9**	5	0	4	7	3	6	2	1	**8**
Rotated Ackley	30000	**8**	0	4	3	7	1	2	5	**9**	6
Rotated Griewank	180000	3	4	0	7	**8**	**9**	5	1	2	6
Rotated Weierstrass	180000	**9**	7	0	3	**8**	6	4	1	2	5
Rotated Rastrigin	30000	6	1	2	0	**8**	7	3	4	5	**9**
Rotated Schwefel	30000	6	0	1	3	**8**	7	2	4	5	**9**
Beale	500000	6	0	2	**9**	7	3	1	4	5	**8**
Easom	500000	4	6	1	**9**	7	**8**	5	2	3	0
Quartic	500000	4	5	6	2	1	3	7	**8**	**9**	0
Rotate Shift Ackly	180000	7	0	1	3	**8**	4	2	5	6	**9**
Rotate Shift Expand Scaffer	30000	7	0	1	3	**8**	4	2	5	6	**9**
Rotate Shift Griewank	30000	6	0	1	2	**8**	7	3	4	5	**9**
Rotate Shift Rastrigrin	30000	7	0	1	4	9	2	3	5	6	**8**
Dixon Price	500000	**8**	1	5	3	0	4	2	7	**9**	6
Total Score		**256**	106	124	143	227	185	174	166	172	**247**

The analysis is taken place in 20 well-known benchmark functions with dimension length of 10, 50 in tables 1 and 2. The evaluation result is compared with well-known metaheuristics for 50 different runs. The values in each table are derived with the following parameter settings:

- 10 iterations with 100 individuals which lead to total of 1000 function evaluations
- 10 clusters, each of which entailing 10 solutions, overall leading to total of 100 individuals
- Condition orders 1,3,4,2

The results show that the suggested algorithm can outperform well-known optimizations. Also, there seem to be no serious deficiencies in robustness. Although for the case of Sphere, the error progressed much than expected. By tuning parameters and designing more suitable specifications, this algorithm can show its real power.

7. CONCLUSIONS

In this paper, a new optimization algorithm named CMO is proposed which tries to use models of fear processing and conflict modulation in the Brain as inspiration for prioritizing meta-heuristics while tuning each of their hyper-parameters and their associated condition settings. The framework is provided both behaviorally and physiologically and pseudo code has been obtained. Specifications for evaluation has been modulated on the general framework and cost errors in 50 independent runs for 4 functions and 1000 function evaluations are extracted. The results are derived in similar situations for each optimizer and CMO outperformed other algorithms in three functions.

In future works, comprehensive investigations on selecting useful specifications for CMO will be done. Also, evaluations will be tested out after tuning parameters of specified programs. Conflict intensity will be discussed more clearly, also.

REFERENCES

Archer, T. and Nilsson, L.G. eds., 2014. Aversion, avoidance, and anxiety: Perspectives on aversively motivated behavior. Psychology Press.

Atashpaz-Gargari, E. and Lucas, C., 2007, September. Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition. In 2007 IEEE congress on evolutionary computation (pp. 4661-4667). IEEE.

Bahrami, H., Faez, K. and Abdechiri, M., 2010, March. Imperialist competitive algorithm using chaos theory for optimization (CICA). In 2010 12th International Conference on Computer Modelling and Simulation (pp. 98-103). IEEE.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. springer.

Botvinick, M.M., 2007. Conflict monitoring and decision making: reconciling two perspectives on anterior cingulate function. Cognitive, Affective, & Behavioral Neuroscience, 7(4), pp.356-366.

Cacioppo, J.T., Visser, P.S., Pickett, C.L. and Berntson, G.G. eds., 2006. Social neuroscience: People thinking about thinking people. MIT press.

Caraffini, F., Neri, F., Iacca, G. and Mol, A., 2013. Parallel memetic structures. Information Sciences, 227, pp.60-82.

Dudai, Y., Jan, Y.N., Byers, D., Quinn, W.G. and Benzer, S., 1976. dunce, a mutant of Drosophila deficient in learning. Proceedings of the National Academy of Sciences, 73(5), pp.1684-1688.

Hu, M., Wu, T. and Weir, J.D., 2012. An adaptive particle swarm optimization with multiple adaptive methods. IEEE Transactions on Evolutionary computation, 17(5), pp.705-720.

Iacca, G., Neri, F., Mininno, E., Ong, Y.S. and Lim, M.H., 2012. Ockham's razor in memetic computing: three stage optimal memetic exploration. Information Sciences, 188, pp.17-43.

LeDoux, J.E., 1996. The Emotional Brain Simon Schuster. New York, 384.

Johansen, J.P., Cain, C.K., Ostroff, L.E. and LeDoux, J.E., 2011. Molecular mechanisms of fear learning and memory. Cell, 147(3), pp.509-524.

LeDoux, J.E., 2014. Coming to terms with fear. Proceedings of the National Academy of Sciences, 111(8), pp.2871-2878.

Maren, S., 2005. Synaptic mechanisms of associative memory in the Amygdala . Neuron, 47(6), pp.783-786.

Oltean, M., 2004, January. Searching for a practical evidence of the no free lunch theorems. In International Workshop on Biologically Inspired Approaches to Advanced Information Technology (pp. 472-483). Springer, Berlin, Heidelberg.

Pape, H.C. and Pare, D., 2010. Plastic synaptic networks of the Amygdala for the acquisition, expression, and extinction of conditioned fear. Physiological reviews, 90(2), pp.419-463.

Pardo, J.V., Pardo, P.J., Janer, K.W. and Raichle, M.E., 1990. The anterior cingulate cortex mediates processing selection in the Stroop attentional conflict paradigm. Proceedings of the National Academy of Sciences, 87(1), pp.256-259.

Precup, R.E. and David, R.C., 2019. Nature-Inspired Optimization Algorithms for Fuzzy Controlled Servo Systems. Butterworth-Heinemann.

Rescorla, R.A. and Holland, P.C., 1982. Behavioral studies of associative learning in animals. Annual review of psychology, 33(1), pp.265-308.

Roshandel, E. and Moattari, M., 2015, May. Novel line search based parameter optimization of multi-machine power system stabilizer enhanced by teaching learning based optimization. In 2015 23rd Iranian Conference on Electrical Engineering (pp. 1428-1433). IEEE.

Ruben, D.H., 1993. Avoidance Syndrome: Doing Things Out of Fear. WH Green.

Shams, M., Rashedi, E., Dashti, S.M. and Hakimi, A., 2017. Ideal gas optimization algorithm. International Journal of Artificial Intelligence, 15(2), pp.116-130.

Schumacher, C., Vose, M.D. and Whitley, L.D., 2001, July. The no free lunch and problem description length. In Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation (pp. 565-570). Morgan Kaufmann Publishers Inc..

Tully, K. and Bolshakov, V.Y., 2010. Emotional enhancement of memory: how norepinephrine enables synaptic plasticity. Molecular brain, 3(1), p.15.

Vaščák, J., 2012. Adaptation of fuzzy cognitive maps by migration algorithms. Kybernetes, 41(3/4), pp.429-443.

Vrkalovic, S., Lunca, E.C. and Borlea, I.D., 2018. Model-free sliding mode and fuzzy controllers for reverse osmosis desalination plants. Int. J. Artif. Intell, 16(2), pp.208-222.

Yang, X.S. and Hossein Gandomi, A., 2012. Bat algorithm: a novel approach for global engineering optimization. Engineering Computations, 29(5), pp.464-483.