

1 Title: Optimization of fed-batch fermentation processes using the Backtracking Search Algorithm.

2 Authors: Mohamad Zihin bin Mohd Zain

3 Last/Family name: Mohd Zain

4 First name: Mohamad Zihin

5 University of Malaya

6 Department of Electrical Engineering

7 Faculty of Engineering, University of Malaya

8 50603, Kuala Lumpur, Malaysia

9 mzihin_91@siswa.um.edu.my

10 * Jeevan Kanesan

11 Last/Family name: Kanesan

12 First name: Jeevan

13 University of Malaya

14 Department of Electrical Engineering

15 Faculty of Engineering, University of Malaya

16 50603, Kuala Lumpur, Malaysia

17 jievan@um.edu.my

18 00603-79675388

19 Graham Kendall

20 Last/Family name: Kendall

21 First name: Graham

22 University of Nottingham

23 School Of Computer Science,

24 University of Nottingham, Jubilee Campus,

25 Nottingham NG8 1BB, UK

26 graham.kendall@nottingham.ac.uk

27

28 Joon Huang Chuah

29 Last/Family name: Chuah

30 First name: Joon Huang

31 University of Malaya

32 Department of Electrical Engineering

33 Faculty of Engineering, University of Malaya

34 50603, Kuala Lumpur, Malaysia

35 jhchuah@um.edu.my

36 * corresponding author

37

38

39

40

41

42

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

58

59 Highlights:

- 60 • Optimizations in winery wastewater and sewage sludge treatment are tackled.
- 61 • Recent metaheuristics namely CMAES, BSA and DE are found to give competent results.
- 62 • Improved DE metaheuristic, BSA gives best overall performance for all problems.

63

64 Keywords:

65 Fed-batch fermentation; Backtracking Search Algorithm; Evolutionary algorithms; Wastewater
66 treatment; Feeding trajectory optimization; Sewage sludge

67

68

69

70

71

72

73

74

75 1. Introduction

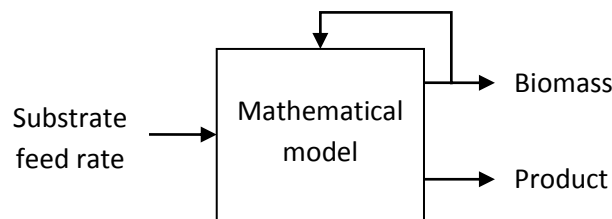
76 The diverse applications of optimization which range from manufacturing and engineering to
77 business and medication have attracted many researchers to explore the field. Since the mid-20th
78 century, researchers have developed a number of high performance optimization methods by taking
79 inspiration from biology, physics, social and cultural behaviour, neurology and other disciplines. For
80 instance, particle swarm optimization (PSO) (Kennedy & Eberhart, 1995) is a bio-inspired
81 metaheuristics which is based on the metaphors of social interaction and communication (e.g., fish
82 schooling and bird flocking). These algorithms are classified as a branch of optimization techniques
83 called swarm intelligence metaheuristics. These metaheuristics use a process of trial and error to
84 discover the solution of a problem and consists of certain trade-off of randomization and local
85 search. They have a unique feature where more than one solution is evaluated simultaneously in a
86 single iteration. Their most appealing characteristics are their derivation-free mechanisms, relatively
87 simple structures and stochastic nature. This enables faster convergence and less expensive
88 computation as compared to deterministic method.

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

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

116 The optimization of fed-batch fermentation process was intensively studied in recent years.
117 Chen et al. (2004) proposed the optimization of a fed-batch bioreactor using a cascade recurrent
118 neural network (RNN) model and modified genetic algorithm (GA). They applied their method in the
119 fed-batch fermentation of a common yeast species in food technology, *Saccharomyces cerevisiae*.
120 Levišauskas and Tekorius (2015) investigated various fed-batch fermentation processes optimization
121 using the feed-rate time profile approximating functions and the parametric optimization procedure.
122 In their work, four types of time functions namely constant feed-rate, ramp-type function,
123 exponential function and a network of radial basis functions are compared. The parametric
124 optimization problems were solved using chemotaxis random search algorithm. Liu et al. (2013)
125 proposed a new nonlinear dynamical system to formulate the fed-batch fermentation process of
126 glycerol bioconversion to 1,3-propanediol (1,3-PD). Peng et al. (2014) studied the fed-batch
127 fermentation process of an antibiotic, iturin A using an artificial neural network-genetic algorithm
128 (ANN-GA) and uniform design (UD).

129



130

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

132

133 In fed-batch fermentation simulation, a key variable in the optimization process is the
134 substrate feed rate. The unit of substrate feed rate is defined as the volume per unit time (V/t). This
135 variable provides the feeding profile for the bioreactor to provide a certain amount of input at a
136 certain time during the fermentation process. Figure 1 shows the schematic illustration of a typical
137 simulated fed-batch fermentation model. The substrate feed rate is given as an input to the system.
138 A mathematical model consists of some ordinary differential equations describing the relationship
139 between operating parameters that includes inputs, intermediary and outputs. The biomass and
140 product form the output of the system. The biomass is continuously used by the substrate to
141 produce yield. The most suitable optimization strategy is the use of numerical methods which
142 depend on the use stochastic algorithms. This is because complexity involved in analytical
143 approaches will increase with the increasing number of state and control variables. Deterministic
144 algorithms also have a high computational overhead as well as have a tendency of premature
145 convergence towards local optima.

146 Stochastic algorithms or metaheuristics have been previously applied on various bioprocess
147 optimization problems. Evolutionary algorithms (EA) have been utilized on the bioprocess of protein
148 production with *E. coli*, and they have been compared with first order gradient algorithms and with
149 dynamic programming by Roubos, van Straten, and van Boxtel (1999). The optimization of feeding
150 profile for ethanol and penicillin production was applied by Kookos (2004) using Simulated Annealing

151 while the optimization of protein production in *E. coli* was applied using Ant Algorithms by
152 Jayaraman et al. (2001). Chiou and Wang (1999) used Differential Evolution (DE) for the optimization
153 of the Zymomous mobilis fed-batch fermentation while Wang and Cheng (1999) used the same
154 algorithm for ethanol production in *Saccharomyces cerevisiae*. Sarkar and Modak (2004) used a
155 genetic algorithm based technique to address fed-batch bioreactor application problems with single
156 or multiple control variables.

