Journal of Theoretical and Applied Information Technology
15th December 2018. Vol.96. No 23

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

A CLASS OF ABCED CONJUGATE GRADIENT METHOD IN SOLVING GENERAL GLOBAL OPTIMIZATION PROBLEMS

[1,2]GOH KHANG WEN, [1,3]YOSZA DASRIL, [1]ABDUL RANI OTHMAN
[1]Center for Telecommunication Research & Innovation, Faculty of Electronic and Computer Engineering, Universiti Teknikal Malaysia Melaka (UTeM), Malaysia
[2]Faculty of Science and Technology, Quest International University Perak (QIUP), Malaysia
[3]Institute for Mathematical Research (INSPEM), Universiti Putra Malaysia (UPM), Malaysia
E-mail: khangwen.goh@qiup.edu.my, yosza@utem.edu.my, rani@utem.edu.my

ABSTRACT

Conjugate Gradient Method (CG) is one of the well-developed gradient based method in solving optimization problems. It been widely used in solving large scale optimization problems due to its low computational cost and high efficiency in locating optimization solution. However, this method often fails to obtain global optimum solution when solving multimodal nonconvex optimization problems because once this method obtained a local optimum solution, it unable to move to another valley to obtain a better optimum solution. In this paper, ABCED Conjugate Gradient Method which consist of a series of enhanced conjugate gradient methods have been introduced to solve multimodal nonconvex optimization problems. The new developed methods have been tested with several benchmark problems. The numerical results had proved the effectiveness of the ABCED Conjugate Gradient Methods. The results showed the ABCED Conjugate Gradient with Fletcher-Reeves formula able to globally solve 80.95% of the selected benchmark test function. Then, ABCED Conjugate Gradient with Hestenes-Stiefel and Dai-Yuan formula had globally solved 76.19% of selected benchmark test function. However, ABCED Conjugate Gradient with Polak-Ribiere only able to solve one third of the selected benchmark test function.

Keywords: *Gradient Based Method, Conjugate gradient method, Artificial Bees Colony (ABC), Multimodal Non-convex Optimization, Global Optimization*

1. INTRODUCTION

Mathematicians believe that every daily problem that we face can be modeled into mathematical model entirely. In mathematical terms, the goal of solving those models in the "best" way is called as optimization. These might mean maximize profit, minimize loss, maximize efficiency or minimize the risk in running business; minimize weight or maximize strength in designing a bridge and minimize the time or fuel use in selecting an aircraft flight plan.

There are several gradient based optimization techniques have been proposed to solving those mathematical models, such as steepest descent method, conjugate gradient method and quasi-Newton method. These methods are well-performed to determined local solution or once say globally determined the solution when solving convex optimization problems, in which there have only one local solution and can also called as global solution.

However, most our daily problem happens as non-convex optimization problem, which may contain multi local optimum solution. Most of the time, the local solution is greatly different and meaningless when compare to the global one. Therefore, the most important objective and challenges in solving these non-convex optimization problems is how to determine the optimum value among all the local optimum solution in the domain or we call it as global optimum solution. By the way, those well-performed methods which mentioned above always lose their efficiency when applied to the global minimizer for non-convex problems.

Artificial Bee Colony (ABC) algorithm is one of the most recent swarm intelligent based algorithm which proposed by Dervis Karaboga in year 2005 [1]. It is a biological-inspired optimization algorithm. ABC is inspired by the foraging behavior of honey bee swarm. The process of the swarm of bees searching for food source is the process used to find optimal solution [1]. The exploration and exploitation are two important mechanisms in ABC. Exploitation process starts when the employed bees approach to the food sources. After determining the nectar amounts of the food sources by the employed bees, the onlooker bees will go to the highest probability value of source and determine the nectar amount. When the source is exhausted, it indicated the end of the exploitation process. Meanwhile, exploration process begins when scouts are sent to search for new food sources randomly. However, there are some insufficiencies regarding the ABC. ABC perform better during exploration stage but weaker at exploitation stage. [2][3][4].

Goh et. al. [5] introduced a Simplexed ABC algorithm that improve the accuracy and efficiency of the ABC in solving global optimization problems. The success of Goh et. al. [5] lead this research to the new direction of investigation. The Nelder-Mead simplex method is a derivative free approach which order of convergent is much slower compare to the gradient based method. However, the enhancement of Simplexed ABC has indicated that even with less number of colony involvement, its accuracy of the obtained optimum solution is much more better than original ABC. Therefore, this research has led to a new path which will enhanced the original ABC with a series of Conjugate Gradient Methods.

In this paper, we have introduced a series of conjugate gradient methods so called ABCED Conjugate Gradient Method (ABCED CG) which its algorithm is hybrid from the several variants of conjugate gradient methods into the Artificial Bees Colony (ABC) algorithm for solving general global optimization problems. The main idea of the ABCED CG method is replacing the exploitation process in original ABC with any variants of conjugate gradient method. The performance of the exploitation process will be improved by the efficiency of the conjugate gradient methods. Besides that, via this hybridization process the ability of conjugate gradient method also improved to able to determine the global optimization solution for non-convex optimization problems.

This paper is organized as follows. In Section 2, we define several basic definitions of global optimization and the properties of gradient type method which must be understood before discussing the CG method in more detail. In Section 3, the ABCED CG method is introduced and its algorithm also has been show in the same section. The numerical results which reflect the effectiveness of the SS method in solving general global optimization problems have been presented in Section 4. Finally, the conclusion which ends this paper is discussed in Section 5.

2. CONJUGATE GRADIENT METHOD

Gradient based methods [6][7][8][9] are motivated by the fact that f decreases most rapidly at a point in the direction of $-\nabla f$. Consequently, the iterative $\left\{x^{(k)}\right\}$ which converges to the minimizer x^* of f are computed by an iterative procedure of the form

$$x^{(k+1)} = x^{(k)} + \lambda^{(k)} d^{(k)} \tag{2.1}$$

where $\lambda \in R^1$ is the positive step length scalar and $d^{(k)} \in R^n$ is the searching direction.

