Hybrid Evolutionary Clonal Selection for Parameter Estimation of Biological Model

Afnizanfaizal Abdullah[1], Safaai Deris[2], Sohail Anwar[3]

[1,2]*Artificial Intelligence and Bioinformatics Group (AIBIG)*
[1,2]*Faculty of Computer Science and Information Systems, Universiti Teknologi Malaysia, 81310 UTM, Johor*
[3]*Pennsylvania State University, 3000 Ivyside Park Altoona, Pennsylvania, USA*

[1]afnizanfaizal@utm.my
[2]safaai@utm.my
[3]sxa15@psu.edu

***Abstract*— The Clonal Selection Algorithm (CSA) is a widely used Artificial Immune Optimization (AIO) approach that tends to mimic the immune response when the pathogenic pattern is detected by the immune cells. However, this method, in its standard form, shows slow convergence and frequently traps in one of the local optima, especially for high dimensional problems. Hence, in this paper, an improved CSA method is introduced by integrating evolutionary operations adopted from the Differential Evolution (DE) method. The proposed method, called Differential Clonal Evolution (DICE) method, utilizes the mutation and crossover operation to exploit the information of different antibodies in the population. Furthermore, antibodies that yield trivial fitness value are relocated randomly so that the method can escape from the local optima in a more straightforward manner. To show the effectiveness of this method, the method is used to estimate parameters of a bacterial lactose production model using noisy and incomplete time series data. The statistical results suggest that the proposed method has better speed and accuracy performance compared to the standard CSA, Particle Swarm Optimization (PSO) and Genetic Algorithm (GA) techniques.**

***Keywords*— Clonal Selection; Differential Evolution; Hybrid Optimization; Parameter Estimation; Bacterial Model**

I. INTRODUCTION

For the past few years, global optimization problems have received significant attention, which led to the implementation of a variety of optimization methods [1]. Among these methods, stochastic population-based approach offers a number of advantages including freedom from derivative constraints, improved accuracy and robustness performance, as well as, utilizing a wide range of search space [6, 8]. Due to these reasons, studies have been carried out to solve many optimization problems in both scientific and industrial fields. As a result, many varieties of optimization methods are proposed, including Particle Swarm Optimization (PSO) [2], Genetic Algorithms (GA) [3], Ant Colony Optimization (ACO) [4], and Artificial Bee Colony (ABC) [5].

However, depending on single method can be very restrictive for certain problems. This is due to the fact that every method has its limitations, especially in terms of searching accuracy and convergence speed. Several recent studies have shown that the hybridization of different methods may considerably improve the searching capability [6-10]. In particular, the hybridization usually overcomes the limitations of the standard methods by exploiting the advantages of the other methods [6]. Hence, this provides a promising opportunity to enhance the accuracy and speed performance of the standard methods.

The Clonal Selection Algorithm (CSA) method is one of the most widely used Artificial Immune Optimization (AIO) approaches [12, 14]. The method is basically motivated by the concept of clonal selection principle that describes the immune response when the pathogenic pattern is identified [15]. Despite of the advantage to perform good approximation for finding global optimum in multimodal problems [16], the main shortcoming of this method consists of premature and slow convergence. For the problem of prematurity, the CSA method usually fails to explore new possible solutions. This is main reason is that the method got trapped in one of the local optima. On the other hand, it is shown that the CSA method frequently converges slowly, particularly when searching for better solutions in high dimensional problems.

Recently, many research investigations have been made to overcome these limitations. They include the introduction of elimination feature to remove the oldest candidate solutions [17] and employ the chaos-based mutation strategy [20] for improving the diversity of the possible candidate solutions. More recently, local search technique [18] and immune memory encoding [19] are incorporated into the the standard CSA method to enhance the exploitation of the population. Alternatively, the evolutionary algorithms such as Differential Evolution (DE) [13] method have been used to improve

the searching capability of the CSA method. The evolutionary operations of the DE method are commonly utilized to enhance the proliferation process in the CSA method, thereby substantially utilizing the information regarding the adjoining clones [15-16].

