



Optimization of fed-batch fermentation processes using the Backtracking Search Algorithm



Mohamad Zihin bin Mohd Zain^a, Jeevan Kanesan^{a,*}, Graham Kendall^b, Joon Huang Chuah^a

^a Department of Electrical Engineering, Faculty of Engineering, University of Malaya, Kuala Lumpur, 50603, Malaysia

^b School of Computer Science, University of Nottingham, Jubilee Campus, Nottingham NG8 1BB, UK

ARTICLE INFO

Article history:

Received 8 November 2016

Revised 19 June 2017

Accepted 22 July 2017

Available online 24 August 2017

Keywords:

Fed-batch fermentation

Backtracking Search Algorithm

Evolutionary algorithms

Wastewater treatment

Feeding trajectory optimization

Sewage sludge

ABSTRACT

Fed-batch fermentation has gained attention in recent years due to its beneficial impact in the economy and productivity of bioprocesses. However, the complexity of these processes requires an expert system that involves swarm intelligence-based metaheuristics such as Artificial Algae Algorithm (AAA), Artificial Bee Colony (ABC), Covariance Matrix Adaptation Evolution Strategy (CMAES) and Differential Evolution (DE) for simulation and optimization of the feeding trajectories. DE traditionally performs better than other evolutionary algorithms and swarm intelligence techniques in optimization of fed-batch fermentation. In this work, an improved version of DE namely Backtracking Search Algorithm (BSA) has edged DE and other recent metaheuristics to emerge as superior optimization method. This is shown by the results obtained by comparing the performance of BSA, DE, CMAES, AAA and ABC in solving six fed batch fermentation case studies. BSA gave the best overall performance by showing improved solutions and more robust convergence in comparison with various metaheuristics used in this work. Also, there is a gap in the study of fed-batch application of wastewater and sewage sludge treatment. Thus, the fed batch fermentation problems in winery wastewater treatment and biogas generation from sewage sludge are investigated and reformulated for optimization.

© 2017 Elsevier Ltd. All rights reserved.

1. Introduction

The diverse applications of optimization which range from manufacturing and engineering to business and medication have attracted many researchers to explore the field. Since the mid-20th century, researchers have developed a number of high performance optimization methods by taking inspiration from biology, physics, social and cultural behaviour, neurology and other disciplines. For instance, particle swarm optimization (PSO) (Kennedy & Eberhart, 1995) is a bio-inspired metaheuristics which is based on the metaphors of social interaction and communication (e.g., fish schooling and bird flocking). These algorithms are classified as a branch of optimization techniques called swarm intelligence metaheuristics. These metaheuristics use a process of trial and error to discover the solution of a problem and consists of certain trade-off of randomization and local search. They have a unique feature where more than one solution is evaluated simultaneously in a single iteration. Their most appealing characteristics are

their derivation-free mechanisms, relatively simple structures and stochastic nature. This enables faster convergence and less expensive computation as compared to deterministic method.

The field of biotechnology, which is considered as one of the important knowledge-based “economy” contains many problems that can take advantage of the optimization process by using metaheuristics. One such problem is the fermentation problem. In fermentation problem, the maximization of yield in a bioreactor is often regarded as the main goal. The yield efficiency is defined as the ratio of product against substrate. In the context of fed-batch fermentation, the timing and the amount of substrate input can directly affect the production of a bioreactor. As the complexity of the chemical reaction process is high along with high experimental cost, an automated system is needed to quickly calculate the optimal input profile that will optimize the yield. In order to obtain proper simulation of the process, usually differential equations that model the mass balances of various state variables are formulated. To this end, an expert system that combines swarm intelligence-based metaheuristics with simulation models of fed-batch fermentation problem is simplest yet effective in optimization of fed-batch problem.

In fermentation and bioprocess technology, the utilization of fed-batch operation is considered common. In biological

* Corresponding author.

E-mail addresses: mzihin_91@siswa.um.edu.my (M.Z.b. Mohd Zain), jievan@um.edu.my, jeevan4@yahoo.com (J. Kanesan), graham.kendall@nottingham.ac.uk (G. Kendall), jhchuah@um.edu.my (J.H. Chuah).

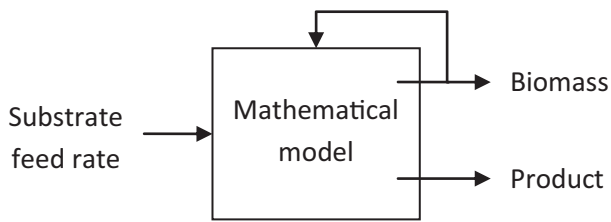


Fig. 1. Schematic illustration of a fed-batch fermentation process simulation.

wastewater treatment however, batch mode is still dominantly used and fed-batch is regarded as a relatively new technique (Montalvo et al., 2010). In a basic process of fed-batch wastewater treatment, the wastewater is fed slowly into the aerated bioreactor to reduce the chemical oxygen demand (COD) in the aeration tank. The disposal of sludge is one of the major problems in municipal wastewater treatment, and constitutes up to half of the operating costs of a Waste Water Treatment Plant (WWTP) (Baeyens, Hosten, & Van Vaerenbergh, 1997). Though different methods for sludge disposal exist, anaerobic digestion is one of the preferred routes (Appels, Baeyens, Degrève, & Dewil, 2008). The anaerobic digestion kinetics for methane fermentation of sewage sludge was proposed by Sosnowski, Klepacz-Smolka, Kaczorek, and Ledakowicz (2008). However, the proposed model was only designed for batch mode operation. Considering the advantages of fed-batch process in various fermentation problems, it is appropriate to convert this model into fed-batch mode. The utilization of fed-batch technique can increase the output of desirable products such as protein and bio-fuel in various fields of biotechnology and hence contribute to the development of renewable energy production and sustainable science.

The optimization of fed-batch fermentation process was intensively studied in recent years. Chen, Nguang, Chen, and Li (2004) proposed the optimization of a fed-batch bioreactor using a cascade recurrent neural network (RNN) model and modified genetic algorithm (GA). They applied their method in the fed-batch fermentation of a common yeast species in food technology, *Saccharomyces cerevisiae*. Levišauskas and Tekorius (2005) investigated various fed-batch fermentation processes optimization using the feed-rate time profile approximating functions and the parametric optimization procedure. In their work, four types of time functions namely constant feed-rate, ramp-type function, exponential function and a network of radial basis functions are compared. The parametric optimization problems were solved using chemotaxis random search algorithm. Liu, Gong, Shen, and Feng (2013) proposed a new nonlinear dynamical system to formulate the fed-batch fermentation process of glycerol bioconversion to 1,3-propanediol (1,3-PD). Peng et al. (2014) studied the fed-batch fermentation process of an antibiotic, iturin A using an artificial neural network-genetic algorithm (ANN-GA) and uniform design (UD).

In fed-batch fermentation simulation, a key variable in the optimization process is the substrate feed rate. The unit of substrate feed rate is defined as the volume per unit time (V/t). This variable provides the feeding profile for the bioreactor to provide a certain amount of input at a certain time during the fermentation process. Fig. 1 shows the schematic illustration of a typical simulated fed-batch fermentation model. The substrate feed rate is given as an input to the system. A mathematical model consists of some ordinary differential equations describing the relationship between operating parameters that includes inputs, intermediary and outputs. The biomass and product form the output of the system. The biomass is continuously used by the substrate to produce yield. The most suitable optimization strategy is the use of numerical methods which depend on the use stochastic algorithms.

This is because complexity involved in analytical approaches will increase with the increasing number of state and control variables. Deterministic algorithms also have a high computational overhead as well as have a tendency of premature convergence towards local optima.

Stochastic algorithms or metaheuristics have been previously applied on various bioprocess optimization problems. Evolutionary algorithms (EA) have been utilized on the bioprocess of protein production with *E. coli*, and they have been compared with first order gradient algorithms and with dynamic programming by Roubos, van Straten, and van Boxtel (1999). The optimization of feeding profile for ethanol and penicillin production was applied by Kookos (2004) using Simulated Annealing while the optimization of protein production in *E. coli* was applied using Ant Algorithms by Jayaraman, Kulkarni, Gupta, Rajesh, and Kusumaker (2001). Chiou and Wang (1999) used Differential Evolution (DE) for the optimization of the *Zymomonas mobilis* fed-batch fermentation while Wang and Cheng (1999) used the same algorithm for ethanol production in *Saccharomyces cerevisiae*. Sarkar and Modak (2004) used a genetic algorithm based technique to address fed-batch bioreactor application problems with single or multiple control variables.

