

Article

Hybridized Particle Swarm—Gravitational Search Algorithm for Process Optimization

Rajendran Shankar¹, Narayanan Ganesh^{2,*} , Robert Čep³ , Rama Chandran Narayanan⁴ , Subham Pal⁵ 
and Kanak Kalita^{6,*} 

- ¹ Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram 522 302, India; shankarrajendran75@kluniversity.in
- ² Department of Computer Science and Engineering, Vel Tech Multi Tech Dr. Rangarajan Dr. Sakunthala Engineering College, Chennai 600 062, India
- ³ Department of Machining, Assembly and Engineering Metrology, Faculty of Mechanical Engineering, VSB-Technical University of Ostrava, 17. Listopadu 2172/15, 708 00 Ostrava, Czech Republic; robert.cep@vsb.cz
- ⁴ Department of Computer Science and Engineering, Sona College of Technology, Salem 636 005, India; narayananrc@sonatech.ac.in
- ⁵ Department of Aerospace Engineering and Applied Mechanics, Indian Institute of Engineering Science and Technology, Shibpur 711 103, India; subhampal982@gmail.com
- ⁶ Department of Mechanical Engineering, Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Avadi 600 062, India
- * Correspondence: ganeshn@veltechmultitech.org (N.G.); drkanakkalita@veltech.edu.in (K.K.)

Abstract: The optimization of industrial processes is a critical task for leveraging profitability and sustainability. To ensure the selection of optimum process parameter levels in any industrial process, numerous metaheuristic algorithms have been proposed so far. However, many algorithms are either computationally too expensive or become trapped in the pit of local optima. To counter these challenges, in this paper, a hybrid metaheuristic called PSO-GSA is employed that works by combining the iterative improvement capability of particle swarm optimization (PSO) and gravitational search algorithm (GSA). A binary PSO is also fused with GSA to develop a BPSO-GSA algorithm. Both the hybrid algorithms i.e., PSO-GSA and BPSO-GSA, are compared against traditional algorithms, such as tabu search (TS), genetic algorithm (GA), differential evolution (DE), GSA and PSO algorithms. Moreover, another popular hybrid algorithm DE-GA is also used for comparison. Since earlier works have already studied the performance of these algorithms on mathematical benchmark functions, in this paper, two real-world-applicable independent case studies on biodiesel production are considered. Based on the extensive comparisons, significantly better solutions are observed in the PSO-GSA algorithm as compared to the traditional algorithms. The outcomes of this work will be beneficial to similar studies that rely on polynomial models.

Keywords: optimization; non-traditional algorithms; process optimization; process parameters; algorithms



Citation: Shankar, R.; Ganesh, N.; Čep, R.; Narayanan, R.C.; Pal, S.; Kalita, K. Hybridized Particle Swarm—Gravitational Search Algorithm for Process Optimization. *Processes* **2022**, *10*, 616. <https://doi.org/10.3390/pr10030616>

Academic Editor: Jean-Pierre Corriou

Received: 26 January 2022

Accepted: 19 March 2022

Published: 21 March 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In the last few decades, many researchers have proposed numerous optimization techniques to find the best values among the sets of alternatives. Many researchers have used metaheuristic techniques to find the optimal values. Metaheuristic techniques are mostly optimization techniques inspired by nature and natural phenomena. Some examples of nature-inspired metaheuristic techniques are the genetic algorithm (GA) [1], particle swarm optimization (PSO) [2], artificial bee colony (ABC) [3], bat algorithm (BA) [4], gravitational search algorithm (GSA) [5], ant colony optimization (ACO) [6], differential evolution (DE) [7], ant lion optimization (ALO) [8], multi-verse optimization (MVO) [9], salp swarm algorithm (SSA) [10], grey wolf optimization (GWO) [11], dragonfly algorithm (DA) [12],

whale optimization algorithm (WOA) [13], firefly algorithm (FA) [14], etc. All population-based metaheuristic optimization algorithms have two main parts—exploration and exploitation. Some optimization techniques are very good at exploration, and some are very good at exploitation. So, an optimizer with good exploration and bad exploitation is often combined with an optimizer with bad exploration and good exploitation to form a hybrid optimizer. Thus, the hybrid optimizers are very good at both exploration and exploitation. Some hybrid metaheuristic techniques are GA-PSO [15], WOA-SA [16], CS-DE [17], PSO-CS [18], PSO-GSA [19], etc.