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

163 In recent years, many new nature-inspired algorithms have emerged such as Particle Swarm
164 Optimization (PSO) (Kennedy & Eberhart, 1995), Artificial Bee Colony Optimization (ABC) (Basturk,
165 2006), Cuckoo Search (CS) (Yang & Suash, 2009), Firefly Algorithm (FA) (Yang, 2010) and Artificial
166 Algae Algorithm (AAA) (Uymaz, Tezel, & Yel, 2015). A detailed discussion on the proliferation of
167 search algorithms can be seen in Sörensen (2015) and an overview of some of the most widely used
168 can be seen in Burke & Kendall (2014). These algorithms were applied to various problems and have
169 shown improved performance compared to classical algorithms. One of these algorithms, the
170 Backtracking Search Optimization Algorithm (BSA) was recently proposed by Civicioglu (2013). It was
171 developed for solving real-valued numerical optimization problems based on the behaviour of living
172 creatures in social groups revisiting at random intervals to preying areas enriched by food source.
173 BSA was developed based on DE and has many elements similar to DE. However, it improved upon
174 DE by incorporating new elements such as improved mutation and crossover operators and the
175 utilization of a dual population. BSA also has only one control parameter compared to DE which
176 requires two parameters for fine-tuning. With these improvements, it is expected that BSA will
177 perform better than DE. BSA has shown promising results in solving boundary-constrained
178 benchmark problems. Due to its encouraging performance, several studies have been done to
179 investigate BSA's capabilities in solving various engineering problems (Song et al., 2015; Guney,
180 Durmus, & Basbug, 2014; El-Fergany, 2015; Askarzadeh & Coelho, 2014; & Das et al., 2014).

181 BSA uses a unique mechanism for generating trial individual by controlling the amplitude of
182 the search direction through mutation parameter, F . This enables a balanced global and local search,
183 thus enhances its problem solving ability. BSA also consults its historical population which is stored
184 in its memory to generate more efficient trial population, resulting in improved searching ability.
185 Other algorithms such as PSO, DE and DE Covariance Matrix Adaptation Evolution Strategy (CMAES)
186 do not use previous generation populations. BSA employs advanced crossover strategy, which has a
187 non-uniform and complex structure that guarantees the generation of new trial population in each
188 generation. This strategy, which enhances BSA's problem-solving capabilities, is different to those
189 used in genetic algorithm and its variants. Also, its mutation strategy uses only one direction
190 individual for each target individual as opposed to the strategy used in DE and its derivatives, where
191 more than one individual can mutate in each generation. BSA also have only one control parameter
192 in comparison to three used by DE for fine-tuning. Even though BSA is robust and less likely to be
193 trapped in local optima, it has a weakness of poor convergence performance and accuracy. The

194 summary table regarding other metaheuristics used in this work is presented in table 1. We chose
195 these algorithms in our work for various reasons. CMAES is used because it is recent swarm
196 intelligence metaheuristic with good global convergence. ABC is chosen because it is a widely-used
197 technique among swarm intelligence with promising performance on various problems. AAA is the
198 latest algorithm used in this work and represents the evolution of modern swarm intelligence
199 method. Finally, DE is used as it is an established method in the field of fed-batch fermentation
200 optimization and regarded as the best performing algorithm in the simulation of fed-batch
201 fermentation problems.

202 Since DE is known to be efficient in solving fermentation problems (Banga, Moles & Alonso,
203 2004; Da Ros et al., 2013 & Rocha et al., 2014), BSA as a recent DE-based metaheuristic is proposed
204 in this paper and we investigate various fermentation problems. Our hypothesis is that it will
205 perform better compared to other stochastic algorithms. BSA, being a powerful EA, is a suitable
206 algorithm to be used in searching for optimal control profiles for the complex bioreactor chemical
207 process. This study applies BSA to different bioprocess case studies and compares its performance
208 with some well-known algorithms from the scientific literature. This study also introduces process
209 optimization in the treatment of winery wastewater. Additionally, we also propose the modelling of
210 fed-batch methane fermentation of sewage sludge. This model is converted from the existing batch
211 model. The bioprocess problems considered in this study cover various aspects of human life,
212 ranging from biofuel production of ethanol and pharmaceutical synthesis of protein and penicillin to
213 treatment of wastewater and sewage sludge. The contributions of this work can be summed as
214 follow:

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

221 This paper is divided into 5 sections. Section 1 is the introduction. Section 2 details the
222 procedures of BSA. Section 3 describes the case studies. Section 4 describes the experiments
223 conducted and presents the results obtained by each algorithm. Section 5 concludes the paper as
224 well as offers suggestions for future work.

225

226

227

228

229

230

231

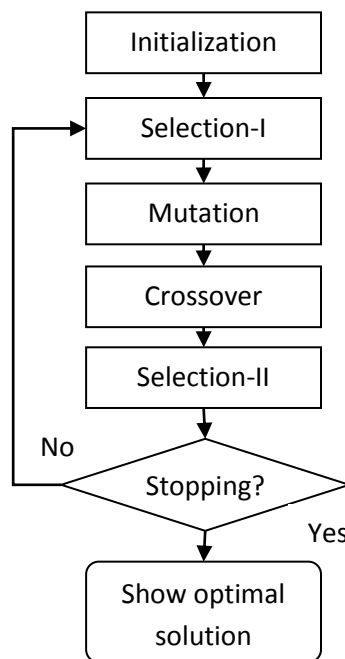
No.	Method	Paper	Pros	Cons
1.	Differential Evolution (DE)	Storn R, Price K (1997) Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces. <i>J Glob Optim</i> 11(4):341–359	A very effective global search algorithm with a quite simple mathematical structure. 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, 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.