A recent study shows DE is a better solution for bio-process applications (Banga, Moles, & Alonso, 2004). Da Ros et al. (2013) have even suggested DE hybrids for these applications after showing DE as the better method in the estimation of the kinetic parameters of an alcoholic fermentation model. Rocha, Mendes, Rocha, Rocha, and Ferreira (2014) compared the performance of EAs, DE and Particle Swarm Optimization (PSO) on four different bioprocess case studies taken from the scientific literature and found that DE had better performance when compared to other algorithms.

In recent years, many new nature-inspired algorithms have emerged such as Particle Swarm Optimization (PSO) (Kennedy & Eberhart, 1995), Artificial Bee Colony Optimization (ABC) (Basturk & Karaboga, 2006), Cuckoo Search (CS) (Yang & Suash, 2009), Firefly Algorithm (FA) (Yang, 2010) and Artificial Algae Algorithm (AAA) (Uymaz, Tezel, & Yel, 2015). A detailed discussion on the proliferation of search algorithms can be seen in Sörensen (2015) and an overview of some of the most widely used can be seen in Burke and Kendall (2014). These algorithms were applied to various problems and have shown improved performance compared to classical algorithms. One of these algorithms, the Backtracking Search Optimization Algorithm (BSA) was recently proposed by Civicioglu (2013). It was developed for solving real-valued numerical optimization problems based on the behaviour of living creatures in social groups revisiting at random intervals to preying areas enriched by food source. BSA was developed based on DE and has many elements similar to DE. However, it improved upon DE by incorporating new elements such as improved mutation and crossover operators and the utilization of a dual population. BSA also has only one control parameter compared to DE which requires two parameters for fine-tuning. With these improvements, it is expected that BSA will perform better than DE. BSA has shown promising results in solving boundary-constrained benchmark problems. Due to its encouraging performance, several studies have been done to investigate BSA's capabilities in solving various engineering problems (Askarzadeh & Coelho, 2014; Das, Mandal, Kar, & Ghoshal, 2014; El-Fergany, 2015; Guney, Durmus, & Basbug, 2014; Song, Zhang, Zhao, & Li, 2015).

BSA uses a unique mechanism for generating trial individual by controlling the amplitude of the search direction through mutation parameter, F . This enables a balanced global and local search, thus enhances its problem solving ability. BSA also consults its historical population which is stored in its memory to generate more efficient trial population, resulting in improved searching ability. Other algorithms such as PSO, DE and DE Covariance Matrix Adap-

Table 1
Pros and cons of related methods.

No.	Method	Paper	Pros	Cons
1.	Differential Evolution (DE)	Storn R, Price K (1997) Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. <i>J Glob Optim</i> 11(4):341–359	A very effective global search algorithm with a quite simple mathematical structure. Able to choose from up to ten different options for its combination of mutation and crossover schemes.	Have three control parameters and the algorithm is sensitive to the initial value of these parameters. The process of determining the optimum mutation and crossover strategies for the problem structure in the DE algorithm is time-consuming.
2.	Covariance Matrix Adaptation Evolution Strategy (CMAES)	Hansen, N. and A. Ostermeier: 1996, 'Adapting Arbitrary Normal Mutation Distributions in Evolution Strategies: The Covariance Matrix Adaptation'. In: <i>Proceedings of the 1996 IEEE Conference on Evolutionary Computation (ICEC '96)</i> . pp. 312–317	A highly competitive, quasi parameter free global optimization algorithm for non-separable objective functions	Poor performance for separable objective functions. Its very algorithmic features are undermined by the presence of constraints
3.	Artificial Bee Colony (ABC)	Karaboga D, Basturk B (2007) A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm. <i>J Glob Optim</i> 39(3):459–471	Sufficiently strong local search ability for various types of problems.	Sensitive to the control parameter used. Poor definition of search direction as it treats the signs of the fitness values equally.
4.	Artificial Algae Algorithm (AAA)	Uymaz, S. A., Tezel, G., & Yel, E. (2015). Artificial algae algorithm (AAA) for nonlinear global optimization. <i>Applied Soft Computing</i> , 31, 153–171.	Robust and high-performance global optimization algorithm.	Have three control parameters. The algorithm is sensitive to the initial value of control parameters.
5.	Genetic Algorithm (GA) Goldberg (1989)	Goldberg, D. E. (1989). <i>Genetic Algorithms in Search, Optimization, and Machine Learning</i> . New York: Addison-Wesley Publishing Company.	Parallelism and ability to solve complex problems.	High sensitivity to its various parameters.

tation Evolution Strategy (CMAES) do not use previous generation populations. BSA employs advanced crossover strategy, which has a non-uniform and complex structure that guarantees the generation of new trial population in each generation. This strategy, which enhances BSA's problem-solving capabilities, is different to those used in genetic algorithm and its variants. Also, its mutation strategy uses only one direction individual for each target individual as opposed to the strategy used in DE and its derivatives, where more than one individual can mutate in each generation. BSA also have only one control parameter in comparison to three used by DE for fine-tuning. Even though BSA is robust and less likely to be trapped in local optima, it has a weakness of poor convergence performance and accuracy. The summary table regarding other metaheuristics used in this work is presented in Table 1. We chose these algorithms in our work for various reasons. CMAES is used because it is recent swarm intelligence metaheuristic with good global convergence. ABC is chosen because it is a widely-used technique among swarm intelligence with promising performance on various problems. AAA is the latest algorithm used in this work and represents the evolution of modern swarm intelligence method. Finally, DE is used as it is an established method in the field of fed-batch fermentation optimization and regarded as the best performing algorithm in the simulation of fed-batch fermentation problems.

Since DE is known to be efficient in solving fermentation problems (Banga et al., 2004; Da Ros et al., 2013; Rocha et al., 2014), BSA as a recent DE-based metaheuristic is proposed in this paper

and we investigate various fermentation problems. Our hypothesis is that it will perform better compared to other stochastic algorithms. BSA, being a powerful EA, is a suitable algorithm to be used in searching for optimal control profiles for the complex bioreactor chemical process. This study applies BSA to different bioprocess case studies and compares its performance with some well-known algorithms from the scientific literature. This study also introduces process optimization in the treatment of winery wastewater. Additionally, we also propose the modelling of fed-batch methane fermentation of sewage sludge. This model is converted from the existing batch model. The bioprocess problems considered in this study cover various aspects of human life, ranging from biofuel production of ethanol and pharmaceutical synthesis of protein and penicillin to treatment of wastewater and sewage sludge. The contributions of this work can be summed as follow:

- Introduces process optimization in the treatment of winery wastewater by applying various metaheuristics to solve the simulation model.
- Proposes the modelling of fed-batch methane fermentation of sewage sludge by converting the existing batch model into a fed-batch model.
- Verify the performance of BSA in solving various bioprocess problems by comparing it with recent metaheuristics including DE.

This paper is divided into five sections. Section 1 is the introduction. Section 2 details the procedures of BSA. Section 3 describes the case studies. Section 4 describes the experiments

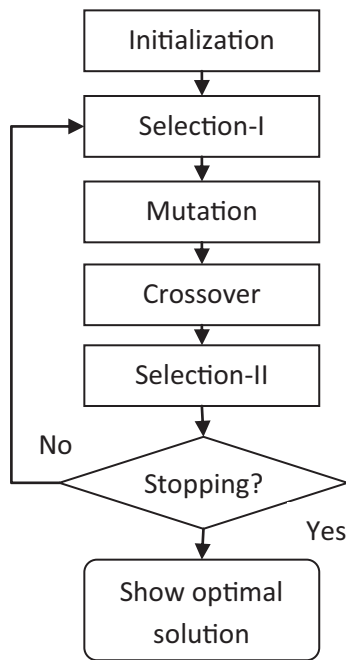


Fig. 2. A general structure of BSA.

conducted and presents the results obtained by each algorithm. Section 5 concludes the paper as well as offers suggestions for future work.

2. Backtracking Search Algorithm (BSA)

BSA is an evolutionary algorithm based on DE (Civicioglu, 2013). It has advanced mutation and crossover operators for the generation of trial populations. It also has balanced exploration and exploitation abilities by generating parameter F . This parameter will control the range of the search direction by adjusting the size of the search amplitude (either large value for global search or low value for local search). The historical population, stored in its memory, promotes effective trial individuals generation and ensures high population diversity. BSA also has the advantage of having only one control parameter, the *mixrate*. This parameter determines the number of elements of individuals that will mutate in a trial, thus facilitating ease of application by reducing the number of parameters that require fine-tuning.