In this paper, as a case study, the industrial use case of biodiesel production is analyzed. The rationale behind this is that alternate fuel demand is at an all-time high. Biodiesel in this regard is a promising sector. However, to ensure the rapid adoption of biodiesel, it is necessary to improve the existing yields and bring down the associated costs. Though the biodiesel generation process is fairly standardized, the process parameters involved have a significant impact on the overall effectiveness of the process. Thus, as a test case for the hybrid algorithms, the biodiesel industry is focused on in this paper. Researchers have mostly relied on multi-criteria decision making (MCDM) methods for optimization in this industry. However, MCDMs methods need a decision matrix, that is made up of the responses to be optimized. Thus, the MCDM method is limited by the provided decision matrix that are often recorded on discrete points. MCDMs work by locating the best solution (in terms of some weighted measure of multiple conflicting alternatives) among the supplied discrete points. By level averaging of the process parameters, the best process parameter levels can also be identified. On the other hand, since metaheuristics directly use objective functions that can perform continuous search in the search space, they are likely to return better optimal values than MCDMs.

A few works on metaheuristics-based optimization of biodiesel production process are seen in the literature. Betiku et al. [20] performed the optimization of process parameters of biodiesel production from shea tree nut butter using the GA, neural network and response surface methodology. The objective of their work is to maximize the biodiesel yield. Garg and Jain [21] performed optimization of process parameters of biodiesel production from algal oil using RSM and ANN. They considered reaction time, catalyst amount, and methanol/oil ratio as process parameters and yield as a response parameter. Miraculas et al. [22] examined process parameter optimization for biodiesel production from mixed feedstock using an empirical model. Patil and Deng [23] optimized biodiesel production from edible and non-edible vegetable oils. Outili et al. [24] optimized biodiesel production from waste cooking oil. They considered temperature, catalyst amount, and methanol/oil ratio as independent variables and conversion, and energy and green chemistry balance as dependent variables to design the regression model. It is observed that for expressing responses as functions of process parameters, mostly RSM (polynomial regression) is used in the literature [20,22–26]. However, the use of neural networks has also gained traction in the last decade [21,27,28].

In this paper, two hybrid metaheuristic algorithms are developed by combining the iterative improvement capabilities of PSO and GSA. The hybrid algorithms are called PSO-GSA and BPSO-GSA and are compared against several traditional algorithms to demonstrate their efficacy. The BPSO is a genetic-algorithm-inspired binary version of the traditional PSO algorithm. Optimal process parameters of biodiesel production are studied by using these two hybrid algorithms. To build sufficient confidence in the derived conclusions, two independent case studies are considered. The performance of these two hybrid algorithms is compared with TS, GA, DE, GSA and PSO algorithms in terms of convergence, the solution quality as well as the computational time requirements. Another popular hybrid algorithm DE-GA is also used for comparisons. The rest of the paper is arranged as follows—the formulation of the metaheuristic algorithms is given in Section 2. Section 3 shows the experimental results of the optimal parameter evaluation of biodiesel production and compares the TS, GA, DE, GSA, PSO, DE-GA BPSOGSA and PSOGSA.

At last, the definitive conclusions, recommendations and the future scope are discussed in Section 4.

2. Methods

2.1. Particle Swarm Optimization

Particle swarm optimization was proposed by Kennedy and Eberhart in 1995 [2]. PSO is inspired by the collective behavior of social insects and other animal societies. PSO is inspired by the flocking of birds or fish schooling. Examples of swarms are the following: bees swarm around their hive, an ant colony has ants as individual agents, a flock of birds is a swarm of birds, the immune system is a swarm of cells, and a crowd is a swarm of people. PSO models the social behavior of bird flocking or fish schooling. Each particle is a potential candidate solution to the optimization problem and has a position and velocity associated with it. Particles change their position by adjusting their velocity to seek food, avoid predators, and identify optimized environmental parameters. In PSO, each particle memorizes the best location identified by it. Particles communicate the information regarding the best location explored by them. The mathematical model of particle swarm optimization is given as follows.