In this work, the research that is relevant to the improvement of the searching capability of standard CSA method is extended by using the evolutionary operations of DE method. In this variant of the CSA method, the crossover and mutation operation are implemented to exploit the information regarding different antibodies in the population. Simultaneously, the antibodies providing insignificant solutions are relocated randomly to enhance the fitness values. By doing so, the method can efficiently improve the searching quality as well as utilizing the computational time. The effectiveness of the proposed method is tested to estimate parameter values in a biological model and the statistical results are then compared with the standard CSA, PSO and GA methods. The rest of the paper is organized as follows. Sections II introduce the standard CSA, standard DE methods, and the proposed Differential Clonal Evolution (DICE) method. Subsequently, Section III presents the experimental results. Section IV discusses the contribution of the work and Section V presents the conclusion and future works.

II. METHODS

A. Standard Clonal Selection Algorithm (CSA) Method

The clonal selection principle [21] describes the reaction of immune system to pathogens and the process of improving the capability to identify these unintended agents. In particular, the theory illustrates that a number of immune cells that identify the pathogens will proliferate. Some of them will become the effector cells while the others maintain their role as memory cells [18]. In general, the CSA method employs three main phases: cloning, mutation and selection. The method starts with a population of *d*-dimensional search vectors, called antibodies. The *i*th antibody, *X*, of the whole population at a specific generation *t* is given by:

$$X_{i(t)} = \{x_{i1(t)}, x_{i2(t)}, \dots, x_{id(t)}\} \quad (1)$$

In CSA method, the fitness value of each antigen is represented as affinity, which implies the goodness of the antibody to generate antigen for the specific pathogen.

Initially, the population of antibody is initiated randomly and the affinity of each antibody is evaluated. The antibodies that produce good affinity values are selected to undergo cloning phase. As a result, a new set of population is created. Next, the mutation process is performed to every clone, based on the mutation constant. Hence, the mutated clones are formed with new components and the affinity values are then been evaluated to measure the fitness. In the last phase, the mutated clones are selected to replace the original antibodies. Eventually, the population is built with the new improved antibodies. The overall procedure of the standard CSA method is outlined in Figure 1:

```
1: Begin
2: Initiate population, X
3: evaluate(X) // evaluate fitness of each antibody
4: While max number of generation is not met
5: A ← best(X) // Select m best antibodies
6: For i = 1 to m antibody
7:     B_i ← clone(A_i)        // cloning selected antibodies
8:     C_i ← hypermutate(B_i)  // mutate clones
9:     A_i ← select(A_i, C_i)  // select improved clones to
10:                               replace old antibodies
11: End For
12: X ← combine(A, X) //include improved best
13:                               antibodies to population
14: End While
15: End Begin
```

Fig. 1. The standard CSA

B. Standard Differential Evolution (DE) Method

The DE method is also a stochastic population-based optimization method. The method is proposed based on the evolutionary operations of the GA method [13]. Compare to GA, this method employs a mutation operation to produce a trivial chromosome from the original chromosome. Then, this trivial chromosome is crossovered with its original counterpart to generate an offspring chromosome. A simple selection operation is performed to select the chromosome with a better fitness value. In each generation, a range of search space is specified to find a good solution. Thus, at initial generation or $t = 0$, each chromosome is initialized, with a lower and an upper bound, x_j^L and x_j^U respectively [13]:

$$x_{ij(t=0)} = x_j^L + R \cdot (x_j^U - x_j^L) \quad (2)$$

where *R* is a random number generated between 0 and 1 and *j* is the dimension size.