The procedures of BSA can be separated into five processes: initialization, Selection-I, mutation, crossover and Selection-II. A general BSA structure is presented in Fig. 2. For further clarification of the processes, refer to Civicioglu (2013). An overview of the five processes are provided below:

2.1. Initialization

The procedures of BSA begin by initializing the population P as follows:

$$P_{i,j} = lower_j + (upper_j - lower_j) \times random, \quad i = (1, 2, \dots, NP), \\ j = (1, 2, \dots, DP) \quad (1)$$

where NP and DP are the size of the population and the number of dimension of the problem respectively. $random$ is a real value uniformly distributed between 0 and 1. $lower_j$ and $upper_j$ represent the lower and upper bound in the j th element of the i th individual respectively.

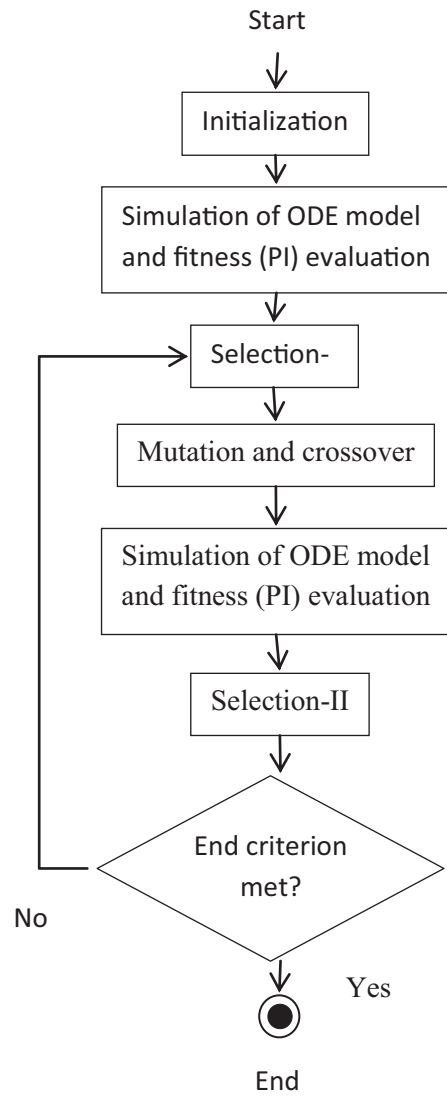


Fig. 3. BSA flowchart.

2.2. Selection-I

In the Selection-I procedure, the historical population $oldP$ is generated to calculate the search direction. Initially, it is calculated as follows:

$$oldP_{i,j} = lower_j + (upper_j - lower_j) \times random, \quad i = (1, 2, \dots, NP), \\ j = (1, 2, \dots, DP) \quad (2)$$

In each iteration, $oldP$ is defined as follows:

$$if \ a < b \ then \ oldP := P|a, \ b \in [0, 1] \quad (3)$$

where: $=$ is the update operation. a and b are two random numbers with uniform distribution between 0 and 1. The above equation ensures that the population in BSA can be randomly selected from historical population. This historical population is memorized by the algorithm until it is changed through a random permutation.

2.3. Mutation

The initial trial population is generated through mutation operation as follows:

$$T = P + (oldP - P) \times F \quad (4)$$

where F is a scale factor which controls the amplitude of the search-direction matrix ($oldP - P$). In this paper, $F = 3 \cdot random$, where $random$ is a random real number with uniform distribution between 0 and 1. By involving the historical population in the calculation of the search-direction matrix, BSA learns from its memory of previous generations to obtain a trial population.

2.4. Crossover

The final trial population T is generated by crossover. The trial individuals with improved fitness values guide the search direction for the optimization problem. The crossover of the BSA works as follows. A binary integer-valued matrix (map) of size $NP \times DP$ is computed in the first step. The individuals of T are generated by using the relevant individuals of P . If $map_{ij} = 1$, T is updated with $T_{ij} = P_{ij}$.

2.5. Selection-II

In the Selection-II phase, the T_i that outperforms the corresponding P_i in terms of fitness value is used to update the P_i . When the best solution P_{best} dominates the previous global optimal value found by the BSA, the global optimal solution is replaced by P_{best} and the global optimal value is also updated to be the fitness value of P_{best} .

3. Case studies

Six fermentation models were used as case studies in this work. These cases are chosen based on the different nature of the bio-processes. The fed batch fermentation case studies considered in this study cover various aspects of human life, ranging from bio-fuel production of ethanol, pharmaceutical synthesis of protein and penicillin, to treatment of wastewater and sewage sludge. The idea is to compare the performance of the BSA in different fed batch fermentation systems.

3.1. Case study I

The first case study in this paper is the fed-batch bioreactor process of ethanol by *Saccharomyces cerevisiae*. This problem was first proposed by [Chen and Hwang \(1990\)](#), with the goal of obtaining the substrate feed rate profile that maximizes the production of ethanol. The model equations ([Chen & Hwang, 1990](#)) are as follows:

$$\frac{dx_1}{dt} = g_1x_1 - u \frac{x_1}{x_4} \tag{5}$$

$$\frac{dx_2}{dt} = -10g_1x_1 + u \frac{150 - x_2}{x_4} \tag{6}$$

$$\frac{dx_3}{dt} = g_1x_1 - u \frac{x_3}{x_4} \tag{7}$$

$$\frac{dx_4}{dt} = u \tag{8}$$

The kinetic variables g_1 and g_2 (h^{-1}) are given by:

$$g_1 = \frac{0.408}{\left(1 + \frac{x_3}{16}\right)} \frac{x_2}{(0.22 + x_2)} \tag{9}$$

$$g_2 = \frac{1}{\left(1 + \frac{x_3}{71.5}\right)} \frac{x_2}{(0.44 + x_2)} \tag{10}$$

The performance index (PI) is defined as:

$$PI = x_3(t_f)x_4(t_f) \tag{11}$$

Table 2
Variables definitions for case study I.

State variables	Definitions
x_1	Cell mass (g/L)
x_2	Substrate concentrations (g/L)
x_3	Ethanol concentrations (g/L)
x_4	Volume of the reactor (L)
u	Feeding rate (L/h)

Table 3
Parameter values for case study I.

Parameter	Value
t_f	54 h
$x_1(0)$	1 g/L
$x_2(0)$	150 g/L
$x_3(0)$	0 g/L
$x_4(0)$	10 L

The variables for case study I are defined in [Table 2](#). The variable constraints are: $0 \leq x_4(t) \leq 200$ and $0 \leq u(t) \leq 12$. The final time, t_f and the initial state conditions are given in [Table 3](#).

3.2. Case study II

The second case study involves induced foreign protein production by recombinant bacteria, firstly proposed by [Lee and Ramirez \(1994\)](#). The problem was later modified by [Tholudur and Ramirez \(1997\)](#). The model equations ([Tholudur & Ramirez, 1997](#)) are as follows:

$$\frac{dx_1}{dt} = u_1 - u_2 \tag{12}$$

$$\frac{dx_2}{dt} = g_1x_2 - \frac{u_1 + u_2}{x_1}x_2 \tag{13}$$

$$\frac{dx_3}{dt} = \frac{100u_1}{x_1} - \frac{u_1 + u_2}{x_1}x_3 - \frac{g_1}{0.51}x_2 \tag{14}$$

$$\frac{dx_4}{dt} = R_{fp}x_2 - \frac{u_1 + u_2}{x_1}x_4 \tag{15}$$

$$\frac{dx_5}{dt} = \frac{4u_2}{x_1} - \frac{u_1 + u_2}{x_1}x_5 \tag{16}$$

$$\frac{dx_6}{dt} = -k_1x_6 \tag{17}$$

$$\frac{dx_7}{dt} = k_2(1 - x_7) \tag{18}$$

The process kinetics is given by:

$$g_1 = \left(\frac{x_3}{14.35 + x_3 \left(1 + \frac{x_3}{111.5}\right)} \right) \left(x_6 + \frac{0.22x_7}{0.22 + x_5} \right) \tag{19}$$

$$R_{fp} = \left(\frac{0.233x_3}{14.35 + x_3 \left(1 + \frac{x_3}{111.5}\right)} \right) \left(\frac{0.005 + x_5}{0.022 + x_5} \right) \tag{20}$$

$$k_1 = k_2 = \frac{0.09x_5}{0.034 + x_5} \tag{21}$$

The PI is defined as:

$$PI = x_4(t_f)x_1(t_f) - Q \int_0^{t_f} u_2(t)dt \tag{22}$$

The variables for case study II are defined in [Table 4](#). The variable constraints are: $0 \leq u_{1,2}(t) \leq 1$. The ratio of the cost of the inducer to the value of the protein product, Q , the final time, t_f and the initial state conditions are given in [Table 5](#).