First, the initial population and the velocity of the particles are generated randomly. The particle velocity is given as follows [2]:

$$v_i^{t+1} = wv_i^t + c_1r_1(p_{best,i}^t - X_i^t) + c_2r_2(g_{best}^{t+1} - X_i^t) \quad (1)$$

where v_i^{t+1} and v_i^t are the velocities of the i^{th} particle at the $(t + 1)^{th}$ and t^{th} iterations, respectively, w is the inertia of the particles, and c_1 and c_2 are the acceleration coefficients. r_1 and r_2 are random numbers in between $[0, 1]$. $p_{best,i}^t$ is the personal best of the i^{th} particle at the t^{th} iteration. g_{best}^{t+1} is the global best of the i^{th} particle at the $(t + 1)^{th}$ iteration. X_i^t is the position vector of the i^{th} particle at the t^{th} iteration.

The position of the particle is updated, using the equation as follows:

$$X_i^{t+1} = X_i^t + v_i^{t+1} \quad (2)$$

Here, X_i^{t+1} is the position vector of the i^{th} particle at the $(t + 1)^{th}$ iteration.

After updating the position of all the particles, fitness is calculated using the current position vector. If the current fitness function gives better objective values, the particles remember the current position; otherwise, the particles remember the older best personal values. If the current position gives better values than the previous global best values, then the global value is updated. Otherwise, the older values of global best remain the same. The pseudocode for the PSO algorithm is given in Appendix A, Algorithm A1.

2.2. Gravitational Search Algorithm

The gravitational search algorithm is inspired by the law of gravity and mass interactions [5]. In GSA, the search agents are considered objects, and their masses measure their performance. Due to the gravitational force, all objects attract each other, which causes global movement of all objects toward the object with heavier masses [5]. Through the gravitational force, the heavy masses (considered a good solution) move slowly compared to the lighter masses. This shows the exploitation of the algorithm. In GSA, each mass has four specifications—position, inertial mass, active gravitational mass and passive gravitational mass.

The positions of a set of masses are represented as [5]

$$X_i = (x_i^1, \dots, x_i^d, \dots, x_i^n) \text{ for } i = 1, 2, \dots, N \quad (3)$$

where x_i^d presents the position of the i^{th} agent in the d^{th} dimension.

The gravitational force acting on mass i for the mass j is given as

$$F_{ij}^d(t) = G(t) \frac{M_{pi}(t) \times M_{aj}(t)}{R_{ij}(t) + \varepsilon} (x_j^d(t) - x_i^d(t)) \quad (4)$$

where M_{aj} represent the active gravitational mass related to agent j , and M_{pi} represents the passive gravitational mass related to agent i . $G(t)$ is the gravitational constant at time t , ε represents the small constant, and $R_{ij}(t)$ is the Euclidean distance between the two masses, i and j .

The G and R are calculated as follows:

$$G(t) = G_0 \times e^{(-\alpha \frac{t}{T})} \quad (5)$$

$$R_{ij}(t) = \|X_i(t), X_j(t)\|_2 \quad (6)$$

where α is a coefficient, G_0 shows the initial value of the gravitational constant, and t and T are the initial iteration and the maximum iteration, respectively.

To give a stochastic characteristic to the algorithm, the total force that acts on agent i in dimension d is a randomly weighted sum of d^{th} components of the forces exerted from other agents.

$$F_i^d(t) = \sum_{j=1, j \neq i}^N \text{rand}_j F_{ij}^d(t) \quad (7)$$

where rand_j is the random number in between the interval $[0, 1]$.

After calculating the forces and in order to move the solutions, the acceleration and velocities should be formulated as

$$a_i^d(t) = \frac{F_i^d(t)}{M_{ii}(t)} \quad (8)$$

$$v_i^d(t+1) = \text{rand}_i \times v_i^d(t) + a_i^d(t) \quad (9)$$

where $M_{ii}(t)$ is the inertial mass of agent i , and rand_i is the random number in between $[0, 1]$.

After calculating the acceleration and velocity, the position of the solutions can be updated as

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1) \quad (10)$$

where $x_i^d(t+1)$ is the the agent at the $(t+1)^{\text{th}}$ iteration, $x_i^d(t)$ is the position of the t^{th} iteration and $v_i^d(t+1)$ is the velocity at the $(t+1)^{\text{th}}$ iteration.

The mass of the solution is the fitness value calculated by the fitness function. So, a heavier mass is the best fitness function value.