In order to produce the trivial chromosome, V_i, the mutation operation is executed according to the differentiation of neighborhood chromosomes, given as following:

$$v_{ij(t+1)} = x_{best(t)} + F \cdot (x_{r1(t)} - x_{r2(t)}) \quad (3)$$

where $x_{best(t)}$ denotes the current best chromosome, *F* is the scaling factor, while $x_{r1(t)}$ and $x_{r2(t)}$ are randomly chosen chromosomes [13]. Using this chromosome, an

offspring chromosome, Y_i, is created by performing a crossover operation between the new and the parent chromosomes:

$$y_{ij(t)} = \begin{cases} v_{ij(t)} \; if \; R < CR \\ x_{ij(t)} \; Otherwise \end{cases} \quad (4)$$

where *CR* is the crossover constant and *R* is a random number between 0 and 1 [13]. As another population of chromosomes is produced, a selection operation is needed to keep the population size constant. The selection is performed based on the calculated fitness value of each chromosome:

$$X_{i(t+1)} = \begin{cases} Y_{i(t)} \; if \; f(Y_{i(t)}) \leq f(X_{i(t)}) \\ X_{i(t)} \; if \; f(Y_{i(t)}) > f(X_{i(t)}) \end{cases} \quad (5)$$

This implies that if the offspring chromosome produces a better fitness value, the current parent chromosome will be replaced. Otherwise, this chromosome will remain in the population of the next generation.

C. Differential Clonal Evolution (DICE) Method

In this paper, a new hybrid method is introduced based on the standard CSA method. The proposed method, that is, the Differential Clonal Evolution (DICE) method, employs the evolutionary operations of DE method to enhance the utilization of information from different clones [15-16]. In the standard CSA method, the standard mutation operation considers only the single clone and its original antibody. Conversely, in this new variant, the mutation and crossover operations are used to include the information of neighboring clones. Different to [15], the DICE method completely replaces the standard mutation of CSA method with the evolutionary operations of DE method. At the same time, DICE method differs with [16] as the proposed method exploits the antibodies that give poor fitness values. Thus, this provides a mechanism that permits the method to increase the possibility of escaping the local optima more effectively.

Firstly, the population of antibodies is initialized randomly and the affinity value of each antibody is evaluated. Then, the population is sorted and a number of *m* antibodies with potential affinity values are selected. These antibodies undergo the cloning phase. The mutation and crossover operations using Eq. 3 and Eq. 4 are performed to these clones and a new population of offspring antibodies is produced. Next, a selection operation is executed between the original antibody and its offspring using Eq. 5. Simultaneously, the antibodies that produced poor affinity values are chosen to endure a randomization process using Eq. 2. Then, these improved antibodies are combined with the selected antibodies to form a new population and the antibody that produces best affinity value is chosen as the current best antibody. The procedure is iterated until the maximum number of generations is met. The overall procedure of DICE method is outlined in Figure 2:

1: **Begin**
2: Initiate population, *X*
3: $evaluate(X)$ *// evaluate fitness of each antibody*
4: **While** max number of generation is not met
5: $X \leftarrow sort(X)$ *// sort antibodies*
6: $A_{best} \leftarrow best(X)$ *// Select m best antibodies*
7: $A_{poor} \leftarrow poor(X)$ *// Select n poor antibodies*
8: **For** *i* = 1 to *m* best antibody
9: $B_i \leftarrow clone(A_{best,i})$ *// cloning best antibodies*
10: **End For**
11: **For** *j* = 1 to *p* clones
12: $C_j \leftarrow mutate(B_i)$ *// DE mutation (Eq. 3)*
13: $D_j \leftarrow crossover(C_i)$ *// DE crossover (Eq. 4)*
14: $A_{best,i} \leftarrow select(A_{best,i}, D_j)$ *// selection (Eq. 5)*
15: **End For**
16:
17: *// Poor antibodies (randomize operation)*
18: **For** *i* = 1 to *n* poor antibody
19: $A_{poor,i} \leftarrow random(A_{poor,i})$ *//randomize (Eq. 2)*
20: **End For**
21: $X \leftarrow combine(A_{best}, A_{Poor})$ *//combine*
22: $G_{gbest} \leftarrow best(X)$ *//select current global best*
23: **End While**
24: **End Begin**

Fig. 2. The DICE algorithm

III. Experimental Results

The effectiveness of the proposed DICE method is tested using a biological model of *Escherichia coli* bacterium. The model is basically described as an interaction network of the regulation of induction in the lac operon in the bacterium [22]. The accuracy and speed performance of the DICE method are compared with the standard CSA, PSO and GA methods. Furthermore, statistical analyses are performed to measure the reliability of the proposed method compared to other methods.