The success of the gradient based method is depending on the effective choices of the direction $d^{(k)}$ and the step length $\lambda^{(k)}$. Most gradient based method required the search direction $d^{(k)}$ to be a descent direction (for minimization), in which the search direction should satisfy the property

$$\left(d^{(k)}\right)^{\mathrm{T}} \nabla f(x^{(k)}) < 0 . \tag{2.2}$$

This property can guarantee that the value of function f can be reduced along the direction. Moreover, the search direction of the gradient based methods usually has the forms

$$d^{(k)} = -\left(\nabla f\left(x^{(k)}\right)\right)^{\mathrm{T}} \left(M^{(k)}\right)^{-1} \tag{2.3}$$

where $M^{(k)}$ is a symmetric and nonsingular matrix.

In the steepest descent (SD) method, $M^{(k)}$ is simply the identity matrix I, while in Newton's method $M^{(k)}$ is the exact Hessian $\nabla^2 f\left(x^{(k)}\right)$. In quasi-Newton method, $M^{(k)}$ is an approximation to the Hessian that is updated at every iteration by mean of a low-rank formula. To make sure the $d^{(k)}$ which defined in (2.3) is a descent direction, we required the $M^{(k)}$ to be positive definite, so that we can have

$$\left(d^{(k)}\right)^T \nabla f(x^{(k)}) = -\left(\nabla f(x^{(k)})\right)^T \left(M^{(k)}\right)^{-1} \nabla f(x^{(k)}) < 0.$$

Conjugate-direction methods at first was developed for solving quadratic optimization problem like Newton's Method and are then extended to the general optimization problem. For a quadratic problem, convergence is achieved in a finite number of iterations. Conjugate-direction methods have been found to be very effective in many types of problems and have been used extensively in the past. Generally, conjugate gradient method is a useful technique for solving large-scale problems because it avoids the computation and storage of some matrices associated with Hessian of objective functions.

The conjugate gradient method is designed by the following definition and concept.

Definition 2.1 [*Conjugacy*]

Let $Q \in \mathbb{R}^{n\times n}$ be symmetric and positive definite. We say that the vectors **x**, **y** $\in \mathbb{R}^n \setminus \{0\}$ are *Q*-conjugate (or *Q*-orthogonal) if $\mathbf{x}^T Q\mathbf{y} = 0$.

Proposition [*Conjugacy implies Linear Independence*]

If $Q \in \mathbb{R}^{n\times n}$ is positive definite and the set of nonzero vectors $d_0, d_1, \ldots, d_k$ are (pairwise) *Q*-conjugate, then these vectors are linearly independent.

Let $\{d_i\}_{i=0}^{n-1}$ be a set of nonzero *Q*-conjugate vectors. For any $x_0 \in \mathbb{R}^n$ the sequence $\{x_k\}$ generated according to

$$x_{k+1} = x_k + \alpha_k d_k,\ k \geq 0$$

with $\alpha_k := \arg\min\{ f(x_k + \alpha d_k) : \alpha \in \mathbb{R}\}$ converges to the unique solution, x^* of *P* after n steps, that is $x_n = x^*$.

Similar to steepest descent method the conjugate gradient method also has the form $x_{k+1} = x_k + \lambda_k d_k$. But the direction d_k is calculate using the conjugate direction as follows:

$$d_k = \begin{cases} -g_k, & \text{if } k = 1, \\ -g_k + \beta_k d_{k-1}, & \text{if } k \geq 2, \end{cases} \quad (2.4)$$

where β_k is a parameter that determines the variants of conjugate gradient methods. For example, well-known choices of β_k can be taken from Hestenes-Stiefel (HS), Fletcher-Reeves (FR), Polak-Ribiere (PR), Fletcher-Conjugate Decent (CD) and Dai-Yuan (DY) formulas.

The fundamental assumption is made that if a steady reduction is achieved in the objective function in successive iterations, the neighborhood of the solution will eventually be reached. If **H** is positive definite near the solution, then convergence will, in principle, follow in at most *n* iterations. For this reason, conjugate-direction methods, like the Newton method, are said to have quadratic termination. In addition, the rate of convergence is quadratic, that is, the order of convergence is two.

The use of conjugate-direction methods for the solution of nonquadratic problems may sometimes be relatively inefficient in reducing the objective function, if the initial point is far from the solution. In such a case, unreliable previous data are likely to accumulate in the current direction vector, since they are calculated based on past directions. Under these circumstances, the solution trajectory may wander through suboptimal areas of the parameter space, and progress will be slow. This problem can be overcome by re-initializing these algorithms periodically, say, every *n* iterations, in order to obliterate previous unreliable information, and in order to provide new vigor to the algorithm through the use of a steepest-descent step. Most of the time, the information accumulated in the current direction is quite reliable and throwing it away is likely to increase the amount of computation. Nevertheless, this seems to be a fair price to pay if the robustness of the algorithm is increased.

An effective method for the generation of conjugate directions proposed by Hestenes and Stiefel [10] is the so-called conjugate-gradient method. In this method, directions are generated sequentially, one per iteration. The Hestenes-Stiefel formula is defined by

$$\beta_k^{HS} = \frac{g_{k+1}^T y_k}{d_k^T y_k} \quad (2.5)$$

where, d_k is search direction and $y_k = g_{k+1} - g_k$.

Fletcher-Reeves formula is defined by

$$\beta_k^{FR} = \frac{\|g_{k+1}\|^2}{\|g_k\|^2}, \quad (2.6)$$

where g_k and g_{k+1} are the gradients $\nabla f(x_k)$ and $\nabla f(x_{k+1})$ of $f(x)$ at the point x_k and x_{k+1} respectively, $\|\cdot\|$ denotes the Euclidian norm of vectors. This formula is called the Fletcher Reeves formula. Another formula for β_k is defined by

$$\beta_k^{PR} = \frac{g_{k+1}^T\left(g_{k+1} - g_k\right)}{\left\|g_k\right\|^2}. \quad (2.7)$$

This formula generalizes to the non-quadratic case and is called the Polak-Ribiere formula.