Table 4
Variables definitions for case study II.

State variables	Definitions
x_1	Reactor volume (L)
x_2	Cell concentrations (g/L)
x_3	Substrate concentrations (g/L)
x_4	Foreign protein concentrations (g/L)
x_5	Inducer concentrations (g/L)
x_6	Inducer shock factors on the cell growth rate
x_7	Recovery factors on the cell growth rate
u_1	Glucose feed rates (L/h)
u_2	Inducer feed rates (L/h)

Table 5
Parameter values for case study II.

Parameter	Value
Q	5
t_f	15 h
$x_1(0)$	1 L
$x_2(0)$	0.1 g/L
$x_3(0)$	40 g/L
$x_4(0)$	0 g/L
$x_5(0)$	0 g/L
$x_6(0)$	1 g/L
$x_7(0)$	0 g/L

Table 6
Variables definitions for case study III.

State variables	Definitions
x_1	Biomass concentrations (g/L)
x_2	penicillin concentrations (g/L)
x_3	substrate concentrations (g/L)
x_4	Volume of the reactor (L)
u	Feeding rate (L/h)

3.3. Case study III

The third case study is the fed-batch fermentation of penicillin which was presented by [Banga, Balsa-Canto, Moles, and Alonso \(2005\)](#). The model equations are as follow:

$$\frac{dx_1}{dt} = g_1x_1 - u\left(\frac{x_1}{500x_4}\right) \tag{23}$$

$$\frac{dx_2}{dt} = g_1x_1 - 0.01x_2 - u\left(\frac{x_2}{500x_4}\right) \tag{24}$$

$$\frac{dx_3}{dt} = -\left(\frac{g_1x_1}{0.47}\right) - \left(\frac{g_2x_2}{1.2}\right) - x_1\left(\frac{0.029x_3}{0.0001 + x_3}\right) + \frac{u}{x_4}\left(1 - \frac{x_3}{500}\right) \tag{25}$$

$$\frac{dx_4}{dt} = \frac{u}{500} \tag{26}$$

The process kinetics are given by:

$$g_1 = 0.11\left(\frac{x_3}{0.006x_1 + x_3}\right) \tag{27}$$

$$g_2 = 0.0055\left(\frac{x_3}{0.0001 + x_3(1 + 10x_3)}\right) \tag{28}$$

The variable constraints are: $0 \leq x_1(t) \leq 40$, $0 \leq x_3(t) \leq 25$, $0 \leq x_4(t) \leq 10$ and $0 \leq u(t) \leq 50$. The PI is defined as:

$$PI = x_2(t_f)x_4(t_f) \tag{29}$$

The variables for case study III are defined in [Table 6](#). The final time, t_f and the initial state conditions are given in [Table 7](#).

Table 7
Parameter values for case study III.

Parameter	Value
t_f	132 h
$x_1(0)$	1.5 g/L
$x_2(0)$	0 g/L
$x_3(0)$	0 g/L
$x_4(0)$	7 L

The above case studies are well-established bioprocess models drawn from the scientific literature. We use these models to verify the robustness of recent metaheuristics. Even though wastewater treatment rarely employs fed-batch operation, [Montalvo et al. \(2010\)](#) are one of the few who used fed-batch operation in biological wastewater treatment. Thus, in the following sections, we propose the applications of fed-batch process optimization using the same metaheuristics on the field of biology wastewater treatment for the purpose of detoxification and methane production and investigate its effectiveness.

3.4. Case study IV & V: pilot-scale fed-batch aerated lagoons treating winery wastewaters

One of the recent techniques in wastewater treatment technology involved the use of fed-batch operation of an aerated lagoon ([Dinçer, 2004](#)). It operates by gradually feeding the highly concentrated wastewater into an aerated lagoon. During this process, the effluent is never removed until after the operating volume of the tank is mostly filled. This enabled reduction of inhibitory or toxic effects through the dilution of organic and toxic compounds in the aeration tank. This results in greater chemical oxygen demand (COD) removal rate. Also, liquid volume in the lagoon increases linearly with time, as it is a process without a stationary phase and has non- constant process variables ([Alberto Vieira Costa, Maria Colla, & Fernando Duarte Filho, 2004](#)).

[Montalvo et al. \(2010\)](#) proposed the treatment of winery wastewaters using two stage pilot-scale fed-batch aerated lagoons. The overall performance of this process can be evaluated by measuring the COD removal efficiency which is defined as the quotient between the difference of the initial COD and effluent COD concentrations and the initial COD concentration ([Pelillo, Rincón, Raposo, Martín, & Borja, 2006](#)). The model equations ([Montalvo et al., 2010](#)) are as follow:

$$\frac{dV}{dt} = F \tag{30}$$

$$\frac{dS}{dt} = \left(\frac{F}{V}\right)(S_0 - S) - \left[\frac{\mu_m(S - S_{nb})}{K_S + (S - S_{nb})} - K_d\right]\left(\frac{X}{Y}\right) \tag{31}$$

$$\frac{dX}{dt} = \left[\left[\frac{\mu_m(S - S_{nb})}{K_S + (S - S_{nb})} - K_d\right] - \left(\frac{F}{V}\right)\right]X \tag{32}$$

The variables for case study IV and V are defined in [Table 8](#). The values for the kinetic parameters are given in [Table 9](#).

The volume constraint is given as: $V \leq V_m$ where V_m is the maximum operational lagoon volume. The values for V_m and the final time, t_f along with the initial conditions for the two stages of operation is given in [Table 10](#).

The bounds on the decision variables are $F \in [0; 2]$ for the first stage and $F \in [0; 1]$ for the second stage. The PI is defined as:

$$PI = (S_0 - S)/S_0 \times 100 - (V_m - V) \times 100 \tag{33}$$

In this paper, we consider the first stage and the second stage of this model as case study IV and case study V respectively.

Table 8
Variables definitions for case study IV and V.

State variables	Definitions
V	Lagoon volume (L or m^3)
F	Volumetric flow-rate (L or m^3 /day),
t	Operation time (days)
μ_m	Maximum specific microbial growth rate (1/days)
S_0	Influent substrate concentrations (mg or g COD/L)
S	Effluent substrate concentrations (mg or g COD/L)
S_{nb}	Non-biodegradable substrate concentration (mg or g COD/L)
X	Cellular or biomass concentration (mg)
Y	Cellular yield coefficient (g VSS/g COD)
K_S	Saturation constant (mg or g COD/L)

Table 9
Kinetic parameters for case study IV and V.

Parameter	Value
μ_m	0.28 1/days
Y	0.26 g VSS/g COD
K_S	175 mg COD/L
K_d	0.12 1/days
S_{nb}	790 mg COD/L

Table 10
Parameter values for case study IV and V.

Parameter	First stage	Second stage
V_m	27.20 m^3	10.80 m^3
t_f	30 days	24 days
$V(0)$	3.470 m^3	5.10 m^3
$S_0(0)$	8700 mg/L	1980.33 mg/L
$X(0)$	900 mg VSS/L	21,373 mg VSS/L

3.5. Case study VI: methane production from sewage sludge fermentation

The model for batch methane fermentation of Sewage Sludge (SS) was proposed by Sosnowski et al. (2008), where the carbon balance process was determined and the simple kinetic model of anaerobic digestion was developed. The batch experiment with the above mentioned feedstock was conducted in a large scale laboratory reactor of working volume of 40.0 dm^{-3} .

The batch operation of methane fermentation can be converted into fed-batch by using the continuity equation:

$$m_{in} - m_{out} - m_{consumed} = \frac{dm}{dt} \quad (34)$$

Replace the formula with the rate of change of substrate:

$$S_{in} - S_{out} - S_{consumed} = \frac{dS}{dt} \quad (35)$$

In fed-batch, no substrate is taken out and the substrate is consumed at a constant rate:

$$S_{in} - kS = \frac{dS}{dt} \quad (36)$$

Where the substrate input is defined as follows:

$$S_{in} = \frac{u \cdot (S_0 - S)}{L} \quad (37)$$

where u is the feed flow rate, S_0 is the substrate concentration in the feed, S is the substrate concentration in the fermentor and L is the volume of the fermentor. When converting a batch model into fed-batch, a diluting term is added into each element. The diluting term is added only to the elements which are either in solid or liquid state. Hence, the elements which are in gaseous state remain unchanged (Rio-Chanona, Zhang, & Vassiliadis, 2016).