The normalization of masses is calculated as follows:

$$m_i(t) = \frac{fit_i(t) - worst(t)}{best(t) - worst(t)} \quad (11)$$

$$M_i(t) = \frac{m_i(t)}{\sum_{j=1}^N m_j(t)} \quad (12)$$

where $fit_i(t)$ is the fitness values of the i^{th} agent at the t^{th} iteration, $worst(t)$ is the worst value at the t^{th} iteration and $best(t)$ is the best value at the t^{th} iteration. The pseudocode of GSA is presented in Appendix A, Algorithm A2.

2.3. Hybrid Particle Swarm Optimization and Gravitational Search Algorithm

The hybrid PSO-GSA algorithm was initially proposed by Mirjalili and Hashim [29], and the BPSO-GSA was proposed by Mirjalili et al. [30]. PSO-GSA combines the social

thinking ability of PSO with the exploration trait of GSA. PSOGSA is realized by using the pseudocode presented in Algorithm 1.

Algorithm 1: Pseudocode of PSOGSA

Inputs: Population size (N), Max. iteration number (t_{max})
while ($t < t_{max}$)
 Evaluate the fitness for each agent
 Update the G and g_{best} of the population
 For each search agent
 Calculate gravitational forces
 Calculate acceleration
 Update velocity
 Update position
 End
End
 $t = t + 1$
Return the g_{best} solution

BPSOGSA is realized by using the pseudocode presented in Algorithm 2.

Algorithm 2: Pseudocode of BPSOGSA

Inputs: Population size (N), Max. iteration number (t_{max})
while ($t < t_{max}$)
 Evaluate the fitness for each agent
 Update the G and g_{best} of the population
 For each search agent
 Calculate gravitational forces
 Calculate acceleration
 Update velocity
 Calculate probability of changing position vector's element
 Update position vector's element
 End
End
 $t = t + 1$
Return the g_{best} solution

3. Case Study 1: Process Optimization for Biodiesel Production

Optimization for biodiesel production from waste frying oil over montmorillonite clay K-30 was examined by Ayoub et al. [31]. They collected waste frying oil from a local restaurant in Bandar Sri Iskandar, Perak, Malaysia. They considered four process parameters (reaction temperature, reaction period, oil/methanol ratio, and amount of catalyst) for the biodiesel production yield. The central composite design (CCD) was selected for the design of experiments. Considering the CCD model, they performed 30 experiments. A second-order polynomial function [31] was modeled using the experimental data, which correlates the process parameters and response parameters.

$$R = 67.7038 - 3.2758x_1 + 4.2341x_2 + 4.7808x_3 - 1.93x_4 - 4.0325x_3x_4 - 4.4823x_2^2 \quad (13)$$

The range of process parameters was considered as, reaction temperature (x_1) between 400 and 1400 °C, reaction period (x_2) between 60 and 300 min, methanol/oil ratio (x_3) between 1:6 and 1:18 and amount of catalyst (x_4) between 1 and 5. However, Equation (13) is based on a coded form of the process parameters, wherein each process parameter is coded as -2 to $+2$ for the lower and upper bounds, respectively.

In this article, the optimal process parameters to maximize the palm oil yield are analyzed using TS, GA, DE, GSA, PSO, DEGA, BPSOGSA and PSOGSA optimizers. The

optimization is performed considering 30 search agents and a maximum iteration of 100. The convergence curves of yield % for TS, GA, DE, GSA, PSO, DEGA, BPSOGSA and PSOOGSA optimizers are shown in Figure 1.