Fletcher formula that we denote by conjugate decent is defined as follows

$$\beta_k^{CD} = \frac{g_k^T g_k}{d_{k-1}^T g_{k-1}}. \quad (2.8)$$

Dai-Yuan proposed formula to compute β_k as follows.

$$\beta_k^{DY} = \frac{\left\|g_{k+1}\right\|^2}{d_k^T y_k} \quad (2.9)$$

where, again d_k is search direction and $y_k = g_{k+1} - g_k$.

To be standardize among all gradient based method, all gradient based approaches in this section will be using Armijo line search for their step size computational. The Armijo line search rule is described as follows. Given

$s > 0$, $\beta \in (0,1)$, $\sigma \in (0,1)$ and
$\lambda_k = \max\{s, s\beta, s\beta^2, \ldots\}$
such that

$$f(x_k + \lambda_k d_k) - f(x_k) \le \sigma\lambda_k g_k^T d_k. \quad (2.10)$$

Algorithm 2.1 (Conjugate Gradient method with Armijo line search)

Input: Initial point $x_0 \in \mathbb{R}^n$, Function to be minimized $f : \mathbb{R}^n \to \mathbb{R}$, and Tolerance $\varepsilon \in \mathbb{R}$.

1. $k = 0$
2. $d_k = -g_k$
3. while $\left(\left\|d_k\right\| \ge \varepsilon\right)$ do
    4. $\lambda_k = 2,\ \beta = 0.618,\ \sigma = 0.8$
    5. while $\left(f(x_k + \lambda_k d_k) - f(x_k) \le \sigma\lambda_k g_k^T d_k\right)$ do
        6. $\lambda_k = \lambda_k \beta.$
    7. $x_{k+1} = x_k + \lambda_k d_k.$
    8. $k = k + 1.$
    9. Compute β_k using any **equation (2.2) – (2.6)**
    10. $d_k = -g_k + \beta_k d_{k-1}$
11. x_k is a minimizer.

3. ARTIFICIAL BEES COLONY (ABC)

Valery Tereshko and Andreas Loengarov [11] had started to solve problems by using honey bee foraging dynamics. They were interested in seeing how the exchanging information interactions between the individual leads to globally intelligent selection of the food sources in an unpredictable environment. Hence, they started to develop a model which will able to quickly search for the "best" food source by considering the bee colony as dynamic system. The system consisted of three essential components, which are the food sources, employed bees and unemployed bees. Meanwhile, the leading modes of the foraging behavior of the bees are recruitment to a nectar source and abandonment of the source.

In the same year, Dervis Karaboga, who was inspired by this idea and initiated Artificial Bee Colony (ABC) which is also an algorithm which adapt to the honey bee swarm's foraging behavior. Similarly, the model included the three essentials components as mentioned above. According to Dervis Karaboga [1], ABC is very simple and flexible compare to the existing swarm based algorithms. Recently, ABC algorithm had been reviewed by many professional researchers.

In year 2010, Guopo Zhu and Sam Kwong [12] did some modifications on the ABC into Global best-guided ABC (Gbest ABC). The modification of ABC into Gbest ABC was inspired by population-based optimization algorithms (PSO), which, in order to improve the exploitation process, took advantage of the information of the global best solution, by modifying the equation

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) \quad (3.1)$$

to be

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) + \varphi_{ij}(x_j - x_{ij}) \quad (3.2)$$

where x_j is the j^{th} element of global best solution, φ_{ij} is the uniform random number in [0,c] and c is the non-negative constant. When the number c is increased, the efficiency of the exploitation process will be improved. At the same time, the number c cannot be too large because it weakens the exploitation process, at the same time causing the Gbest term (3.2) driving the new candidate solution moves over the global best solution ([12]. However, after some experiments had been carried out by both researchers, noticed that, GABC outperformed the original ABC in most of the experiments when c = 1.5. Therefore, this shows that GABC can

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

perform better than the original ABC with appropriate parameter applied.

Anan Banharnsakun and his fellow researchers [3] had initiated Best-so-far ABC in year 2011, by adding three things to the modification of ABC that are, best-so-far method, adjustable search radius and objective-value-based comparison method.

In the original ABC algorithm, all the onlookers choose a food source based on the probability of respective fitness function explored by a single employed bee and the new solutions are generated by using equation (3.1). In the contrary, in the best-so-far method, the onlookers make decision on a new food sources by making use of all the information from all employed bees so that they can compare all the information that are available and are able to select the best-so-far food position. The modified equation to generate new food source is:

$$v_{id} = x_{ij} + \phi f_k (x_{ij} - x_{kj}) \qquad (3.3)$$

where f_k is the fitness value of the best food source so far; x_{kj} is the best-so-far food source selected dimension j. This method can improve the local search ability compared to the original ABC algorithm.

The second modification in this modified algorithm is the adjustable search radius, which is especially for the scout bee. The scout bee will randomly generate a new food source by using equation (3.4) whenever the solution stagnates in the local optimum.

$$v_{ij} = x_{ij} + \phi_{ij}[\omega_{\max} - \frac{iteration}{MCN}(\omega_{\max} - \omega_{\min})]x_{ij} \qquad (3.4)$$

v_{ij} is a new feasible solution of a scout bee that is modified from the current position of an abandoned food source, x_{ij} and ϕ_{ij} is the random number between [-1,1]. The value $\omega_{\max}$ and $\omega_{\min}$ represent the maximum and minimum percentage of the position adjustment for the scout bee [3].

The third modification is the objective-value-based comparison method, which had made changes to the finding of the fitness value.

$$Fitness(f(x)) = \begin{cases} \frac{1}{1+f(x)} & if\ f(x) \geq 0 \\ 1+|f(x)| & if\ f(x) < 0 \end{cases} \qquad (3.5)$$

According to the results from numerical experiments conducted by Anan Banharnsakun and his fellow researchers [3], Best-so-far ABC obtained a better convergence rate than the original ABC. A smaller rate of convergence indicates that less iteration is needed for a function to converge to the optimal solution. Results showed that Best-so-far ABC can produce the optimal solution more quickly on almost all benchmark functions.