A. Lac Operon Regulation Model

The networks of interacting biomolecules usually accomplish several fundamental functions in the cells. However, frequently, the processes are difficult to be extracted as the interactions commonly involve complex behaviors. Hence, these networks are reconstructed using mathematical modeling to represent the actual processes. Unfortunately, the modeling of such networks typically involves several parameters that explicitly represent the entire processes. To determine these parameters, the experimental data are usually fitted with model so that these parameters can be estimated computationally. Therefore, optimization methods are utilized to perform this parameter estimation procedure. Yildirim and

Mackey [22] introduced a mathematical model for the regulation of induction process in the lac operon that considered the dynamics of the permease enabling the internalization of several biomolecules such as lactose and β-galactosidase. The model is important for the observation of the conversion of lactose to allolactose, glucose and galactose; the allolactose interactions with the lac repressor; and the mRNA [22]. The model is formed through the following equations:

$$\frac{dB}{dt} = \alpha_B e^{-\mu\tau_B} M_{\tau_B} - \tilde{\gamma}_B B \quad (6)$$

$$\frac{dA}{dt} = \alpha_A B \frac{L}{K_L - L} - \beta_A B \frac{A}{K_A - A} - \tilde{\gamma}_A A \quad (7)$$

where *A*, *B* and *L* are the concentrations of allolactose, β-galactosidase and lactose, respectively; *M* is the mRNA translation; τ_B is time; α_B, α_A and β_A are the production rate constants; $\tilde{\gamma}_B$ and $\tilde{\gamma}_A$ are the loss rate constants; μ is the dilution rate constant; K_A and K_L are the equilibrium constants of allolactose and lactose, respectively [22]. Thus, in this work, the values of α_B, α_A, β_A, $\tilde{\gamma}_B$, $\tilde{\gamma}_A$, μ, K_A and K_L parameters are tended to be estimated. The experimental values of these parameters are given in Table 1 [22].

TABLE I
EXPERIMENTAL VALUES OF THE REGULATION MODEL

Parameter	Experimental Value
α_A	1.76×10^4 min^{-1}
α_B	1.66×10^{-2} min^{-1}
β_A	2.15×10^4 min^{-1}
$\tilde{\gamma}_A$	5.20×10^{-1} min^{-1}
$\tilde{\gamma}_B$	8.33×10^{-4} min^{-1}
μ	2.26×10^{-2} min^{-1}
K_A	1.95×10^{-3} M
K_L	9.70×10^{-4} M

In this work, the experimental data is obtained *in silico* by generating noisy and sparse version of the model data. Firstly, the model is simulated and the values at several randomly chosen time points are evaluated. Then, the Gaussian noise is added to the values so that it will simulate the measurement noise [23]. The model data and the generated noisy and sparse experimental data of β-galactosidase and allolactose are illustrated in Figure 3 and Figure 4, respectively.

B. Parameter Estimation

Generally, the parameter estimation problem is formulated in the following way. Suppose that a system is formed by the d-dimensional state variable, *x*, at time *t*, which is the distinctive solution of the initial value problem:

$$\dot{x_t} = f(x_t, t, p) \quad (8)$$

where *p* is the parameters [24]. So, let *y* signify the observation of experimental value, *i*, corresponding to the measurement, *j*, and represented by the following equation:

$$y_{ij} = h_j(x_t, p) + \sigma_{ij}\varepsilon_{ij} \quad (9)$$

where $\sigma_{ij} > 0$ and ε_{ij} is a Gaussian distributed random variable [24]. Thus, the parameter estimation problem of a biological system consists of finding the optimal parameter *p* such that the difference of the experimental data and the simulated data is minimized:

$$\min J = \sum_{i=0}^{n} \sum_{j=0}^{m} (y_{ij} - h_j(x(t_i; x_0, p), p)^2 \quad (10)$$

where $x(t_i; x_0, p)$ is the trajectory at time *t*, *n* is the total number of parameters and *m* is the total number of observed values [24].