235 **2. Backtracking Search Algorithm (BSA)**

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

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

249



250

251

Fig. 2. A general structure of BSA

252

253 **2.1. Initialization**

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

255
$$P_{i,j} = lower_j + (upper_j - lower_j) \times random, i = (1,2, \dots, NP), j = (1,2, \dots, DP) \quad (1)$$

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

259 2.2. Selection-I

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

$$262 \quad oldP_{i,j} = lower_j + (upper_j - lower_j) \times random, i = (1, 2, \dots, NP), j = (1, 2, \dots, DP) \quad (2)$$

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

$$264 \quad \text{if } a < b \text{ then } oldP := P | a, b \in [0, 1] \quad (3)$$

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

269

270 2.3. Mutation

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

$$272 \quad T = P + (oldP - P) \times F \quad (4)$$

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

277

278 2.4. Crossover

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

284

285 2.5. Selection-II

286 In the Selection-II phase, the T_i that outperforms the corresponding P_i in terms of fitness
287 value is used to update the P_i . When the best solution P_{best} dominates the previous global optimal

288 value found by the BSA, the global optimal solution is replaced by P_{best} and the global optimal
 289 value is also updated to be the fitness value of P_{best} .

290 3. Case studies

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

297

298 3.1. Case study I

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

$$303 \frac{dx_1}{dt} = g_1 x_1 - u \frac{x_1}{x_4} \quad (5)$$

$$304 \frac{dx_2}{dt} = -10g_1 x_1 + u \frac{150-x_2}{x_4} \quad (6)$$

$$305 \frac{dx_3}{dt} = g_1 x_1 - u \frac{x_3}{x_4} \quad (7)$$

$$306 \frac{dx_4}{dt} = u \quad (8)$$

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

$$308 g_1 = \frac{0.408}{(1+\frac{x_3}{16})} \frac{x_2}{(0.22+x_2)} \quad (9)$$

$$309 g_2 = \frac{1}{(1+\frac{x_3}{71.5})} \frac{x_2}{(0.44+x_2)} \quad (10)$$

310 The performance index (PI) is defined as:

$$311 PI = x_3(t_f)x_4(t_f) \quad (11)$$

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

314

315

316

317

318 **Table 2**
 319 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)

320

321 **Table 3**
 322 Parameter values for case study I.

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

323

324 3.2. Case study II

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

$$328 \quad \frac{dx_1}{dt} = u_1 - u_2 \quad (12)$$

$$329 \quad \frac{dx_2}{dt} = g_1 x_2 - \frac{u_1 + u_2}{x_1} x_2 \quad (13)$$

$$330 \quad \frac{dx_3}{dt} = \frac{100u_1}{x_1} - \frac{u_1 + u_2}{x_1} x_3 - \frac{g_1}{0.51} x_2 \quad (14)$$

$$331 \quad \frac{dx_4}{dt} = R_{fp} x_2 - \frac{u_1 + u_2}{x_1} x_4 \quad (15)$$

$$332 \quad \frac{dx_5}{dt} = \frac{4u_2}{x_1} - \frac{u_1 + u_2}{x_1} x_5 \quad (16)$$

$$333 \quad \frac{dx_6}{dt} = -k_1 x_6 \quad (17)$$

$$334 \quad \frac{dx_7}{dt} = k_2 (1 - x_7) \quad (18)$$

335 The process kinetics is given by:

$$336 \quad g_1 = \left(\frac{x_3}{14.35 + x_3 \left(1 + \frac{x_3}{111.5}\right)} \right) \left(x_6 + \frac{0.22x_7}{0.22 + x_5} \right) \quad (19)$$

337
$$R_{fp} = \left(\frac{0.233x_3}{14.35+x_3(1+\frac{x_3}{111.5})} \right) \left(\frac{0.005+x_5}{0.022+x_5} \right) \quad (20)$$

338
$$k_1 = k_2 = \frac{0.09x_5}{0.034+x_5} \quad (21)$$

339 The PI is defined as:

340
$$PI = x_4(t_f)x_1(t_f) - Q \int_0^{t_f} u_2(t)dt \quad (22)$$

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

344

345 **Table 4**

346 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)

347

348 **Table 5**

349 Parameter values for case study II.

Parameter	Value
Q	5
t_f	15 hours
$x_1(0)$	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

350

351 *3.3. Case study III*

352 The third case study is the fed-batch fermentation of penicillin which was presented by
 353 Banga et al. (2005).The model equations are as follow:

354
$$\frac{dx_1}{dt} = g_1x_1 - u \left(\frac{x_1}{500x_4} \right) \quad (23)$$

355
$$\frac{dx_2}{dt} = g_1x_1 - 0.01x_2 - u \left(\frac{x_2}{500x_4} \right) \quad (24)$$

356
$$\frac{dx_3}{dt} = - \left(\frac{g_1x_1}{0.47} \right) - \left(\frac{g_2x_2}{1.2} \right) - x_1 \left(\frac{0.029x_3}{0.0001+x_3} \right) + \frac{u}{x_4} \left(1 - \frac{x_3}{500} \right) \quad (25)$$

357
$$\frac{dx_4}{dt} = \frac{u}{500} \quad (26)$$

358 The process kinetics are given by:

359
$$g_1 = 0.11 \left(\frac{x_3}{0.006x_1+x_3} \right) \quad (27)$$

360
$$g_2 = 0.0055 \left(\frac{x_3}{0.0001+x_3(1+10x_3)} \right) \quad (28)$$

361 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 $0 \leq$
 362 $u(t) \leq 50$. The PI is defined as:

363
$$PI = x_2(t_f)x_4(t_f) \quad (29)$$

364 The variables for case study III are defined in Table 6. The final time, t_f and the initial state
 365 conditions are given in Table 7.

366

367 **Table 6**

368 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)

369

370 **Table 7**

371 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

372

373 The above case studies are well-established bioprocess models drawn from the scientific
 374 literature. We use these models to verify the robustness of recent metaheuristics. Even though

375 wastewater treatment rarely employs fed-batch operation, Montalvo et al. (2010) are one of the few
 376 who used fed-batch operation in biological wastewater treatment. Thus, in the following sections,
 377 we propose the applications of fed-batch process optimization using the same metaheuristics on the
 378 field of biology wastewater treatment for the purpose of detoxification and methane production and
 379 investigate its effectiveness.