At the same time, GuoQiang Li and 2 other researchers [2] proposed an improved ABC called I-ABC and another PS-ABC with the ability of prediction and selection. The latter is the combination of the bright sides from ABC, GABC and I-ABC. Before knowing what is PS-ABC, best-so-far solution, inertia weight and acceleration coefficients are introduced to modify the searching process in I-ABC. I-ABC could not only find the global optimal values for many numerical functions, but also own an extremely fast convergence speed. Yet, in some cases, I-ABC traps in local optimal and therefore not able to find better solutions than ABC or GABC. The equation is modified as the following form:

$$\begin{aligned} v_{ij} = x_{ij} w_{ij} + 2(\phi_{ij} - 0.5)(x_{ij} - x_{kj})\phi_1 \\ + \varphi_{ij}(x_j - x_{kj})\phi_2 \end{aligned} \qquad (3.6)$$

w_{ij} is the inertia weight which controls impacts of the previous solution x_{ij}. x_j is the j^{th} element of global best solution ϕ_{ij}, and φ_{ij} are random numbers between [0,1], ϕ_1 and ϕ_2 are positive parameters that could control the maximum step size. Somehow, when the global fitness is very large, bees are further away from optimal solutions.

To further improve the search efficiency of the bees, the researchers had modified the parameters that are involved in equation (3.6). The inertia weight and acceleration coefficient are defined as follows:

$$w_{ij} = \phi_1 = \frac{1}{(1+\exp(-fitness(i)/ap))} \qquad (3.7)$$

$$\phi_2 = \begin{cases} 1 & \text{employed bee} \\ \frac{1}{(1+\exp(-fitness(i)/ap))} & \text{onlooker} \end{cases} \qquad (3.8)$$

where ap is the fitness value found in first iteration [2]. After all, I-ABC is able to perform better in terms of convergence ability as well in finding a better optimal solution.

For producing high efficient ABC algorithm with the abilities to predict and select, the researchers produced the PS-ABC by gathering all the bright sights of ABC, GABC and I-ABC to form a hybrid ABC algorithm. The main difference between PS-ABC and any of ABC, GABC, and I-ABC is how to determine the candidate solutions process. In PS-ABC, the employed bees will firstly work out 3 possible solutions with 3 types of search equations and then choose and determines the best one as the candidate solution.

From original ABC,

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj})$$

From I-ABC,

$$v_{ij} = x_{ij}w_{ij} + 2(\phi_{ij} - 0.5)(x_{ij} - x_{kj})\phi_1 + \varphi_{ij}(x_j - x_{kj})\phi_2$$

From GABC,

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) + \varphi_{ij}(x_j - x_{ij})$$

Both the I-ABC and PS-ABC were tested with 13 classical functions comparing to the solutions from the ABC and GABC. It was found that I-ABC obtained faster convergence speed than ABC or GABC for most functions although it did not achieve better optimality ability than the ABCs in few of the functions. The results showed that the convergence and searching ability generated by using PS-ABC is better than the other methods for almost all functions. In PS-ABC, the global search ability had increased, and convergence ability of this algorithm had been enhanced at the same time. This shows that there is no specific algorithm to substantially achieve the best solutions for all the optimization problems. Some algorithm gives best solutions in some cases and some not. Hence, researchers nowadays try to search for a well improved or new optimization method [2].

Fei Kang, Junjie Li and Zhenyue Ma [13] are the initiators of the Rosenbrock Artificial Bee Colony Algorithm (RABC). This method is proposed to improve the exploitation process of the original ABC. The researchers modified the Rosenbrock's rotational direction method (RM) at the termination criteria of the two loops and the step sizes of the RM. The step sizes are not reset after orthonormal basis is updated and this is the intention to reduce the number of iterations needed to reset the step size at every stage.

After the modification of the original RM, the modified RM is added into the original ABC as an exploitation tool. The rank-based fitness transformation is adopted to replace the original fitness equation from ABC:

$$fit_i = 2 - SP + \frac{2(SP-1)(r_i - 1)}{NS} - 1 \quad (3.9)$$

Where, r_i is the position solution i in the entire population after ranking, $SP \in [1.0, 2.0]$ is the selection pressure. The appropriate value of *SP* will be 1.5.

Therefore, there are two phases in RABC, which are the exploration phase from the ABC and exploitation by the RM. Results shown that, RABC have reliable performances, whereby it has demonstrated strong competitive capabilities in terms of robustness, efficiency and accuracy by comparing with others algorithm including ABC [13].

In year 2012, a simulated annealing based artificial bee colony algorithm (SAABC) is created. This method makes use of the idea from annealing process of solids which is a process of heating solids at a very high temperature and cooling it gradually to allow crystallize. The experimental results shows that SAABC able to outperform ABC and Gbest ABC in most of the experiments. ABC exhibits the slowest convergence rate to locate a local optimal, GABC and SAABC have the similar convergence speed, yet, GABC locates a local optimal whereas the SAABC locates the global optimal result [14]. In general, SAABC often offers the most robust solutions compare to the results obtained from ABC and GABC.

Bahriye Akay together with Dervis Karaboga [15], the first who introduce the original artificial bee colony algorithm, had proposed a modified artificial bee colony algorithm for real-parameter optimization in year 2012. They modified the original ABC algorithm by introducing a control parameter, modification rate (MR). A uniformly distributed random number $(0 \le R_{ij} \le 1)$ is produced for each parameter of x_{ij} and if the random number is less than MR, then the equation of the model is modified as follow:

$$v_{ij} \begin{cases} x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) & ,if\ R_{ij} < MR, \\ x_{ij} & ,Otherwise \end{cases} \quad (3.10)$$

Besides, there is another modification which is about the ratio of the variance operator of the original ABC algorithm. A scaling factor (*SF*), *[-SF, SF*] is added to replace ϕ_{ij}, [-1, 1] in basic ABC algorithm which acts as the control parameter for the random perturbation $(x_{ij} - x_{kj})$. The *SF* value

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

is set before running the algorithm and it may change automatically during the search because the large *SF* causing slow convergence while small *SF* reduces the capability of the exploitation process. This is called the adaptive *SF* (ASF), which is conducted by using the Rechenberg's 1/5 mutation rule which states that the ratio of successful mutations to all mutations should be 1/5 [16]. Therefore, the step size is changing according to 1/5 rule in every cycles.

$$SF(t+1)=\begin{cases} SF(t)*0.85 & if\ \varphi(m)<1/5, \\ \dfrac{SF(t)}{0.85} & if\ \varphi(m)>1/5, \\ SF(t) & if\ \varphi(m)=1/5. \end{cases} \quad (3.11)$$

From the experimental results, the modified ABC algorithm produces promising results on hybrid functions compared to state-of-the-art algorithms where the original ABC algorithm can efficiently solve basic functions [15].