Table 11
Variables definitions for case study VI.

State variables	Definitions
k	Constant of first-order reaction (d^{-1})
S	Carbon content in TSS ($g C dm^{-3}$)
V	Carbon content in VFA ($g C dm^{-3}$)
K_S	Saturation constant ($g C dm^{-3}$)
X_0	Biomass concentration ($g C dm^{-3}$)
v_V	Maximum specific utilization of VFA rate (d^{-1})
$Y_{V/S}$	Yield factor of VFA from substrate
$Y_{CH_4/V}$	Yield factor of CH_4 from VFA
$Y_{CO_2/S}$	Yield factor of CO_2 from S
$Y_{CO_2/V}$	Yield factor of CO_2 from VFA

Table 12
Parameter values for case study VI.

Parameter	Value
X_0	5 $g C dm^{-3}$
S_0	20 $g C dm^{-3}$
k	0.11 d^{-1}
$Y_{V/S}$	0.72 d^{-1}
K_S	11.24 $g C dm^{-3}$
v_V	2.08 d^{-1}
$Y_{CH_4/V}$	0.71 d^{-1}
$Y_{CO_2/S}$	0.17 d^{-1}
$Y_{CO_2/V}$	0.22 d^{-1}
t_f	23 d
$S(0)$	4.75 $g C dm^{-3}$
$V(0)$	0 $g C dm^{-3}$
$CH_4(0)$	0 $g C dm^{-3}$
$CO_2(0)$	0 $g C dm^{-3}$
$L(0)$	2.4 dm^3

In this study, the methane fermentation of sewage sludge in fed-batch mode was investigated and is considered as case study VI. The fed-batch operation of sewage sludge fermentation, which was converted from the batch model by Sosnowski et al. (2008), was modelled as follow:

$$\frac{dS}{dt} = \frac{u}{L} * (S_0 - S) - k \cdot S \quad (38)$$

$$\frac{dV}{dt} = Y_{V/S} \cdot k \cdot S - v_V \cdot \frac{V}{K_S + V} \cdot X_0 - V * \frac{u}{L} \quad (39)$$

$$\frac{dCH_4}{dt} = Y_{CH_4/V} \cdot v_V \cdot \frac{V}{K_S + V} \cdot X_0 \quad (40)$$

$$\frac{dCO_2}{dt} = Y_{CO_2/S} \cdot k \cdot S + Y_{CO_2/V} \cdot v_V \cdot \frac{V}{K_S + V} \cdot X_0 \quad (41)$$

$$\frac{dL}{dt} = u \quad (42)$$

The variables for case study VI are defined in Table 11. The constant parameter values, the final time, t_f and the initial state conditions are given in Table 12.

The variable constraints are: $u \in [0; 1]$, $S(t) \leq 5$, $L(t) \leq 40$. The total mass of carbon in the fermentor is constrained as follow:

$$[S(t) + V(t) + CH_4(t) + CO_2(t)] \cdot L(t) \leq 12 \quad (43)$$

The performance index (PI) is given by:

$$PI = CH_4(t_f) \quad (44)$$

4. Experiments and results

In this experiment, BSA is compared with four different meta-heuristics: Covariance Matrix Adaptation Evolution Strategy (CMA-ES) (Hansen & Ostermeier, 1996), Differential Evolution (DE)

Table 13
Symbolic encoding for comparing *t*-tests results.

<i>p</i> -Value	Condition	Symbol
$p \leq 0.001$	mean(A1) > mean(A2)	+++
$p \leq 0.001$	mean(A1) < mean(A2)	---
$0.001 < p \leq 0.01$	mean(A1) > mean(A2)	++
$0.001 < p \leq 0.01$	mean(A1) < mean(A2)	--
$0.01 < p \leq 0.05$	mean(A1) > mean(A2)	+
$0.01 < p \leq 0.05$	mean(A1) < mean(A2)	-
$p \geq 0.05$		0

(Storn & Price, 1997), Artificial Bee Colony (ABC) (Basturk & Karaboga, 2006) and Artificial Algae Algorithm (AAA) (Uymaz et al., 2015). All the algorithms are population-based algorithm. In the context of fed-batch fermentation processes optimization, the solutions found by the algorithms represent the trajectory of input variables. The solutions or input variables are represented by $M \times (N+1)$ real valued vectors. M is the predetermined number of input variables. N is the predetermined size of input variables or the number of feeding intervals. Each vector encodes an input variable as a temporal sequence of values, defined as a piecewise linear function, with N equally spaced, linearly interpolated segments. For the cases where there are more than one input variables, all the M vectors are joined sequentially to create a solution. In this paper, all the case studies have only one input variable except for case study II which has two input variables.

Each solution is evaluated by running a numerical simulation of the differential equation model defined in each case. This simulation is achieved using the Runge–Kutta method provided by Matlab ODE suite. After the simulation, the fitness value of the solution is calculated according to the PI of each case. Also, the relative and absolute error tolerances for integrations of the system dynamics were set to 10^{-8} in order to provide accurate and consistent results. The constraints for each case are handled by implementing constant penalty method. Fig. 3 shows the flowchart of BSA implementation in the experiments.

4.1. Experimental analysis

The means of 30 runs along with its 95% confidence intervals are presented as results in this paper. *t*-test (Goulden, 1956) for two-sample comparisons is implemented in this work. We also employed the Holm correction for the *p*-values (Holm, 1979) for the multiple pairwise comparisons. For ease of presentation, we used a symbolic encoding for the *p*-values obtained from *t*-tests results. Different symbols are employed that gives straightforward comparison between the algorithms and reports whether the mean of algorithm A1 is greater than the mean of A2 or vice versa, as shown in Table 13. In the experiments, some algorithms may show insignificant difference between each other based on their statistical evaluation. However, our goal is to determine the algorithm that can provide consistent good results by having high average and narrow confidence interval for all cases.

4.2. Parameter settings

In our experiments, we use the standard parameters for each algorithm that were suggested by previous studies. The termination condition is set after 200,000 FEs (function evaluations) and the population size for all algorithms is 20. For DE in particular, the parameters are as follow: $F = 0.5$ and $CR = 0.6$. The value of N is equal to the value of t_f in all cases except for case studies II and III (25 and 10 respectively).

4.3. Results and discussion

The results of our experiments for each case study will be shown in a pair of tables. The first table of each pair provide the mean and the 95% confidence intervals for the PI of each algorithm. We probe the PI at four different time-steps: when 25,000, 50,000, 100,000 and 200,000 FEs are performed by each algorithm. This decision is made to estimate the possibilities for terminating the optimization process earlier, immediately after good enough solutions are obtained. The second table of each pair provide the pairwise *t*-test results at 200,000 FEs. These results are intended to signify the statistical differences among the algorithms, where the algorithm on each row of the tables represents A1 on Table 13 while the algorithm on each column represents A2. The results for case studies I–III are provided in Tables 14–19. The results for case studies IV and V are provided in Tables 20–23 while the results for case study VI are provided in Tables 24 and 25.

In case study I, during the early stages of optimization, namely at 25,000 FEs, DE obtains the highest PI as shown in Table 14. Later, CMAES edged other algorithms to obtain better PI at 50,000 and 100,000 FEs. However, at the saturation of optimization, BSA obtained the highest PI after 200,000 FEs. According to the *t*-test in Table 15, BSA performed better than DE, AAA and ABC while performing equally well in comparison to CMAES.

In case study II, during the early stages of optimization namely at 25,000 FEs, DE obtains the highest PI as shown in Table 16. At 50,000 FEs, CMAES improved compared to other algorithms to obtain better PI though DE emerged to perform equally well as CMAES at 100,000 FEs to obtain the highest PI. At the saturation of optimization, BSA, DE and CMAES obtained the highest PI after 200,000 FEs. According to the *t*-test in Table 17, BSA performed better than AAA and ABC while performing equally well in comparison to CMAES and DE.

In case study III, prior to convergence of optimization namely at 25,000, 50,000 and 100,000 FEs, CMAES obtains the highest PI as shown in Table 18. However, at the convergence of optimization, BSA obtained the highest PI after 200,000 FEs. According to the *t*-test in Table 19, BSA performed better than AAA and ABC while performing equally well in comparison to CMAES and DE.

In case study IV, during the early stages of optimization namely at 25,000 FEs, AAA obtains the highest PI as shown in Table 20. At 50,000 FEs, both BSA and AAA obtain the highest PI. However at the later stages of optimization namely at 100,000, and 200,000 FEs, BSA obtained the highest PI. According to the *t*-test in Table 21, all algorithms perform equally well.