The results obtained from the proposed method are compared with those from the standard CSA, PSO and GA methods. For each method, a population size of 50 particles or chromosomes is initiated and the maximum number of generations is set to 200. Furthermore, each method is executed 100 times independently to observe its reliability and consistency. Table 2 shows the average fitness values and the corresponding standard deviation for each method. In general, the proposed DICE method has outperformed the standard methods. Hence, the accuracy of the proposed method is better compared to the other methods, as the overall fitness value obtained is the lowest among those from the other methods.

TABLE II
ACCURACY AND SPEED PERFORMANCE

Method	GA	PSO	CSA	DICE
Average	3.72×10^{-3}	3.56×10^{-3}	4.64×10^{-4}	1.93×10^{-9}
Standard Deviation	3.07×10^{-3}	3.00×10^{-3}	7.94×10^{-4}	4.15×10^{-9}
Average Speed (second)	0.358	6.240	0.483	0.452

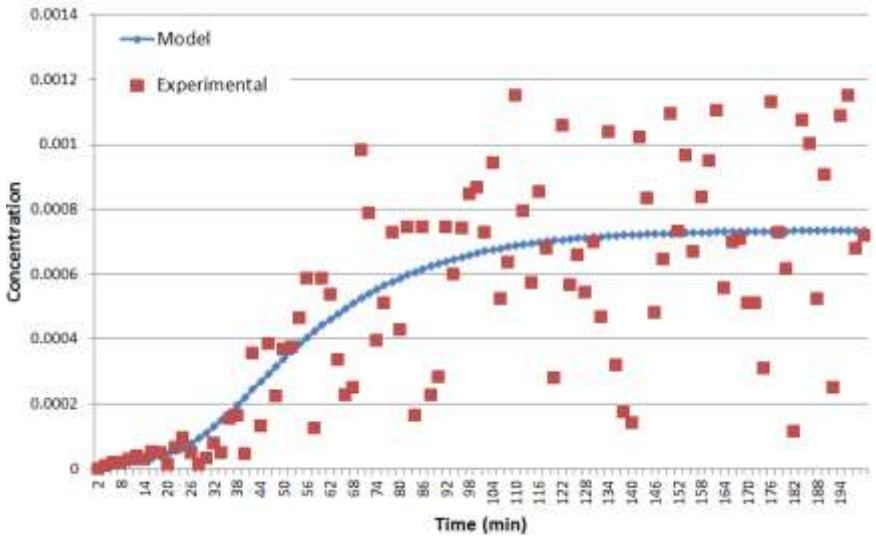

Fig. 3. Comparison of the model data and the experimental data for concentration of β-galactosidase

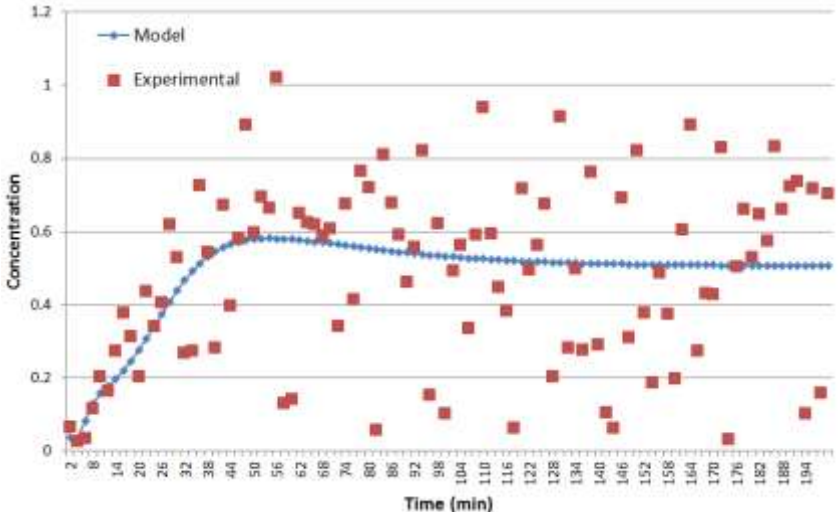

Fig. 4. Comparison of the model data and the experimental data concentration of allolactose