After the introduction of global best ABC by Guopu Zhu and Sam Kwang [12], Weifeng Gao and fellow researchers did the similar modification as the Gbest ABC [17]. The differences is the initial population and scout bees are generated by combining chaotic systems with opposition-based learning method and the solution search is based on that each bee searches only around the best solution of the previous iteration to improve the exploitation.

After that, Weifeng Gao and Sanyang Liu together designed a modified aritificial bee colony in the same year. In this algorithm, they proposed a new framework without probabilistic selection scheme and scout bee phase [18]. In addition, they combine the chaotic systems with opposition-based learning method to generate the initial population. This method enhanced the global convergence. Therefore, both the global best ABC and modified ABC by Weifeng Gao with fellow researchers outperforms the original ABC and GABC algorithms.

Chen et al. [19] reported that the generation of scout bees from a standard initial population provides strong diversity but may deprive of solution quality. Therefore, they proposed an improved ABC algorithm that provides a balance between exploration and exploitation. Compared results showed that, improved ABC algorithm outperformed others metaheuristics approaches and original ABC in terms of diversity, convergence, and effectiveness. Bacanin et al. [20] applied ABC to constrained portfolio optimization problem with an efficient constraint handling method. They compared ABC algorithm with GA and pointed out ABC algorithm's potential on effectively solving portfolio optimization problems. Suthiwong and Sodanil [21], to improve the exploitation capability of employed bee phase, proposed an ABC algorithm inspired by PSO that takes advantage of the information of the global best solution to guide the search. Ge [22] proposed another promising ABC algorithm that outperforms standard ABC. Kumar and Mishra [23] proposed a powerful co-variance guided ABC algorithm for portfolio optimization with cardinality constraints and investment limit constraints and tested on benchmark data sets from OR-library confirming its capability of handling real life portfolio management tasks. Kalayci et al. [24] proposed a novel methodology based on ABC algorithm, confirming its superior performance on benchmark data sets from OR-Library, with feasibility enforcement and infeasibility toleration procedures that handles boundary constraints and cardinality constraints efficiently.

4. ABCED CONJUGATE GRADIENT

Artificial Bee Colony algorithm (ABC) is inspired by the foraging behaviors of honey bee swarm. In fact, bees are divided into two groups: employed and unemployed (onlookers and scouts). At first, the bees will be searching for a food source and become employed bee when the bee manage to bring the nectars back to their hives. The employed bees can either go back to her discovered source site or spread it to the onlookers by performing a dance on dancing area. The onlookers will select one profitable source by watching the dance advertising according to the quality of the source. When a source is exhausted or abandoned, the employed bees will become a scout and start to randomly search for a new source [15].

This mechanism is applied in ABC algorithm. First of all, randomly distributed initial food source positions, which generated by the objective function values of the sampled point from each employed bee. The process can be represented by $f(x_i),\ x_i \in R^D, i \in \{1,2,3,...,SN\}$. x_i is a position of food source as D-dimensional vector, where D is the number of optimization parameter in the model. $f(x_i)$ is the objective function which determines the quality of the solution, and *SN* is the number of food sources. After the initialization, the population is subjected to repeated cycles of the three major steps that are updating feasible solutions, selecting feasible solutions and avoiding suboptimal solutions (Anan Banharnsakun, 2011). In order to test the fitness value, fit_i of the new food source, the employed bee could produce a modification on the solution in its memory.

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) \quad (4.1)$$

In equation (3.1), v_{ij} is the new feasible solution and *SN* is the size of food source which generated, ϕ_{ij} is the random number between [0, 1] which is used to randomly adjust the old solution become the new solution in the next iteration. $k \in \{1, 2, \dots, SN\}$ and $j \in \{1, 2, \dots, D\}$ are random generated indexes but *k* must different from *i*. When the different between x_{ij} and x_{kj} decreases, the perturbation of the position also decreases. Thus, the step size reduced when the search approaches to the optimum solution in the search space [2].

$$fit_i = \begin{cases} \dfrac{1}{1+f(x_i)} & if\ f(x_i) \geq 0 \\ 1+abs(f(x_i)) & if\ f(x_i) < 0 \end{cases} \quad (4.2)$$

The fitness value is proportional to the nectar amount of the food source in the *i*th position. If the fitness value is better than the previous one, the employed bee would memorize the new food position and forget the old one. Otherwise, it keeps the current food position in its memory. Information about nectar amount and positions of food will be shared by the employed bees when all of them completed the searching process to the onlookers. The onlookers will then evaluate the nectar information by all the employed bees and chooses a food source according to the probability which is related to the nectar amount. Therefore, during onlooker bees phase, new solution v_{ij} is produced for the solutions *x* by means of their fitness values by using the formula of the probability of the fitness. The onlookers can produce a modification on the position in its memory as what employed bees do. The onlookers check the nectar amount of the candidate source. If the nectar amount is better than that of the previous one, the bee would memorize the new position instead of the previous one. An onlooker chooses a food source completely according to the probability value associated with the food source, p_i where fit_i, the fitness value of the i^{th} solution is:

$$p_i = \frac{fit_i}{\sum_{j=1}^{SN} fit_j} \quad (4.3)$$

The food source which is exhausted or abandoned by the bees would be replaced with a new food source found by scout bees. The function values will be identified as abandoned values when they had been undergo a specific number of trials and the solutions cannot be improved. Then, a new solution will be generated randomly to replace with the abandoned one. The new random position chosen will be calculated by using the equation below:

$$x_i^j = x_{\min}^j + rand(0,1)(x_{\max}^j - x_{\min}^j) \quad (4.4)$$

where $x_{\max}^j$ is the upper bound of the food source position in dimension *j* while $x_{\min}^j$ is the lower bound of the food source position in dimension *j*. The boundaries act as one of the constraints of the algorithm. When is parameters generated exceeds the boundaries, they will be shifted onto the boundaries. Besides that, *rand(0,1)* represents the random number between [0, 1] and the maximum number of cycle (*MCN*) is used to control the number of iterations and it acts as a termination criterion.