380

381 3.4. Case study IV & V: Pilot-scale fed-batch aerated lagoons treating winery wastewaters

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

390 Montalvo et al. (2010) proposed the treatment of winery wastewaters using two stage pilot-
 391 scale fed-batch aerated lagoons. The overall performance of this process can be evaluated by
 392 measuring the COD removal efficiency which is defined as the quotient between the difference of
 393 the initial COD and effluent COD concentrations and the initial COD concentration (Pelillo et al.,
 394 2006). The model equations (Montalvo et al., 2010) are as follow:

$$395 \quad \frac{dV}{dt} = F \quad (30)$$

$$396 \quad \frac{dS}{dt} = \left(\frac{F}{V}\right) (S_0 - S) - \left[\frac{\mu_m(S-S_{nb})}{K_S+(S-S_{nb})} - K_d\right] \left(\frac{X}{Y}\right) \quad (31)$$

$$397 \quad \frac{dX}{dt} = \left[\frac{\mu_m(S-S_{nb})}{K_S+(S-S_{nb})} - K_d\right] - \left(\frac{F}{V}\right) X \quad (32)$$

398 The variables for case study IV and V are defined in Table 8. The values for the kinetic parameters
 399 are given in Table 9.

400

401

402

403

404

405

406

407 **Table 8**
 408 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)

409

410 **Table 9**
 411 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

412

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

416

417 **Table 10**
 418 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	21373 mg VSS/L

419

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

$$422 \quad PI = (S_0 - S)/S_0 \times 100 - (V_m - V) \times 100 \quad (33)$$

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

425 3.5. Case study VI: Methane production from sewage sludge fermentation

426 The model for batch methane fermentation of Sewage Sludge (SS) was proposed by
 427 Sosnowski et al. (2008), where the carbon balance process was determined and the simple kinetic
 428 model of anaerobic digestion was developed. The batch experiment with the above mentioned
 429 feedstock was conducted in a large scale laboratory reactor of working volume of 40.0 dm³.

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

$$432 \quad m_{in} - m_{out} - m_{consumed} = \frac{dm}{dt} \quad (34)$$

433 Replace the formula with the rate of change of substrate:

$$434 \quad S_{in} - S_{out} - S_{consumed} = \frac{dS}{dt} \quad (35)$$

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

$$436 \quad S_{in} - kS = \frac{dS}{dt} \quad (36)$$

437 Where the substrate input is defined as follows:

$$438 \quad S_{in} = \frac{u \cdot (S_0 - S)}{L} \quad (37)$$

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

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

$$448 \quad \frac{dS}{dt} = \frac{u}{L} \cdot (S_0 - S) - k \cdot S \quad (38)$$

$$449 \quad \frac{dV}{dt} = Y_{V/S} \cdot k \cdot S - v_V \cdot \frac{V}{K_S + V} \cdot X_0 - V \cdot \frac{u}{L} \quad (39)$$

$$450 \quad \frac{dCH_4}{dt} = Y_{CH_4/V} \cdot v_V \cdot \frac{V}{K_S + V} \cdot X_0 \quad (40)$$

$$451 \quad \frac{dCO_2}{dt} = Y_{CO_2/S} \cdot k \cdot S + Y_{CO_2/V} \cdot v_V \cdot \frac{V}{K_S + V} \cdot X_0 \quad (41)$$

$$452 \quad \frac{dL}{dt} = u \quad (42)$$

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

455 **Table 11**
 456 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

457
 458 The variable constraints are: $u \in [0; 1], S(t) \leq 5, L(t) \leq 40$. The total mass of carbon in the
 459 fermentor is constrained as follow:

$$460 [S(t) + V(t) + CH_4(t) + CO_2(t)] \cdot L(t) \leq 12 \quad (43)$$

461 The performance index (PI) is given by:

$$462 PI = CH_4(t_f) \quad (44)$$

463
 464 **Table 12**
 465 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	$23 d$
$S(0)$	$4.75 g C dm^{-3}$
$V(0)$	$0 g C dm^{-3}$
$CH_4(0)$	$0 g C dm^{-3}$
$CO_2(0)$	$0 g C dm^{-3}$
$L(0)$	$2.4 dm^3$

466

467 4. Experiments and results

468 In this experiment, BSA is compared with four different metaheuristics: Covariance Matrix
469 Adaptation Evolution Strategy (CMA-ES) (Hansen & Ostermeier, 1996), Differential Evolution (DE)
470 (Storn & Price, 1997), Artificial Bee Colony (ABC) (Basturk, 2006) and Artificial Algae Algorithm (AAA)
471 (Uymaz et al., 2015). All the algorithms are population-based algorithm. In the context of fed-batch
472 fermentation processes optimization, the solutions found by the algorithms represent the trajectory
473 of input variables. The solutions or input variables are represented by $M \times (N + 1)$ real valued
474 vectors. M is the predetermined number of input variables. N is the predetermined size of input
475 variables or the number of feeding intervals. Each vector encodes an input variable as a temporal
476 sequence of values, defined as a piecewise linear function, with N equally spaced, linearly
477 interpolated segments. For the cases where there are more than one input variables, all the M
478 vectors are joined sequentially to create a solution. In this paper, all the case studies have only one
479 input variable except for case study II which has two input variables.