In case study V, during the early stages of optimization, namely at 25,000 FEs, AAA obtains the highest PI as shown in Table 22. Later, BSA edged other algorithms to obtain better PI at 50,000 FEs. At 100,000 FEs, AAA obtains the highest PI. At the saturation of optimization, both BSA and AAA obtained the highest PI after 200,000 FEs. According to the *t*-test in Table 23, BSA performed better than CMAES while performing equally well in comparison to AAA, ABC and DE.

In case study VI, during the early stages of optimization namely at 25,000 and 50,000 FEs, CMAES obtains the highest PI as shown in Table 24. Later, DE edged other algorithms to obtain better PI at 100,000 FEs. However at the saturation of optimization, BSA obtained the highest PI after 200,000 FEs. According to the *t*-test in Table 25, BSA performed better than AAA and ABC while performing equally well in comparison to DE and CMAES.

4.3.1. Validation of batch results and improvement using fed batch for case study VI

To show the improvements of fed-batch operation over batch in the methane production from sewage sludge fermentation, we ran a preliminary test for this model. Fig. 4 shows the comparison

Table 14
Mean and confidence intervals for case study I.

Algorithm	PI 25,000 FEs	PI 50,000 FEs	PI 100,000 FEs	PI 200,000 FEs
BSA	20,285 ± 30.73	20,341 ± 26.56	20,392 ± 14.26	20,418 ± 4.71
AAA	20,348 ± 10.42	20,357 ± 14.87	20,369 ± 9.91	20,382 ± 7.02
ABC	7875 ± 2576	11,258 ± 4605	20,299 ± 61.62	20,317 ± 36.98
DE	20,384 ± 4.82	20,381 ± 24.62	20,388 ± 18.93	20,406 ± 2.27
CMAES	20,211 ± 100.2	20,373 ± 46.09	20,403 ± 29.87	20,412 ± 30.03

Table 15
t-test results for case study I.

	BSA	AAA	ABC	DE	CMAES
BSA		+++	+++	++	0
AAA	---		+	---	0
ABC	---	-		-	-
DE	---	+++	++		0
CMAES	0	0	+	0	

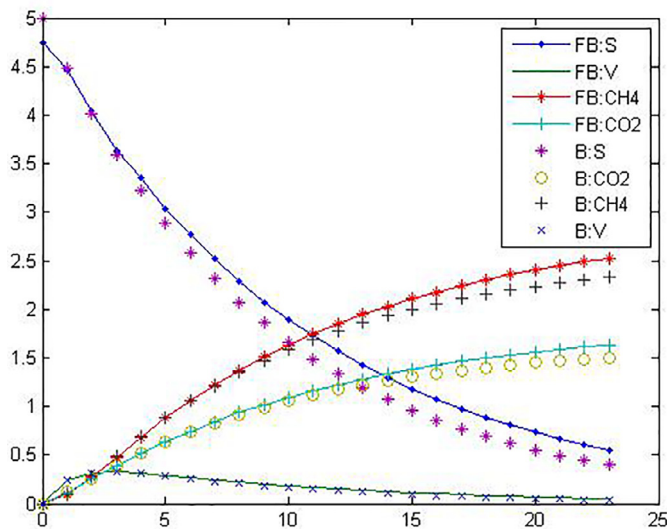


Fig. 4. Comparison of batch and fed-batch for sludge fermentation.

of batch and fed-batch for sludge fermentation where FB stands for fed-batch while B stands for batch. The result for fed-batch was obtained from our preliminary simulation using the methodology described above and BSA as the optimization algorithm. We found that fed-batch produced 8.95% more methane compared to the conventional batch process. This improvement comes from the controlled feeding for each day during the fermentation process. The amount of methane produced by fed-batch starts to increase over batch after the ninth day. It is worth noting that fed-batch was able to produce more methane even when the initial substrate is less than the amount used in batch (4.75 g dm⁻³ for fed-batch compared to 5 g dm⁻³ for batch). Fig. 5 shows the best feeding rate obtained by BSA for case VI.

The results provide several insights on the capabilities of each algorithm in solving fermentation problems. The problems investi-

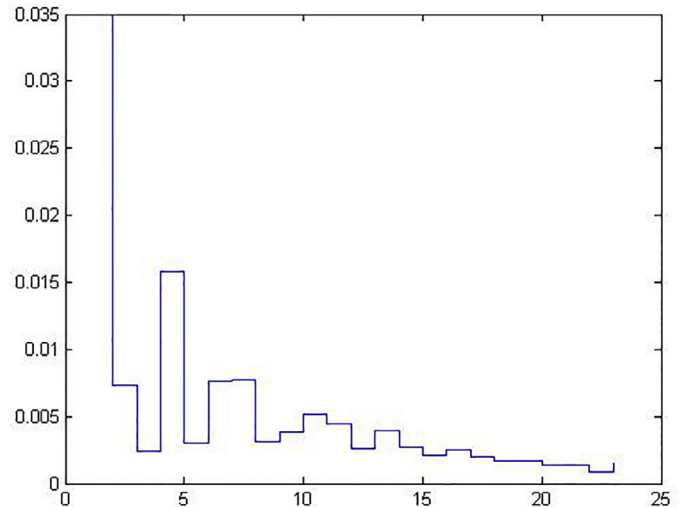


Fig. 5. Control profile for the fed-batch sludge fermentation.

Table 17
t-test results for case study II.

	BSA	AAA	ABC	DE	CMAES
BSA		+++	+++	0	0
AAA	---		+++	---	---
ABC	---	---		---	---
DE	0	+++	+++		0
CMAES	0	+++	+++	0	

gated in this paper can be divided into two categories: constrained and unconstrained. Case study II is unconstrained problem while the rest are constrained problems. For unconstrained problem, all algorithms performed almost equally well and saturated at almost the same PI value. This means that for unconstrained problems, there is flexibility in choosing an algorithm to solve a given problem as most of them converged to the same solution. However, a different scenario exists for constrained problems. For constrained problems, different algorithms performed differently in each problem with the exception of BSA. In overall, BSA is able to obtain the best results in all case studies by providing the highest means and narrow confidence interval. BSA obtained the highest means at 200,000 FEs for all problems except for case II where DE and CMAES saturated at the same highest value as BSA. Case V is an exception for constrained problem where AAA managed to obtain equal means as BSA. Even though DE and CMAES obtained higher

Table 16
Mean and confidence intervals for case study II.

Algorithm	PI 25,000 FEs	PI 50,000 FEs	PI 100,000 FEs	PI 200,000 FEs
BSA	5.5488 ± 0.0038	5.5668 ± 0.0002	5.5676 ± 0.0000	5.5677 ± 0.0000
AAA	5.5642 ± 0.0010	5.5659 ± 0.0004	5.5669 ± 0.0001	5.5673 ± 0.0000
ABC	3.1832 ± 1.1607	5.4637 ± 0.0749	5.5532 ± 0.0072	5.5652 ± 0.0005
DE	5.5671 ± 0.0001	5.5676 ± 0.0000	5.5677 ± 0.0000	5.5677 ± 0.0000
CMAES	0.0000 ± 0.0000	5.5677 ± 0.0000	5.5677 ± 0.0000	5.5677 ± 0.0000

Table 18
Mean and confidence intervals for case study III.

Algorithm	PI 25,000 FEs	PI 50,000 FEs	PI 100,000 FEs	PI 200,000 FEs
BSA	69.352 ± 22.656	87.487 ± 0.2997	87.876 ± 0.0699	87.976 ± 0.0251
AAA	32.433 ± 25.991	85.017 ± 1.0445	85.844 ± 0.6977	86.365 ± 0.7140
ABC	14.733 ± 19.259	78.110 ± 2.4286	78.612 ± 2.1388	78.612 ± 2.1387
DE	43.995 ± 28.743	43.974 ± 28.73	43.99 ± 28.74	43.996 ± 28.744
CMAES	87.770 ± 0.2776	87.968 ± 0.0192	87.968 ± 0.0192	87.968 ± 0.0192

Table 19
t-test results for case study III.

	BSA	AAA	ABC	DE	CMAES
BSA		++	+++	O	O
AAA	--		+++	O	--
ABC	---	---		O	---
DE	O	O	O		O
CMAES	O	++	+++	O	

Table 21
t-test results for case study IV.

	BSA	AAA	ABC	DE	CMAES
BSA		O	O	O	O
AAA	O		O	O	O
ABC	O	O		O	O
DE	O	O	O		O
CMAES	O	O	O	O	

means than BSA at NFE lower than 200,000 for some cases, BSA manages to obtain higher means than both algorithms at the end of 200,000 FEs for all constrained problems. This shows that when given a sufficient amount of NFE, BSA is the best option for solving constrained fermentation problems and provides improved performance compared to DE and other metaheuristics studied in this work for solving bioreactor application problems in general.