4.1 Algorithm of ABCED Conjugate Gradient Method

Initialize the population of solutions x_{ij} , *i=1, 2...SB*, *j= 1, 2...n*, *trial= 0* is the non-improvement number of the solution x_{ij} , used for abandonment
Evaluate the population
Set Cycle= 1
Repeat
  {Produce a new food source population for employed bees}
  for *i*=1 to *SN* **do**
    Produce a new food source v_i for the employed bee of the food source x_i with **algorithm 2.1**

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

```
            Apply a greedy selection process between v_i and x_i and select the better one.
            If solution x_i does not improve trial_i = trial_i + 1, otherwise, trial_i + 1=0
    end for
    Calculate the probability values p_i by equation (4.3) with the fitness values in equation (4.2)

    {Produce a new food source population for onlooker bees}
    t=0, i=1
    repeat
        if random<p_i then
                Produce a new v_i food source by algorithm 2.1 for onlooker bee
                Apply a greedy selection process between v_i and x_i and select the better one
                If solution x_i does not improve trial_i = trial_i + 1, otherwise trial_i = 0, t= t+1
        end if
    until (t=SN)

    {Determine scout}
        if max (trial_i)> limit then
                Replace x_i with a new randomly produced solution by
                    x_i^j = x_min^j + rand(0,1)(x_max^j - x_min^j)
        end if
        Memorize the best solution achieved so far
        cycle= cycle+1
until (cycle= Maximum Cycle Number)
```

5. NUMERICAL RESULTS

The algorithm ABCED Conjugate Gradient Methods with 5 different of popular conjugate direction equations have been programmed into C++ and tested to the selected 21 global optimization problems as listed in Table 5.1. The numerical results have been presented in Table 5.2. Since the ABCED Conjugate Gradient methods is hybridized from a Metaheuristic approach, each problems have been run for 20 times. Then, only the mean value of the global optimum takes into counter. This is to make sure the efficiency of the method won't be misjudging for some misconduct in its heuristic behavior. The numerical results show that not all ABCED Conjugate Gradient Methods are able to globally solve all the selected global optimization problems. In the Table 5.2, those results shaded mean that method globally solved that particular problem. According to the numerical results, it seem that ABCED Conjugate Gradient with Fletcher-Reeve (ABCED-CGFR) equation globally solved 80.95% of the selected global optimization problems possess the best performance among five different conjugate direction equations. Another two ABCED Conjugate Gradient Methods with Hestenes-Stiefel equation (ABCED-CGHS) and Dai Yuan's equation (ABCED-CGDY) both globally solved 76.19% of the tested global optimization problems, retain as the second-best performer among five selected ABCED Conjugate Gradient methods. However, the performance of ABCED-CGHS in several global optimization problems indicated 'Fail' in Table 5.2 is because the it totally fails to locate even one global optimum in its 20 runs. So, ABCED-CGDY should be the second best of ABCED Conjugate Gradient Methods.

The rest of the failure that still recorded their global optimum value is because among 20 runs, the respective method still able to obtain at least 5 global optimums out of 20 runs. Then, those shaded boxes are obtained at least more than 18 global optimums out of 20 runs.

Among all the selected ABCED Conjugate Gradient methods, the one using Polak-Ribiere's equation is the worst performer. It's only able to globally solve one third of the tested global optimization problems. Lastly, the ABCED Conjugate Gradient method with proposed by Fletcher (ABCED-CGCD) able to globally solve two third of the tested global optimization problems.