To address the performance of the proposed method in terms of convergence speed, Figure 5 illustrates the graph of convergence for all methods. Obviously, the standard GA and PSO methods converged prematurely while the standard CSA method successfully finds better fitness values compared to the GA and PSO methods. However, the method was eventually trapped in one of the local optima starting at the 165th generation. This problem has been effectively solved by the proposed method as the values are kept decreasing until the maximum number of generations is reached.

In addition, a statistical analysis of the observed measurements and the fitted data produced by the proposed DICE method is conducted. In this analysis, confidence interval estimates using chi-squared (χ^2) distribution is used. The result of this analysis is presented in Table 3. The result shows that the proposed method is reliable for the estimation of the parameter values as the mean error is substantially small for both components of the model. Moreover, the variance point lies between the interval estimates. Thus it is confirmed that the estimate obtained using the DICE method can be generally considered as valid.

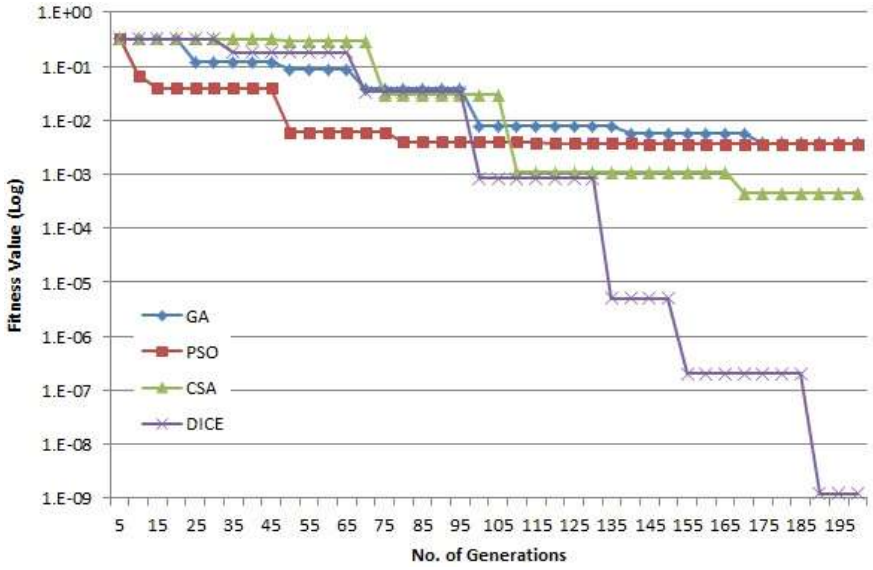

Fig. 5. Convergence behaviours of each method

Parameter estimation of complex biological models is usually presented as an optimization problem [23, 24]. The approximation of the parameter values is always hindered by the noise and incompleteness of the experimental data. Thus, optimization methods such as the GA and PSO methods have always been considered for this problem because they are capable of fitting the experimental data with the model prediction effectively. However, a substantial number of studies have shown that these methods are frequently trapped in one of the local optima [6]. Moreover, these methods always involve a huge search space that requires a large amount of computational time. Hence, a significant number of studies have been conducted to merge several methods to overcome this challenge [6-11]. Nevertheless, this approach shows potential in improving the accuracy and speed of the standard methods.