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

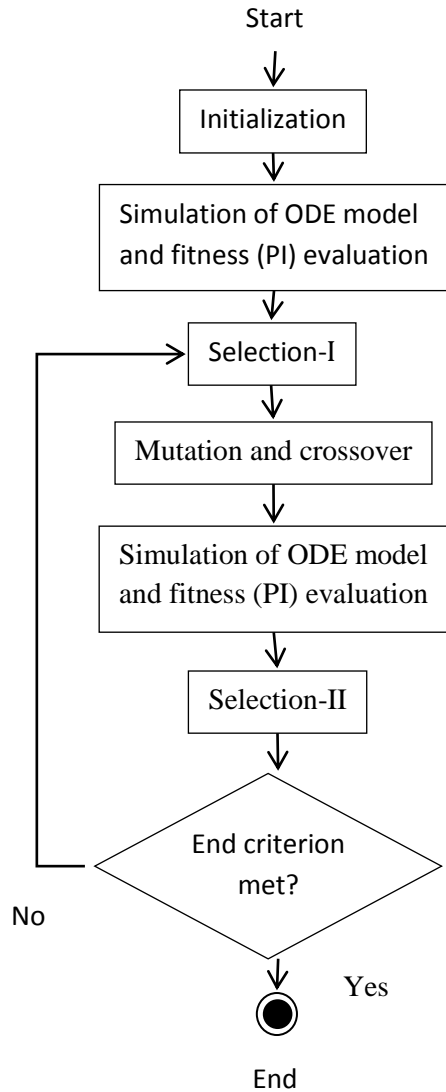


Fig. 3. BSA flowchart

487

488

489

490 4.1. Experimental analysis

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

500

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

p-Value	Condition	Symbol
$p \leq 0.001$	$\text{mean}(A1) > \text{mean}(A2)$	+++
$p \leq 0.001$	$\text{mean}(A1) < \text{mean}(A2)$	---
$0.001 < p \leq 0.01$	$\text{mean}(A1) > \text{mean}(A2)$	++
$0.001 < p \leq 0.01$	$\text{mean}(A1) < \text{mean}(A2)$	--
$0.01 < p \leq 0.05$	$\text{mean}(A1) > \text{mean}(A2)$	+
$0.01 < p \leq 0.05$	$\text{mean}(A1) < \text{mean}(A2)$	-
$p \geq 0.05$		O

503

504 *4.2. Parameter settings*

505 In our experiments, we use the standard parameters for each algorithm that were suggested
506 by previous studies. The termination condition is set after 200,000 FEs (function evaluations) and the
507 population size for all algorithms is 20. For DE in particular, the parameters are as follow: $F = 0.5$ and
508 $CR = 0.6$. The value of N is equal to the value of t_f in all cases except for case studies II and III (25
509 and 10 respectively).

510

511 *4.3. Results and discussion*

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

522

523 **Table 14**
524 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	20285 ± 30.73	20341 ± 26.56	20392 ± 14.26	20418 ± 4.71
AAA	20348 ± 10.42	20357 ± 14.87	20369 ± 9.91	20382 ± 7.02
ABC	7875 ± 2576	11258 ± 4605	20299 ± 61.62	20317 ± 36.98
DE	20384 ± 4.82	20381 ± 24.62	20388 ± 18.93	20406 ± 2.27
CMAES	20211 ± 100.2	20373 ± 46.09	20403 ± 29.87	20412 ± 30.03

525

526 **Table 15**
 527 T-test results for case study I.

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

528

529 In case study I, during the early stages of optimization, namely at 25,000 FEs, DE obtains the
 530 highest PI as shown in Table 14. Later, CMAES edged other algorithms to obtain better PI at 50,000
 531 and 100,000 FEs. However, at the saturation of optimization, BSA obtained the highest PI after
 532 200,000 FEs. According to the t-test in Table 15, BSA performed better than DE, AAA and ABC while
 533 performing equally well in comparison to CMAES.

534

535 **Table 16**
 536 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

537

538 **Table 17**
 539 T-test results for case study II.

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

540

541 In case study II, during the early stages of optimization namely at 25,000 FEs, DE obtains the
 542 highest PI as shown in Table 16. At 50,000 FEs, CMAES improved compared to other algorithms to
 543 obtain better PI though DE emerged to perform equally well as CMAES at 100,000 FEs to obtain the
 544 highest PI. At the saturation of optimization, BSA, DE and CMAES obtained the highest PI after 200,
 545 000 FEs. According to the t-test in Table 17, BSA performed better than AAA and ABC while
 546 performing equally well in comparison to CMAES and DE.

547

548

549 **Table 18**

550 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

551

552 **Table 19**

553 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	

554

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

559

560 **Table 20**

561 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

562

563 **Table 21**

564 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		
CMAES	O	O	O	O	

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

569

570 **Table 22**

571 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

572

573 **Table 23**

574 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	---	---	---	-	

575

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

581

582 **Table 24**

583 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

584

585

586 **Table 25**
 587 T-test results for case study VI.