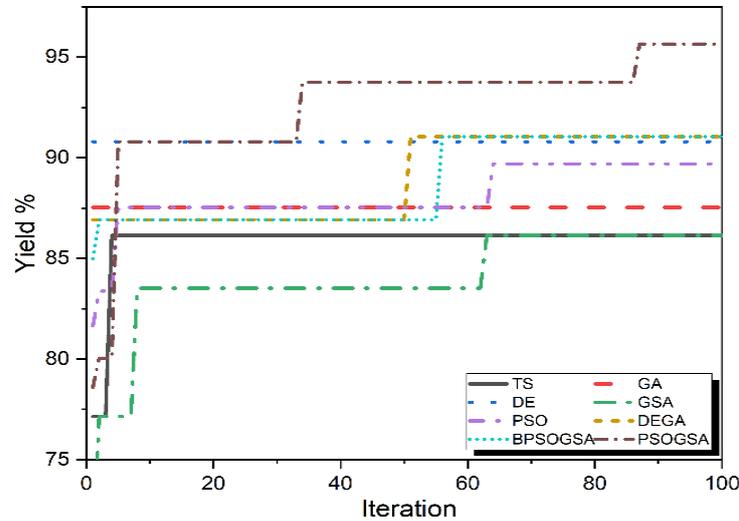


Figure 1. Convergence curve of GSA, PSO, BPSOGSA and PSOOGSA.

From Figure 1, it is seen that for GA and DE, there is no improvement in its best value even after 100 iterations. TS and GSA require very few iterations to locate their personal best values, which appear to be the local optima. DEGA and BPSOGSA have similar performance in terms of convergence to their respective best value. PSOOGSA, on the other hand, shows stepwise improvement in its best solution over the iterations. It converges at the best optimal value found so far by all the algorithms. The distribution of 3000 functional evaluation values for TS, GA, DE, GSA, PSO, DEGA, BPSOGSA and PSOOGSA is presented in Figure 2. From these scatters, it is observed that GSA, PSO, BPSOGSA and PSOOGSA have a maximum number of evaluated function values in the mid-range region. Further, PSOOGSA appears to have reached the best-known function value very few times. DEGA shows a good overall distribution of the evaluated function values.

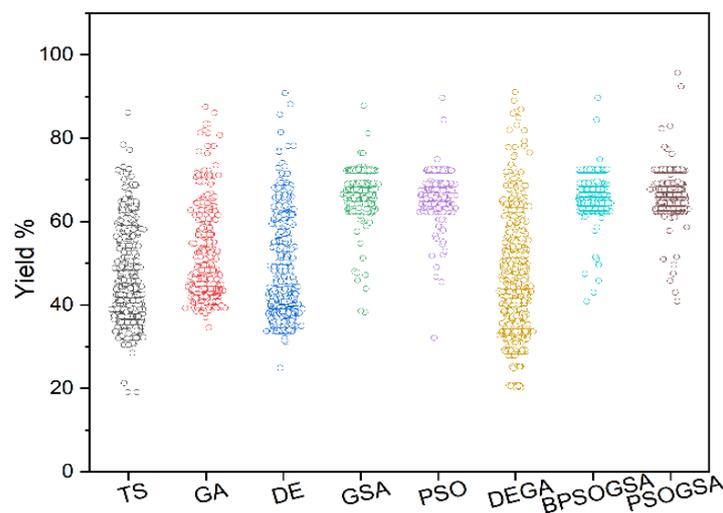


Figure 2. Total function evaluations during a typical optimization run.

For unbiased comparisons, each algorithm is independently run for 10 times. The performance analysis (best value, worst value, mean value, median value, and standard deviation value) of TS, GA, DE, GSA, PSO, DEGA, BPSOGSA and PSOOGSA optimizers

is presented in Table 1. From Table 1, it is seen that the maximum best value of yield is obtained by the PSOGSA algorithm (i.e., 95.646). Among the eight algorithms, PSOGSA generates better results than TS, GA, DE, GSA, PSO, DEGA and BPSOGSA. In PSOGSA, the functional values lie between 91.055 and 95.646, with a mean value of 93.455 and median value of 93.750. However, the standard deviation is the minimum for DE (i.e., 0.238) and low for BPSOGSA (i.e., 0.287).

Table 1. Performance of GSA, PSO, BPSOGSA and PSOGSA.

Optimizer	Best	Worst	Mean	Median	Standard Deviation
TS	87.614	83.687	85.675	86.017	1.354
GA	87.529	82.057	85.022	85.613	2.188
DE	90.930	90.188	90.539	90.512	0.238
GSA	87.796	83.520	86.138	86.143	1.441
PSO	89.697	87.514	88.147	87.529	0.833
DEGA	91.055	90.786	90.890	90.786	0.130
BPSOGSA	91.055	90.082	90.724	90.786	0.287
PSOGSA	95.646	91.055	93.455	93.750	1.520

Optimal values of process parameters and response using TS, GA, DE, GSA, PSO, DEGA, BPSOGSA and PSOGSA are presented in Table 2. The obtained results are compared with the previously published results by Ayoub et al. [31]. The maximum yield values are improved by 11.75%, 11.64%, 15.98%, 11.98%, 14.41%, 16.14%, 16.14% and 22.00% using TS, GA, DE, GSA, PSO, DEGA, BPSOGSA and PSOGSA, respectively. The PSOGSA optimizer gives the greatest improvement (i.e., 22.00%) on the previous known yield value.