Table 5.1 List of the Benchmark Multimodal Optimization Problems

No.	Problem
1	Griewank $f(x)=1+\frac{1}{4000}\sum_{i=1}^{5}x_i^2-\prod_{i=1}^{5}\cos\left(\frac{x_i}{\sqrt{i}}\right)$
2	Sphere $f(x)=\sum_{i=1}^{5}\left[x_i^2\right]$
3	Rosenbrock $f(x)=\sum_{i=1}^{4}\left[100(x_{i+1}-x_i^2)^2+(x_i-1)^2\right]$
4	Rastrigin function (5 variables) $f(x)=50+\sum_{i=1}^{5}\left[x_i^2-10\cos(2\pi x_i)\right]$
5	Rastrigin function (2 variables) $f(x)=20+\sum_{i=1}^{2}\left[x_i^2-10\cos(2\pi x_i)\right]$
6	Ackley's function $f(x_1,x_2)=-20\exp\left[-0.2\sqrt{0.5(x_1^2+x_2^2)}\right]-\exp\left[0.5\left(\cos\left(2\pi x_1\right)\cos\left(2\pi x_2\right)\right)\right]+e+20$
7	Beale's function $f(x_1,x_2)=\left(1.5-x_1+x_1x_2\right)^2+\left(2.25-x_1+x_1x_2^2\right)^2+\left(2.625-x_1+x_1x_2^3\right)^2$
8	Goldstein–Price function $f(x_1,x_2)=\left[1+\left(x_1+x_2+1\right)^2\left(19-14x_1+3x_1^2-14x_2+6x_1x_2+3x_2^2\right)\right]\left[30+\left(2x_1-3x_2\right)^2\left(18-32x_1+12x_1^2+48x_2-36x_1x_2+27x_2^2\right)\right]$
9	Booth's function $f(x_1,x_2)=(x_1+2x_2-7)^2+(2x_1+x_2-5)^2$
10	Matyas function $f(x_1,x_2)=0.26(x_1^2+x_2^2)-0.48x_1x_2$
11	Lévi function N.13 $f(x_1,x_2)=\sin^2\left(3\pi x_1\right)+\left(x_1-1\right)^2\left(1+\sin^2\left(3\pi x_2\right)\right)+\left(x_2-1\right)^2\left(1+\sin^2\left(2\pi x_2\right)\right)$
12	Three-hump camel function $f(x_1,x_2)=2x_1^2-1.05x_1^4+\frac{x_1^6}{6}+x_1x_2+x_2^2$
13	Easom function $f(x_1,x_2)=-\cos(x_1)\cos(x_2)\exp\left(-\left((x_1-\pi)^2+(x_2-\pi)^2\right)\right)$
14	Adjiman function $f(x_1,x_2)=\cos(x_1)\sin(x_2)-\frac{x_1}{x_2^2+1}$
15	bird function $f(x_1,x_2)=\sin(x_1)e^{(1-\cos(x_2))^2}+\cos(x_2)e^{(1-\sin(x_1))^2}+(x_1-x_2)^2$
16	Bohachevsky 1 Function $f(x_1,x_2)=x_1^2+2x_2^2–0.3\cos(3\pi x_1)-0.4\cos(4\pi x_2)+0.7$
17	Bohachevsky 2 Function $f(x_1,x_2)=x_1^2+2x_2^2–0.3\cos(3\pi x_1)\cdot 0.4\cos(4\pi x_2)+0.3$
18	Bohachevsky 3 Function $f(x_1,x_2)=x_1^2+2x_2^2–0.3\cos(3\pi x_1+4\pi x_2)+0.3$
19	Branin RCOS function 1 $f(x_1,x_2)=\left(x_2-\frac{5.1x_1^2}{4\pi^2}+\frac{5x_1}{\pi}-6\right)^2+10\left(1-\frac{1}{8\pi}\right)\cos(x_1)+10$
20	Branin RCOS function 2 $f(x_1,x_2)=\left(x_2-\frac{5.1x_1^2}{4\pi^2}+\frac{5x_1}{\pi}-6\right)^2+10\left(1-\frac{1}{8\pi}\right)\cos(x_1)\cos(x_2)\ln(x_1^2+x_2^2+1)+10$
21	Bukin 2 function $f(x_1,x_2)=100(x_2-0.01x_1^2+1)+0.01(x_1+10)^2$

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 5.2 Numerical Results of ABCED Conjugate Gradient Methods

No	Problem	CGDY	CGCD	CGFR	CGPR	CGHS
1	Griewank function	0.03169107	0.01299987	0.01899957	0.07204749	0.004241837
2	Sphere function	4.130516e-012	1.534866e-012	2.167668e-012	1.037357	1.339346e-012
3	Rosenbrock	5.289350e-7	0.01534409	2.073671e-006	2.207111	Fail
4	Rastrigin function (5 variables)	4.521959	5.621514	6.865650	11.26486	5.024540
5	Rastrigin function (2 variables)	1.072989e-009	9.949591e-002	5.771485e-002	8.337036e-001	9.949591e-002
6	Ackley's function	2.934435e-004	4.976581e-003	6.544239e-002	5.390010e-001	6.146079e-006
7	Beale's function	3.306578e-011	3.292605e-004	2.892012e-010	4.361799e-003	Fail
8	Goldstein–Price function	3.000000	3.000000	3.000000	3.098475	Fail
9	Booth's function	1.500000e-001	2.075949e-010	1.557476e-011	2.964338e-002	2.140854e-011
10	Matyas function	7.500000e-003	4.528473e-006	3.579961e-012	1.373807e-009	1.842513e-010
11	Lévi function N.13	3.956416e-010	5.923011e-006	7.659820e-005	3.411834e-002	3.686733e-009
12	Three-hump camel function	2.251716e-011	2.961625e-007	3.560178e-012	1.039893e-002	1.112625e-011
13	Easom function	-1.000000	-1.000000	-1.000000	-0.9983190	Fail
14	adjiman function	-4.951813	-5.053946e	-4.814400e	-4.896541e	-5.004011
15	bird function	-107.0121	-107.0172	-106.7645	-105.5517	-106.7645
16	Bohachevsky 1 Function	5.946584e-011	1.407258e-010	4.315649e-009	9.405638e-002	1.016121e-010
17	Bohachevsky 2 Function	0.18	0.18	0.18	0.2579728	0.18
18	Bohachevsky 3 Function	0.009	0.009001557	5.219617e-010	0.03441181	1.061773e-010
19	Branin RCOS function 1	0.3978874	0.3983374	0.3978874	0.4412183	0.3978874
20	Branin RCOS function 2	-9.538853	-9.558770	-9.558770	-9.235987	-9.558770
21	Bukin 2 function	-412.8628	-369.9772	-370.8660	-371.2948	-424.1500
	Percentage of Success	**76.19%**	**66.67%**	**80.95%**	**33.33%**	**76.19%**

6. CONCLUSION AND DISCUSSION

In this paper, the enhancements had produced a new gradient-based method called ABCED Conjugate Gradient Methods. The numerical results show that not all ABCED Conjugate Gradient Methods are able to globally solve all the selected global optimization problems.

In the Table 5.2, those results shaded mean that method globally solved that particular problem. According to the numerical results, it seem that ABCED Conjugate Gradient with Fletcher-Reeve (ABCED-CGFR) equation globally solved 80.95% of the selected global optimization problems possess the best performance among five different conjugate direction equations. Another two ABCED Conjugate Gradient Methods with Hestenes-Stiefel equation (ABCED-CGHS) and Dai Yuan's equation (ABCED-CGDY) both globally solved 76.19% of the tested global optimization problems, retain as the second-best performer among five selected ABCED Conjugate Gradient methods. However, the performance of ABCED-CGHS in several global optimization problems indicated 'Fail' in Table 5.2 is because the it totally fails to locate even one global optimum in its 20 runs. So, ABCED-CGDY should be the second best of ABCED Conjugate Gradient Methods.