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

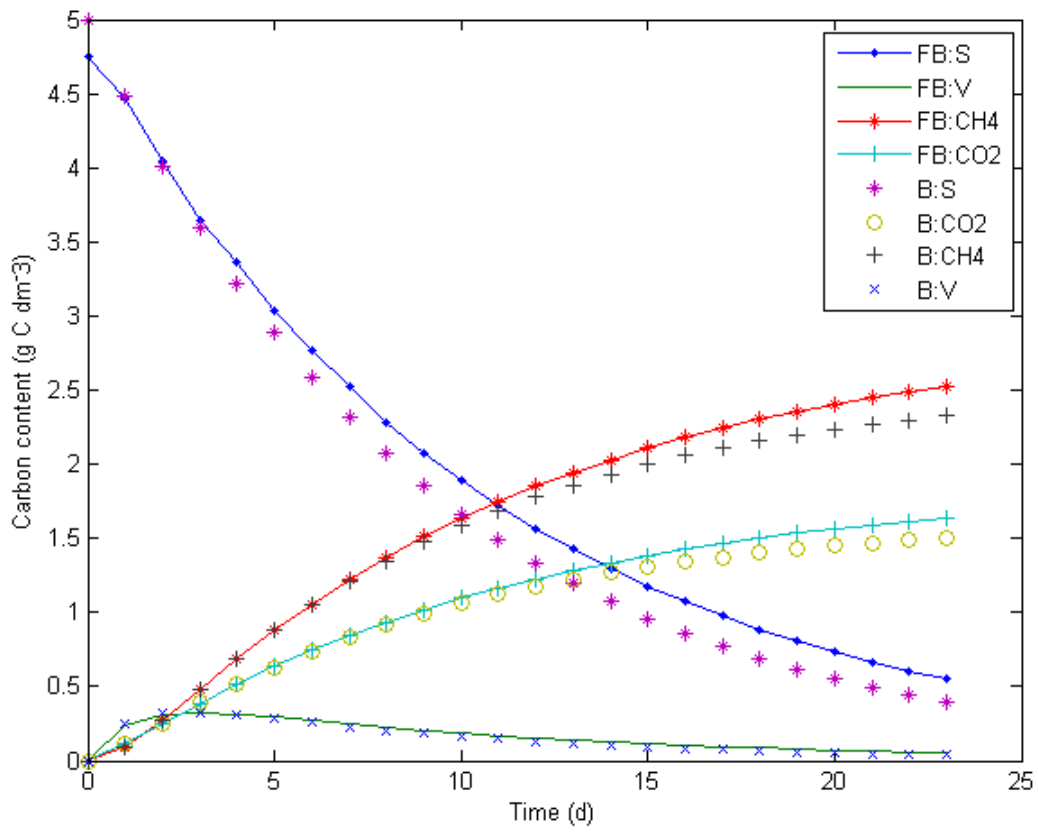
588

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

594

595 *4.3.1 Validation of batch results and improvement using fed batch for case study VI*

596 To show the improvements of fed-batch operation over batch in the methane production
 597 from sewage sludge fermentation, we ran a preliminary test for this model. Figure 4 shows the
 598 comparison of batch and fed-batch for sludge fermentation where FB stands for fed-batch while B
 599 stands for batch. The result for fed-batch was obtained from our preliminary simulation using the
 600 methodology described above and BSA as the optimization algorithm. We found that fed-batch
 601 produced 8.95% more methane compared to the conventional batch process. This improvement
 602 comes from the controlled feeding for each day during the fermentation process. The amount of
 603 methane produced by fed-batch starts to increase over batch after the ninth day. It is worth noting
 604 that fed-batch was able to produce more methane even when the initial substrate is less than the
 605 amount used in batch (4.75 g dm⁻³ for fed-batch compared to 5 g dm⁻³ for batch). Figure 5 shows the
 606 best feeding rate obtained by BSA for case VI.



607

608

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

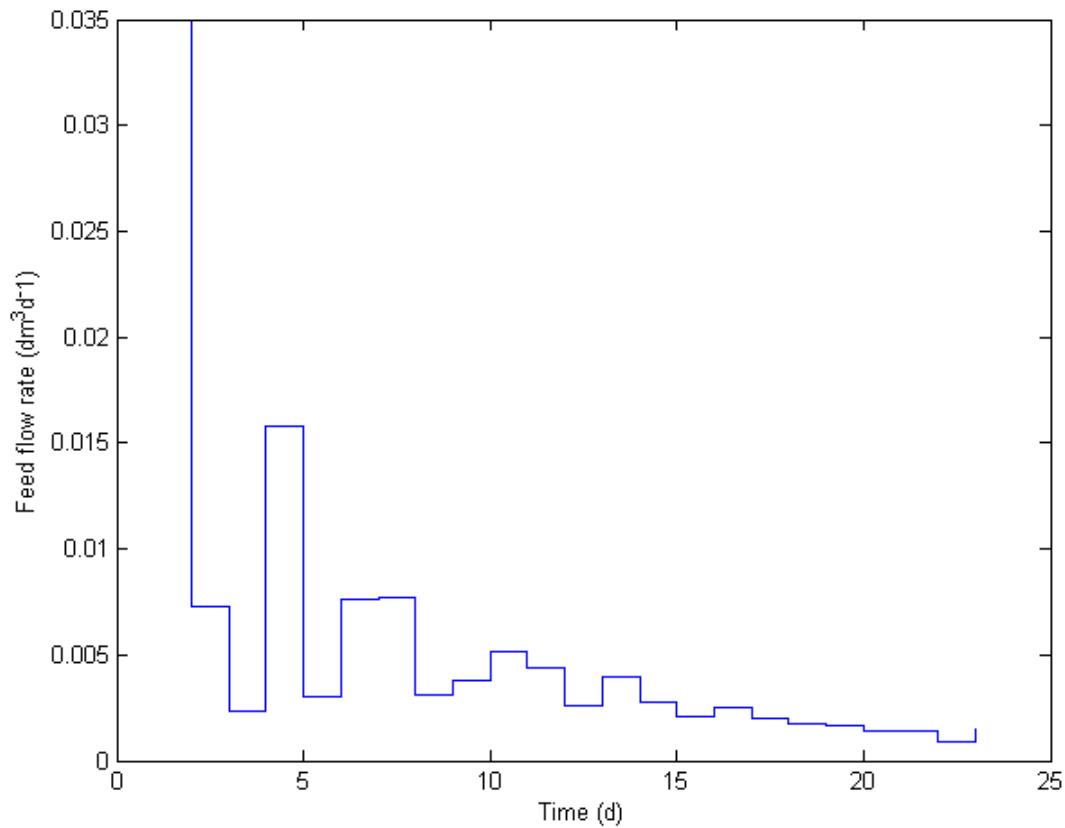


Fig. 5. Control profile for the fed-batch sludge fermentation

609

610

611

612 The results provide several insights on the capabilities of each algorithm in solving
 613 fermentation problems. The problems investigated in this paper can be divided into two categories:
 614 constrained and unconstrained. Case study II is unconstrained problem while the rest are
 615 constrained problems. For unconstrained problem, all algorithms performed almost equally well and
 616 saturated at almost the same PI value. This means that for unconstrained problems, there is
 617 flexibility in choosing an algorithm to solve a given problem as most of them converged to the same
 618 solution. However, a different scenario exists for constrained problems. For constrained problems,
 619 different algorithms performed differently in each problem with the exception of BSA. In overall, BSA
 620 is able to obtain the best results in all case studies by providing the highest means and narrow
 621 confidence interval. BSA obtained the highest means at 200,000 FEs for all problems except for case
 622 II where DE and CMAES saturated at the same highest value as BSA. Case V is an exception for
 623 constrained problem where AAA managed to obtain equal means as BSA. Even though DE and
 624 CMAES obtained higher means than BSA at NFE lower than 200,000 for some cases, BSA manages to
 625 obtain higher means than both algorithms at the end of 200,000 FEs for all constrained problems.
 626 This shows that when given a sufficient amount of NFE, BSA is the best option for solving
 627 constrained fermentation problems and provides improved performance compared to DE and other
 628 metaheuristics studied in this work for solving bioreactor application problems in general.