AAA shows equal in performance as BSA for case IV and case V while it performs worse in other problems especially for case I and case III. ABC performs the worst in all the case studies except for case IV and case V where it performs relatively well. DE performs well for cases I, II, IV and VI. However, it shows significantly worse results for case III and the V because of the difficulty of satisfying the constraints in these problems. Case III has three constraints to be satisfied, while case V has a single strict constraint as compared to other problems which either have more relaxed constraint or no constraints. CMAES performs well for most cases and even converged faster than BSA in cases I, II, III and VI. However, it struggles to solve case V for the same reason as DE. Previously, Rocha et al. (2014) found that DE obtains the best overall performance for fed-batch fermentation problems. BSA, as an improved DE-based algorithm is expected to perform better than DE. The results obtained from our experiments confirmed that BSA is a superior algorithm.

Zhang and Banks (2013) investigated the impact of different particle size distributions on anaerobic digestion of the organic fraction of municipal solid waste. They mentioned that negligible effect on the enhancement of biogas production was achieved. However the kinetics of the process was faster at semi-continuous experiments. This finding is consistent with our result obtained in case VI (Fig. 4), where only marginal improvement in methane production is observed in fed-batch mode as compared to batch.

Based on the experimental results, all tested algorithms performed almost equally well for the unconstrained problem. All algorithms converged at almost similar value for the unconstrained problem at the end of the run. However, for constrained problems,

which made up the majority of the test problems in this work as well as assumed exist in real-life, we found that BSA is the best performing algorithm. This is due to its high converging accuracy and better stability shown for all the constrained problems. This outcome leads to the implication that BSA improves upon DE and is suitable to be used for solving fed-batch bioreactor process problems.

The performance of BSA compared to other algorithms can be attributed to some of its unique features. For example, BSA employs a more complex and advance crossover strategy compared to DE. This process has two steps. The first step indicates the elements of the individuals to be mutated. The second step is to mutate the indicated elements of trial individuals. There are two strategies that determine which elements of individuals to be manipulated. The first strategy is to use the control parameter *mixrate* to control the number of elements of individuals that will mutate in a trial. The second strategy is by randomly choosing only one individual to be allowed to mutate. This elaborate crossover strategy employed by BSA ensures better generation of its trial population. BSA uses only a single control parameter compared to three parameters used in ABC and AAA. This made BSA easier to be implemented in various types of problems as it requires less effort for fine-tuning the algorithm to suit different types of problems. BSA's unique generation strategy for the mutation parameter *F* enables it to automatically adapt between global search and local search without the need of additional parameters. This is in contrast to AAA which requires the determination of the 'Energy Loss' parameter in order to prefer local search or global search. BSA's boundary control mechanism is also very effective in achieving population diversity and enables it to perform well even in problems with strict constraint requirements. CMA-ES however, performs poorly due to its algorithmic features on problems with strict constraints such as case V.

Table 20
Mean and confidence intervals for case study IV.

Algorithm	PI 25,000 FEs	PI 50,000 FEs	PI 100,000 FEs	PI 200,000 FEs
BSA	89.117 ± 0.1457	89.404 ± 0.0027	89.406 ± 0.0015	89.408 ± 0.0012
AAA	89.402 ± 0.0049	89.404 ± 0.0057	89.405 ± 0.0057	89.407 ± 0.0045
ABC	89.340 ± 0.0530	89.391 ± 0.0102	89.392 ± 0.0101	89.395 ± 0.0069
DE	89.364 ± 0.0272	89.347 ± 0.0290	89.376 ± 0.0141	89.391 ± 0.0134
CMAES	89.140 ± 0.2024	89.359 ± 0.0407	89.371 ± 0.0387	89.373 ± 0.0382

Table 22
Mean and confidence intervals for case study V.

Algorithm	PI 25,000 FEs	PI 50,000 FEs	PI 100,000 FEs	PI 200,000 FEs
BSA	95.049 ± 0.0211	95.071 ± 0.0015	95.072 ± 0.0009	95.073 ± 0.0001
AAA	95.065 ± 0.0083	95.068 ± 0.0051	95.073 ± 0.0001	95.073 ± 0.0000
ABC	95.046 ± 0.0176	95.041 ± 0.0127	95.047 ± 0.0110	95.061 ± 0.0089
DE	75.907 ± 24.797	57.042 ± 30.428	57.043 ± 30.429	57.043 ± 30.429
CMAES	0.0000 ± 0.0000	0.0000 ± 0.0000	0.0000 ± 0.0000	0.0000 ± 0.0000

Table 23
t-test results for case study V.

	BSA	AAA	ABC	DE	CMAES
BSA		O	O	O	+++
AAA	O		O	O	+++
ABC	O	O		O	+++
DE	O	O	O		+
CMAES	---	---	---	-	

Table 24
Mean and confidence intervals for case study VI.

Algorithm	PI 25,000 FEs	PI 50,000 FEs	PI 100,000 FEs	PI 200,000 FEs
BSA	2.5044 ± 0.0028	2.5153 ± 0.0011	2.5186 ± 0.0010	2.522 ± 0.0010
AAA	2.5068 ± 0.0024	2.5112 ± 0.0011	2.5142 ± 0.0009	2.5165 ± 0.0007
ABC	2.4739 ± 0.0072	2.4739 ± 0.0072	2.4739 ± 0.0072	2.4739 ± 0.0072
DE	2.5176 ± 0.0004	2.5192 ± 0.0005	2.5206 ± 0.0004	2.5219 ± 0.0003
CMAES	2.5196 ± 0.0012	2.5196 ± 0.0012	2.5196 ± 0.0012	2.5196 ± 0.0012

Table 25
t-test results for case study VI.

	BSA	AAA	ABC	DE	CMAES
BSA		+++	+++	O	O
AAA	---		+++	---	--
ABC	---	---		---	---
DE	O	+++	+++		+
CMAES	O	++	+++	-	

5. Conclusions

This paper proposes the application of Backtracking Search Algorithm (BSA) on fed-batch fermentation processes. In fed-batch fermentation, nutrient feeding during fermentation process enhances higher product yield. Optimized nutrient feeding stimulates biomass growth and this increases product concentrations while curtailing biomass inhibition due to product and/or nutrient accumulation. Hence, the substrate feed rate plays crucial role in fed-batch process optimization.

This paper also demonstrates the application of metaheuristics on fed-batch aerated lagoon wastewater treatment. This process involves the intermittent feeding of concentrated wastewater into an aerated lagoon. The amount of wastewater to be fed into the lagoon at each day is treated as the variables to be optimized by the metaheuristic. Another contribution of this paper is the formulation of fed-batch model for methane production from sewage sludge fermentation. Apart from the proper and cost-effective disposal of sewage sludge from the Waste Water Treatment Plant (WWTP), anaerobic digestion of sewage sludge plays a key role in the production of biogas namely methane. Usually batch mode fermentation is used to generate biogas. In the current work, biogas production was shown to be further enhanced by using fed-batch operation as feed rate becomes key optimization variable for metaheuristics.

Based on past literature, Differential Evolution (DE) is considered as a more appropriate solution for bio-process applications. Since DE is known to be efficient in solving fermentation prob-

lems, BSA as a recent DE-based metaheuristic is deemed to be superior to the former. Four recent metaheuristics that included DE were applied on three bioprocess engineering problems widely used in literature alongside with the problems mentioned above and the results were compared with BSA. From the results, BSA showed consistency of obtaining highest fitness value in comparison to other four metaheuristics for all the cases at convergence point. Therefore, BSA is suggested as the first choice metaheuristic to use when solving bioprocess engineering problems.

All the case studies presented in this paper consisted of single-objective problems. It is interesting to evaluate the performance of metaheuristics in solving multi-objectives fed-batch fermentation problems. In multi-objectives problems, the objectives to be optimized can extend beyond the production rate and include substrate utilization, environmental impact and economic benefits. This can be considered in future works.

Acknowledgement

This research work is supported by Fundamental Research Grant Scheme (FRGS), FP042-2014B.