The rest of the failure that still recorded their global optimum value is because among 20 runs, the respective method still able to obtain at least 5 global optimums out of 20 runs. Then, those shaded boxes are obtained at least more than 18 global optimums out of 20 runs.

Among all the selected ABCED Conjugate Gradient methods, the one using Polak-Ribiere's equation is the worst performer. It's only able to globally solve one third of the tested global optimization problems. Lastly, the ABCED Conjugate Gradient method with proposed by Fletcher (ABCED-CGCD) able to globally solve two third of the tested global optimization problems.

ACKNOWLOEDGEMENT

The authors are grateful to the Center of Excellences (CoEs) and Centre for Research and Innovation Management (CRIM), Universiti Teknikal Malaysia Melaka (UTeM) for providing financial assistance for this research work.

REFRENCES:

[1] Karaboga, D. "An Idea Based on Honey Bee Swarm for Numerical Optimization", Computer Engineering, Oct 2015, p. 10.
[2] Li, G., Niu, P., and Xiao, X. "Development and Investigation of Efficient Artificial Bee Colony Algorithm For Numerical Function Optimization", Applied soft computing, 2012, vol. 12, pp. 320-332.
[3] Banharnsakun, A., Achalakul,T. and Sirinaovakul,B. "The Best-so-far Selection in Artificial Bee Colony Algorithm", Applied

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Soft Computing, 2011, vol. 11, pp. 2888-2901.

[4] Zhu,G.P., Kwong,S. "Gbest-guided artificial bee colony algorithm for numericl function optimization", Appied Mathematics and Computation, 2010, vol. 217, pp. 3166-3173.

[5] Goh Khang Wen, Yosza Bin Dasril and Abdul Rani Bin Othman, "Global Optimization With Simplexed Artificial Bees Colony Algorithm", Far East Journal of Mathematical Sciences, 2018, 103 (8), 1303-1321

[6] Nocedal, J. & Wright, S.J. "Numerical optimization: Springer Series in Operations Research". 1999, New York: Springer-Verlag.

[7] Frandsen, P. E., Jonasson, K., Nielsen, H. B. & Tingleff, O., "Unconstrained Optimization 3rd edition, Informatics and Mathematical Modelling Technical", 2004, University of Denmark, Denmark.

[8] Joshi, M.C. & Moudgalya, K.M. "Optimization: Theory and Practice. Harrow", U.K.: 2004, Alpha Science International Ltd.

[9] Yosza, B. D., Goh, K. W., Ismail, B. M. and Hendra, S., "Avoid Oscillation of Steepest Descent Method by Using Fixed Range of Step Size", Proceeding of the National Seminar on Mathematics and Society 2008, pp. 389-400.

[10] Hestenes, M.R. and Stiefel, E., "Methods of Conjugate Gradients for Solving Linear Systems". Journal of Research of the National Bureau of Standards, 1952, 49, 409-435.

[11] Tereshko,V. and Loengarov,A. "Collective Decision-Making in Honey Bee Foraging Dynamics", 2005, School of Computing, Oct, pp. 1-7.

[12] Zhu,G.P., Kwong,S. "Gbest-guided artificial bee colony algorithm for numericl function optimization", 2010, Appied Mathematics and Computation, vol. 217, pp. 3166-3173.

[13] Kang,F., Li,J. and Ma,Z. "Rosenbrock Aritificial Bee Colony Algorithm for Accurate Global Optimization of Numerical Functions", Information Sciences, 2011, vol. 181, pp. 3508-3531.

[14] Chen,S.M., Sarosh,A. and Dong,Y.F. "Simulated Annealing Based Artificial Bee COlony ALgorithm for Global Numerical Optimization", 2012, Applied Mathematics and Computation, vol. 219, pp. 3575-3589.

[15] Akay,B. and Karaboga.D, "A modified Aritificial Bee Colony Algorithm for Real-Parameter Optimization", Information Sciences, 2012, vol. 192, pp. 120-142.

[16] Mechalewics,Z. and Schoenauer,M. "Evolutionary Algorithm, Encyclopedia of Information Systems", 2003, vol. 2, pp. 259 - 267.

[17] Gao,W.F., Liu,S. and Huang,L. "A global best artificial bee colony algorithm for global optimization", journal of computational and applied mathematics, 2012. vol. 236, pp. 2741-2753.

[18] Liu, Gau.W.F., and Sanyang. "A modified aritifical bee colony algorithm", Computers and Operational Research, 2012. vol. 39, pp. 687-697.

[19] Chen A.H.L., Liang Y.C, Liu C.C., "Portfolio optimization using improved artificial bee colony approach", Proceedings of the 2013 IEEE Conference on Computational Intelligence for Financial Engineering and Economics, CIFEr 2013-2013 IEEE Symposium Series on Computational Intelligence, SSCI 2013, pp. 60-67.

[20] Bacanin N., Tuba M., Pelevic B., "Constrained portfolio selection using artificial bee colony (ABC) algorithm", Int. J. Math. Models Methods Appl. Sci., 2014. 8, pp. 190-198.

[21] Suthiwong D. and Sodanil M., "Cardinality-constrained portfolio optimization using an improved quick artificial bee colony algorithm", Computer Science and Engineering Conference (ICSEC), 2016 International, pp. 1-4.

[22] Ge M. , "Artificial bee colony algorithm for portfolio optimization", 5th International Conference on Intelligent Control and Information Processing, ICICIP 2014-Proceedings 2015, pp. 449-453.

[23] Kumar D., Mishra K.K., "Portfolio optimization using novel co-variance guided Artificial Bee Colony algorithm", Swarm Evolut. Comput., 2017. 33, pp. 119-130.

[24] Kalayci C.B., Ertenlice O., Akyer H., Aygoren H., "An artificial bee colony algorithm with feasibility enforcement and infeasibility toleration procedures for cardinality constrained portfolio optimization", Expert Syst. Appl., 2017. 85 (Supplement C), pp. 61-75.