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

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

646 Based on the experimental results, all tested algorithms performed almost equally well for
647 the unconstrained problem. All algorithms converged at almost similar value for the unconstrained
648 problem at the end of the run. However, for constrained problems, which made up the majority of
649 the test problems in this work as well as assumed exist in real-life, we found that BSA is the best
650 performing algorithm. This is due to its high converging accuracy and better stability shown for all
651 the constrained problems. This outcome leads to the implication that BSA improves upon DE and is
652 suitable to be used for solving fed-batch bioreactor process problems.

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

670 **5. Conclusions**

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

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

686 Based on past literature, Differential Evolution (DE) is considered as a more appropriate
687 solution for bio-process applications. Since DE is known to be efficient in solving fermentation
688 problems, BSA as a recent DE-based metaheuristic is deemed to be superior to the former. Four
689 recent metaheuristics that included DE were applied on three bioprocess engineering problems
690 widely used in literature alongside with the problems mentioned above and the results were
691 compared with BSA. From the results, BSA showed consistency of obtaining highest fitness value in
692 comparison to other four metaheuristics for all the cases at convergence point. Therefore, BSA is
693 suggested as the first choice metaheuristic to use when solving bioprocess engineering problems.

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

699

700 **6. Acknowledgement**

701 This research work is supported by University of Malaya Research Grant (UMRG) RG 333-
702 15AFR.

703

704 **Bibliography**

- 705 Alberto Vieira Costa, J., Maria Colla, L., & Fernando Duarte Filho, P. (2004). Improving *Spirulina*
706 *platensis* biomass yield using a fed-batch process. *Bioresource Technology*, 92(3), 237-241.
- 707 Appels, L., Baeyens, J., Degève, J., & Dewil, R. (2008). Principles and potential of the anaerobic
708 digestion of waste-activated sludge. *Progress in Energy and Combustion Science*, 34(6), 755-
709 781.
- 710 Askarzadeh, A., & Coelho, L. d. S. (2014). A backtracking search algorithm combined with Burger's
711 chaotic map for parameter estimation of PEMFC electrochemical model. *International*
712 *Journal of Hydrogen Energy*, 39(21), 11165-11174.
- 713 Banga, J. R., Balsa-Canto, E., Moles, C. G., & Alonso, A. A. (2005). Dynamic optimization of
714 bioprocesses: Efficient and robust numerical strategies. *Journal of Biotechnology*, 117(4),
715 407-419.
- 716 Banga, J. R., Moles, C. G., & Alonso, A. A. (2004). Global Optimization of Bioprocesses using
717 Stochastic and Hybrid Methods. In C. A. Floudas & P. Pardalos (Eds.), *Frontiers in Global*
718 *Optimization* (pp. 45-70). Boston, MA: Springer US.
- 719 Basturk, B., Karaboga, D. (2006). An artificial bee colony (ABC) algorithm for numeric function
720 optimization. In *Proceedings of the IEEE Swarm Intelligence Symposium* (pp. 687-697),
721 Indianapolis, Indiana, USA. IEEE
- 722 Burke, E. K., & Kendall, G. (2014). *Search Methodologies - Introductory Tutorials in Optimization and*
723 *Decision Support Techniques* (2 ed.). New York: Springer US.
- 724 Chen, C.-T., & Hwang, C. (1990). Optimal control computation for differential-algebraic process
725 systems with general constraints. *Chemical Engineering Communications*, 97(1), 9-26.
- 726 Chen, L., Nguang, S. K., Chen, X. D., & Li, X. M. (2004). Modelling and optimization of fed-batch
727 fermentation processes using dynamic neural networks and genetic algorithms. *Biochemical*
728 *Engineering Journal*, 22(1), 51-61.
- 729 Chiou, J.-P., & Wang, F.-S. (1999). Hybrid method of evolutionary algorithms for static and dynamic
730 optimization problems with application to a fed-batch fermentation process. *Computers &*
731 *Chemical Engineering*, 23(9), 1277-1291.
- 732 Civicioglu, P. (2013). Backtracking search optimization algorithm for numerical optimization
733 problems. *Applied Mathematics and Computation*, 219(15), 8121-8144.
- 734 Da Ros, S., Colusso, G., Weschenfelder, T. A., de Marsillac Terra, L., de Castilhos, F., Corazza, M. L., &
735 Schwaab, M. (2013). A comparison among stochastic optimization algorithms for parameter
736 estimation of biochemical kinetic models. *Applied Soft Computing*, 13(5), 2205-2214.
- 737 Das, S., Mandal, D., Kar, R., & Ghoshal, S. P. (2014). Interference suppression of linear antenna arrays
738 with combined Backtracking Search Algorithm and Differential Evolution. In *International*
739 *Conference on Communications and Signal Processing (ICCSP)* (pp. 162-166).
- 740 del Rio-Chanona, E. A., Zhang, D., & Vassiliadis, V. S. (2016). Model-based real-time optimisation of a
741 fed-batch cyanobacterial hydrogen production process using economic model predictive
742 control strategy. *Chemical Engineering Science*, 142, 289-298.
- 743 Dinçer, A. R. (2004). Use of activated sludge in biological treatment of boron containing wastewater
744 by fed-batch operation. *Process Biochemistry*, 39(6), 723-730.
- 745 El-Fergany, A. (2015). Optimal allocation of multi-type distributed generators using backtracking
746 search optimization algorithm. *International Journal of Electrical Power & Energy Systems*,
747 64, 1197-1205.
- 748 Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. New York:
749 Addison-Wesley Publishing Company.
- 750 Goulden, C. H. (1956). *Methods of statistical analysis* (2nd ed.): John Wiley & Sons Ltd.
- 751 Guney, K., Durmus, A., & Basbug, S. (2014). Backtracking search optimization algorithm for synthesis
752 of concentric circular antenna arrays. *International Journal of Antennas and Propagation*,
753 2014.