Reference

- Alberto Vieira Costa, J., Maria Colla, L., & Fernando Duarte Filho, P. (2004). Improving *Spirulina platensis* biomass yield using a fed-batch process. *Bioresource Technology*, 92(3), 237–241.
- Appels, L., Baeyens, J., Degreè, J., & Dewil, R. (2008). Principles and potential of the anaerobic digestion of waste-activated sludge. *Progress in Energy and Combustion Science*, 34(6), 755–781.
- Askarzadeh, A., & Coelho, L. d. S. (2014). A backtracking search algorithm combined with Burger's chaotic map for parameter estimation of PEMFC electrochemical model. *International Journal of Hydrogen Energy*, 39(21), 11165–11174.
- Banga, J. R., Balsa-Canto, E., Moles, C. G., & Alonso, A. A. (2005). Dynamic optimization of bioprocesses: Efficient and robust numerical strategies. *Journal of Biotechnology*, 117(4), 407–419.
- Banga, J. R., Moles, C. G., & Alonso, A. A. (2004). Global optimization of bioprocesses using stochastic and hybrid methods. In C. A. Floudas, & P. Pardalos (Eds.), *Frontiers in global optimization* (pp. 45–70). Boston, MA: Springer US.
- Basturk, B., & Karaboga, D. (2006). An artificial bee colony (ABC) algorithm for numeric function optimization. In *Proceedings of the IEEE swarm intelligence symposium* (pp. 687–697). IEEE.
- Burke, E. K., & Kendall, G. (2014). *Search methodologies - introductory tutorials in optimization and decision support techniques* (2nd ed.). New York: Springer US.
- Chen, C.-T., & Hwang, C. (1990). Optimal control computation for differential-algebraic process systems with general constraints. *Chemical Engineering Communications*, 97(1), 9–26.
- Chen, L., Nguang, S. K., Chen, X. D., & Li, X. M. (2004). Modelling and optimization of fed-batch fermentation processes using dynamic neural networks and genetic algorithms. *Biochemical Engineering Journal*, 22(1), 51–61.
- Chiou, J.-P., & Wang, F.-S. (1999). Hybrid method of evolutionary algorithms for static and dynamic optimization problems with application to a fed-batch fermentation process. *Computers & Chemical Engineering*, 23(9), 1277–1291.
- Civicioglu, P. (2013). Backtracking search optimization algorithm for numerical optimization problems. *Applied Mathematics and Computation*, 219(15), 8121–8144.
- Da Ros, S., Colusso, G., Weschenfelder, T. A., de Marsillac Terra, L., de Castilhos, F., Corazza, M. L., et al. (2013). A comparison among stochastic optimization algorithms for parameter estimation of biochemical kinetic models. *Applied Soft Computing*, 13(5), 2205–2214.
- Das, S., Mandal, D., Kar, R., & Ghoshal, S. P. (2014). Interference suppression of linear antenna arrays with combined Backtracking Search Algorithm and Differential Evolution. In *International conference on communications and signal processing (ICCSP)* (pp. 162–166).
- del Rio-Chanona, E. A., Zhang, D., & Vassiliadis, V. S. (2016). Model-based real-time optimisation of a fed-batch cyanobacterial hydrogen production process using economic model predictive control strategy. *Chemical Engineering Science*, 142, 289–298.

- Dinçer, A. R. (2004). Use of activated sludge in biological treatment of boron containing wastewater by fed-batch operation. *Process Biochemistry*, 39(6), 723–730.
- El-Fergany, A. (2015). Optimal allocation of multi-type distributed generators using backtracking search optimization algorithm. *International Journal of Electrical Power & Energy Systems*, 64, 1197–1205.
- Goldberg, D. E. (1989). *Genetic algorithms in search, optimization, and machine learning*. New York: Addison-Wesley Publishing Company.
- Goulden, C. H. (1956). *Methods of statistical analysis* (2nd ed.). John Wiley & Sons Ltd.
- Guney, K., Durmus, A., & Basbug, S. (2014). Backtracking search optimization algorithm for synthesis of concentric circular antenna arrays. *International Journal of Antennas and Propagation*, 2014. Article ID 250841, 11 pages.
- Hansen, N., & Ostermeier, A. (1996). Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation. In *Proceedings of IEEE international conference on evolutionary computation* (pp. 312–317). IEEE.
- Holm, S. (1979). A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6(2), 65–70.
- Baeyens, J., Hosten, L., & Van Vaerenbergh, E. (1997). *Afvalwaterzuivering (wastewater treatment)* (2nd ed). The Netherlands: Kluwer Academic Publishers.
- Jayaraman, V. K., Kulkarni, B. D., Gupta, K., Rajesh, J., & Kusumaker, H. S. (2001). Dynamic optimization of fed-batch bioreactors using the ant algorithm. *Biotechnology Progress*, 17(1), 81–88.
- Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In *IEEE international conference on neural networks* (pp. 1942–1948).
- Kookos, I. K. (2004). Optimization of batch and fed-batch bioreactors using simulated annealing. *Biotechnology Progress*, 20(4), 1285–1288.
- Lee, J., & Ramirez, W. F. (1994). Optimal fed-batch control of induced foreign protein production by recombinant bacteria. *AIChE Journal*, 40(5), 899–907.
- Levišauskas, D., & Tekorius, T. (2005). Model-based optimization of fed-batch fermentation processes using predetermined type feed-rate time profiles. A comparative study. *Information Technology and Control*, 34(3), 231–236.
- Liu, C., Gong, Z., Shen, B., & Feng, E. (2013). Modelling and optimal control for a fed-batch fermentation process. *Applied Mathematical Modelling*, 37(3), 695–706.
- Montalvo, S., Guerrero, L., Rivera, E., Borja, R., Chica, A., & Martín, A. (2010). Kinetic evaluation and performance of pilot-scale fed-batch aerated lagoons treating winery wastewaters. *Bioresource Technology*, 101(10), 3452–3456.
- Pelillo, M., Rincón, B., Raposo, F., Martín, A., & Borja, R. (2006). Mathematical modelling of the aerobic degradation of two-phase olive mill effluents in a batch reactor. *Biochemical Engineering Journal*, 30(3), 308–315.
- Peng, W., Zhong, J., Yang, J., Ren, Y., Xu, T., Xiao, S., ... Tan, H. (2014). The artificial neural network approach based on uniform design to optimize the fed-batch fermentation condition: Application to the production of iturin A. *Microbial Cell Factories*, 13(1), 54.
- Rocha, M., Mendes, R., Rocha, O., Rocha, I., & Ferreira, E. C. (2014). Optimization of fed-batch fermentation processes with bio-inspired algorithms. *Expert Systems with Applications*, 41(5), 2186–2195.
- Roubos, J. A., van Straten, G., & van Boxtel, A. J. B. (1999). An evolutionary strategy for fed-batch bioreactor optimization; concepts and performance. *Journal of Biotechnology*, 67(2–3), 173–187.
- Sarkar, D., & Modak, J. M. (2004). Optimization of fed-batch bioreactors using genetic algorithm: Multiple control variables. *Computers & Chemical Engineering*, 28(5), 789–798.
- Song, X., Zhang, X., Zhao, S., & Li, L. (2015). Backtracking search algorithm for effective and efficient surface wave analysis. *Journal of Applied Geophysics*, 114, 19–31.
- Sörensen, K. (2015). Metaheuristics—the metaphor exposed. *International Transactions in Operational Research*, 22(1), 3–18.
- Sosnowski, P., Klepacz-Smolka, A., Kaczorek, K., & Ledakowicz, S. (2008). Kinetic investigations of methane co-fermentation of sewage sludge and organic fraction of municipal solid wastes. *Bioresource Technology*, 99(13), 5731–5737.
- Storn, R., & Price, K. (1997). Differential evolution – A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11(4), 341–359.
- Tholudur, A., & Ramirez, W. F. (1997). Obtaining smoother singular arc policies using a modified iterative dynamic programming algorithm. *International Journal of Control*, 68(5), 1115–1128.
- Uymaz, S. A., Tezel, G., & Yel, E. (2015). Artificial algae algorithm (AAA) for nonlinear global optimization. *Applied Soft Computing*, 31, 153–171.
- Wang, F.-S., & Cheng, W.-M. (1999). Simultaneous optimization of feeding rate and operation parameters for fed-batch fermentation processes. *Biotechnology Progress*, 15(5), 949–952.
- Yang, X. S. (2010). *Nature-inspired metaheuristic algorithms*. Beckington, UK: Luniver Press.
- Yang, X. S., & Suash, D. (2009). Cuckoo search via levy flights. In *World congress on nature & biologically inspired computing (NaBIC 2009)* (pp. 210–214).
- Zhang, Y., & Banks, C. J. (2013). Impact of different particle size distributions on anaerobic digestion of the organic fraction of municipal solid waste. *Waste Management*, 33(2), 297–307.