TABLE III
STATISTICAL ANALYSIS OF FITTED DATA BY DICE METHOD

Component	β-galactosidase	Allolactose
Error	0.21%	0.40%
Variance Point	4.65×10^{-8}	2.49×10^{-1}
Variance Interval	$[3.74\times10^{-8}, 6.26\times10^{-8}]$	$[2.00\times10^{-1}, 3.35\times10^{-1}]$
Real Variance	4.64×10^{-8}	2.48×10^{-1}
χ^2 Test	Pass	

IV. DISCUSSION

In this work, the proposed DICE method has presented another prospective alternative for enhancing the quality of the parameter estimation results. As shown in Table 2, the method has outperformed all the competitive methods efficiently, in terms of both, accuracy and speed performance. The accuracy performance of the proposed DICE method has shown remarkable improvement compared to the results produced by the other methods. This is because of the two main reasons. Firstly, the DICE method employs evolutionary operations to the antibodies that yield potentially good fitness values. As the operations are performed to these antibodies, the fitness values are improved significantly at each generation as the information regarding different antibodies is utilized to produce more significant fitness values. Secondly, the antibodies that produced insignificant fitness values are subjected to undergo randomization operation. By doing so, the method can enhance the fitness values of these antibodies, thus allowing the method to escape the local optima more effectively. This is shown by the convergence behavior of the DICE method in Figure 4.

Nonetheless, there is only a small difference of speed performance between the proposed method and its standard counterpart. This is due to the fact that the proposed method uses the computational time extensively for each antibody to exchange information between its neighbors. Even though finding the possible best values can be achieved more effectively, this requires numerous runtimes to execute the evolutionary operation on every antibody in the population. Hence, the scalability of the problem dimension may affect the speed performance of the method. However, statistical analysis performed on the results produced by the proposed DICE method show that the method is capable of estimating the parameter values accurately. The method passed the $\chi 2$ test, indicating that the values estimated by the proposed method are very close to the actual values.

V. CONCLUSION

Global optimization problems present a major challenge in both scientific and industrial fields. Thus, a significant number of optimization methods have been

developed to overcome these problems. In most cases, global optimization methods are always chosen due to the capability to handle nonlinearity of the problems. However, these methods are usually hampered by some limitations including huge computational time consumption and getting stuck in one of the local optima. This led to the development of hybrid optimization methods, which mainly tends to combine several different methods to improve the limitations by utilizing the advantages of the combined methods.

This paper presented a new hybrid optimization method based on the CSA method and the evolutionary operations adopted from the DE method. The effectiveness of the new method is tested using noisy and incomplete experimental data of a bacterial lactose production model. The results are compared to the standard CSA, PSO and GA methods. The comparison suggests that the accuracy and the speed performance of the proposed method are better than that can be obtained from other methods. Despite of this achievement, there are several limitations which need to be addressed. The computational time constitutes one such limitation. Hence, research is needed to overcome this challenge. The future research work may involve the improvement of the proposed DICE method through a use of local optimization approach and adaptive features. In addition, this study only considered one nonlinear model, which may ponder the restriction of the actual performance of the proposed method. Therefore, in the future, the performance of the method will be verified by using a number of different models to show the reliability and robustness of the method.

REFERENCES

[1] N. Noman and H. Iba, "Accelerating differential evolution using an adaptive local search," IEEE Transactions on Evolutionary Computation, pp.107-125, volume 12, no. 1, 2008.

[2] J. Kennedy and R. Eberhart, "Particle swarm optimization," *in Proc. IEEE International Conference on Neural Networks*, 1995, pp.1942-1948.

[3] D.E. Goldberg and J.H. Holland, "Genetic algorithms and machine learning", Machine Learning, pp. 95-99, volume 3, no. 2, 1988.

[4] M. Dorigo and G. Di Caro,"The ant colony optimization meta-heuristic," New ideas in optimization, pp.11-32, 1999.

[5] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm," Journal of Global Optimization, pp.459-471, volume 39, no. 3, 2007.

[6] S. Das, A. Abraham and A. Konar, "Particle swarm optimization and differential evolution algorithms: technical analysis, applications and hybridization perspective," Studies in Computational Intelligence, pp. 1-38, 2008.