754 Hansen, N., & Ostermeier, A. (1996). Adapting arbitrary normal mutation distributions in evolution
755 strategies: The covariance matrix adaptation. In *Proceedings of IEEE International*
756 *Conference on Evolutionary Computation* (pp. 312-317). IEEE.

757 Holm, S. (1979). A simple sequentially rejective multiple test procedure. *Scandinavian journal of*
758 *statistics*, 65-70.

759 J. Baeyens, L. Hosten, & E. Van Vaerenbergh. (1997). *Afvalwaterzuivering (Wastewater treatment)*
760 (2nd ed.). The Netherlands: Kluwer Academic Publishers.

761 Jayaraman, V. K., Kulkarni, B. D., Gupta, K., Rajesh, J., & Kusumaker, H. S. (2001). Dynamic
762 Optimization of Fed-Batch Bioreactors Using the Ant Algorithm. *Biotechnology Progress*,
763 17(1), 81-88.

764 Juma, C., & Konde, V. (2001). The new bioeconomy: industrial and environmental biotechnology in
765 developing countries. In *United Nations Conference on Trade and Development*.

766 Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In *IEEE Int. Conf. on Neural*
767 *Networks*. (pp. 1942–1948). Piscataway, NJ.

768 Kookos, I. K. (2004). Optimization of Batch and Fed-Batch Bioreactors using Simulated Annealing.
769 *Biotechnology Progress*, 20(4), 1285-1288.

770 Lee, J., & Ramirez, W. F. (1994). Optimal fed-batch control of induced foreign protein production by
771 recombinant bacteria. *AIChE Journal*, 40(5), 899-907.

772 Levišauskas, D., & Tekorius, T. (2015). Model-based optimization of fed-batch fermentation
773 processes using predetermined type feed-rate time profiles. A comparative study.
774 *Information Technology and Control*, 34(3).

775 Liu, C., Gong, Z., Shen, B., & Feng, E. (2013). Modelling and optimal control for a fed-batch
776 fermentation process. *Applied Mathematical Modelling*, 37(3), 695-706.

777 Montalvo, S., Guerrero, L., Rivera, E., Borja, R., Chica, A., & Martín, A. (2010). Kinetic evaluation and
778 performance of pilot-scale fed-batch aerated lagoons treating winery wastewaters.
779 *Bioresource Technology*, 101(10), 3452-3456.

780 Pelillo, M., Rincón, B., Raposo, F., Martín, A., & Borja, R. (2006). Mathematical modelling of the
781 aerobic degradation of two-phase olive mill effluents in a batch reactor. *Biochemical*
782 *Engineering Journal*, 30(3), 308-315.

783 Peng, W., Zhong, J., Yang, J., Ren, Y., Xu, T., Xiao, S., . . . Tan, H. (2014). The artificial neural network
784 approach based on uniform design to optimize the fed-batch fermentation condition:
785 application to the production of iturin A. *Microbial Cell Factories*, 13(1), 54.

786 Rocha, M., Mendes, R., Rocha, O., Rocha, I., & Ferreira, E. C. (2014). Optimization of fed-batch
787 fermentation processes with bio-inspired algorithms. *Expert Systems with Applications*,
788 41(5), 2186-2195.

789 Roubos, J. A., van Straten, G., & van Boxtel, A. J. B. (1999). An evolutionary strategy for fed-batch
790 bioreactor optimization; concepts and performance. *Journal of Biotechnology*, 67(2–3), 173-
791 187.

792 Sarkar, D., & Modak, J. M. (2004). Optimization of fed-batch bioreactors using genetic algorithm:
793 multiple control variables. *Computers & Chemical Engineering*, 28(5), 789-798.

794 Song, X., Zhang, X., Zhao, S., & Li, L. (2015). Backtracking search algorithm for effective and efficient
795 surface wave analysis. *Journal of Applied Geophysics*, 114, 19-31.

796 Sörensen, K. (2015). Metaheuristics-the metaphor exposed. *International Transactions in*
797 *Operational Research*, 22(1), 3-18

798 Sosnowski, P., Klepacz-Smolka, A., Kaczorek, K., & Ledakowicz, S. (2008). Kinetic investigations of
799 methane co-fermentation of sewage sludge and organic fraction of municipal solid wastes.
800 *Bioresource Technology*, 99(13), 5731-5737.

801 Storn, R., & Price, K. (1997). Differential Evolution – A Simple and Efficient Heuristic for global
802 Optimization over Continuous Spaces. *Journal of Global Optimization*, 11(4), 341-359.

- 803 Tholudur, A., & Ramirez, W. F. (1997). Obtaining smoother singular arc policies using a modified
804 iterative dynamic programming algorithm. *International Journal of Control*, 68(5), 1115-
805 1128.
- 806 Uymaz, S. A., Tezel, G., & Yel, E. (2015). Artificial algae algorithm (AAA) for nonlinear global
807 optimization. *Applied Soft Computing*, 31, 153-171.
- 808 Wang, F.-S., & Cheng, W.-M. (1999). Simultaneous Optimization of Feeding Rate and Operation
809 Parameters for Fed-Batch Fermentation Processes. *Biotechnology Progress*, 15(5), 949-952.
- 810 Yang, X. S. (2010). *Nature-inspired metaheuristic algorithms*. Beckington, UK: Luniver press.
- 811 Yang, X. S., & Suash, D. (2009). Cuckoo Search via Levy flights. In *World Congress on Nature &*
812 *Biologically Inspired Computing (NaBIC 2009)* (pp. 210-214).
- 813 Zhang, Y., & Banks, C. J. (2013). Impact of different particle size distributions on anaerobic digestion
814 of the organic fraction of municipal solid waste. *Waste management*, 33(2), 297-307.