[7] P. Kaelo and M.M. Ali, "Differential evolution algorithms using hybrid mutation," Computational Optimization Application, pp. 231-246, volume 37, no.2, 2007.

[8] S. Das, P. Koduru, M. Gui, M. Cochran, A. Wareing, S.M. Welch and B.R. Rabin, "Adding local search to particle swarm optimization," *in Proc. in IEEE Congress on Evolutionary Computation*, 2006, pp. 428-433.

[9] C. Zhang, J. Ning, S. Lu, D. Ouyang and T. Ding, "A novel hybrid differential evolution and particle swarm optimization algorithm for unconstrained optimization," Operational Research Letters, pp. 117-122, volume 37, 2009.

[10] Z.H. Zhan, J. Zhang, Y. Li, H.S.H. Chung, "Adaptive Particle Swarm Optimization," IEEE Trans. Syst. Man Cybr., pp.1362-1381, volume 39, no.6, 2009.

[11] W. Fu, M. Johnston, M. Zhang, "Hybrid particle swarm optimization algorithms based on differential evolution and local search," AI 2010:Adv. Artif. Intel., pp.313-322, 2010

[12] L.N. De Castro and F.J. Von Zuben, "The clonal selection algorithm with engineering applications," *in Proc. of GECCO'00, workshop on artificial immune systems and their applications*, volume 3637, 2000, pp. 36-37.

[13] R. Storn and K. Price, "Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces." J. Glob. Optim, pp.341–359 , volume 11, 1997.

[14] L.N.de Castro, F.J.von Zuben, "Learning and optimization using the clonal selection principle," IEEE Transactions on Evolutionary Computation, pp.239–251, volume 6, no. 3, 2002.

[15] X.Z. Gao, X. Wang and S.J. Ovaska, "Fusion of clonal selection algorithm and differential evolution method in training cascade-correlation neural network," Neurocomputing, pp. 2483-2490, volume 72, no. 10-12, 2009,

[16] M. Gong, L. Zhang, L. Jiao and W. Ma, "Differential immune clonal selection algorithm," *Int. Symp. on Intelligent Signal Processing and Communication Systems (ISPACS 2007)*, 2007, pp.666-669.

[17] V. Cutello, G. Nicosia and M. Pavone, "Real coded clonal selection algorithm for unconstrained global optimization using a hybrid inversely proportional hypermutation operator," *in Proc. ACM symposium on Applied computing*, 2006, pp. 950-954

[18] J. Yang, M. Gong, L. Jiao and Zhang, "Improved clonal selection algorithm based on lamarckian local search technique," *IEEE Congress on Evolutionary Computation (CEC 2008)*, 2008,pp.535-541

[19] W. Dong, G. Shi and L. Zhang, "Immune memory clonal selection algorithms for designing stack filters," Neurocomputing, pp.777-784, volume 70, no. 4-6, 2007.

[20] M. Gong, L. Jiao, L. Zhang and W. Ma, "Improved real-valued clonal selection algorithm based on a novel mutation method," *Int. Symp. on Intelligent Signal Processing and Communication Systems (ISPACS 2007)*, 2007, pp. 662-665.

[21] G.L. Ada and G.J.V. Nossal, "The Clonal Selection Theory," Scientific American, pp. 50-57, volume 257, no. 2, 1987.

[22] N. Yildirim and M.C. Mackey, "Feedback regulation in the lactose operon: a mathematical modeling study and comparison with experimental data," Biophysical Journal, pp.2841-2851, volume 84, no. 5, 2003.

[23] G. Lillacci and M. Khammash, "Parameter estimation and model selection in computational biology," PLoS Computational Biology, e1000696, volume 6, no.3, 2010.

[24] E. Balsa-Canto, M, Peifer, J.R. Banga, J. Timmer and C. Fleck:, "Hybrid optimization method with general switching strategy for parameter estimation, " BMC Systems Biology, pp.26-35, volume 2, no. 1, 2008.