Table 2. Optimal process parameter and responses.

Optimizer	Reaction Time	Reaction Temp	Catalyst Amount	Ratio O/M	Yield %	Improvement %
Ayoub et al. [31]	180.00	90.00	3.00	1:12	78.4	-
TS	75.00	101.00	4.87	1:11	87.614	11.75%
GA	75.00	101.00	4.87	1:10	87.529	11.64%
DE	97.20	109.25	4.46	1:8	90.930	15.98%
GSA	148.80	103.75	4.88	1:9	87.796	11.98%
PSO	262.80	108.75	4.86	1:7	89.697	14.41%
DEGA	196.80	120.00	4.79	1:6	91.055	16.14%
BPSOGSA	196.80	120.00	4.79	1:6	91.055	16.14%
PSOGSA	66.60	125.00	4.79	1:7	95.646	22.00%

4. Case Study 2: Multi-Objective Process Optimization for Biodiesel Production

Mostafaei et al. [32] used the response surface methodology (RSM) to optimize the ultrasonic-assisted continuous biodiesel production from waste cooking oil. They used CCD to design 50 experimental runs, using a combination of five process parameters. The process parameters were irradiation distance (x_1), UP amplitude (x_2), prob diameter (x_3), vibration pulse (x_4) and flow rate (x_5). The responses considered were yield (R_1) and energy consumption (R_2). It should be noted that the objective is to maximize yield (R_1) and minimize energy consumption (R_2). The true range of the process parameters was considered to be 30 to 90 mm for x_1 , 20% to 100% for x_2 , 14 to 42 mm for x_3 , 20% to 100% for x_4 and 40 to 80 mL/min for x_5 . However, the process parameters are coded as -2 to $+2$, and the following mathematical model for yield (R_1) and energy consumption (R_2) is used.

$$\begin{aligned}
 R_1 = & 90.14 + 0.6275x_1 + 0.8975x_2 + 0.6125x_3 + 0.6975x_4 - 0.5325x_5 - 0.428125x_1x_2 \\
 & + 0.353125x_1x_4 - 0.296875x_2x_3 - 0.571875x_2x_4 + 0.509375x_2x_5 - 0.178125x_3x_4 \\
 & + 0.665625x_4x_5 - 1.035625x_2^2 - 0.985625x_3^2 - 1.248125x_4^2 + 0.214375x_5^2
 \end{aligned} \quad (14)$$

$$\begin{aligned}
 R_2 = & 120.3575 - 10.2975x_1 + 18.4825x_2 + 29.3425x_3 + 14.6175x_4 - 0.032499x_5 - 1.321875x_1x_2 \\
 & - 2.509375x_1x_3 + 0.646875x_1x_5 + 3.921875x_2x_3 + 2.409375x_2x_4 - 1.021875x_2x_5 \\
 & + 3.884375x_3x_4 + 0.590625x_4x_5 - 3.428125x_1^2 + 0.809375x_2^2 - 0.965625x_3^2
 \end{aligned} \quad (15)$$

In this case study, the maximization of yield % (R_1) and minimization of power consumption (R_2) using TS, GA, DE, GSA, PSO, DEGA, BPSOGSA and PSOOGSA is carried out. The convergence curve of yield % and power consumption with the iterations is shown in Figure 3. The optimization is performed considering 30 search members and maximum iterations of 100. Figure 3a shows that despite the maximization of yield %, GSA and PSO did not show any improvement in the best value as the iterations progressed. On the other hand, the three remaining traditional algorithms, i.e., TS, GA and DE, converged to their respective best value in the 12th, 7th and 15th iterations. The hybrid algorithms, i.e., DEGA, BPSOGSA and PSOOGSA, converged to their respective best values in the 13th, 9th and 8th iterations. Thus, for the yield % objective, all the functions recorded very fast convergence to their respective best values. Nevertheless, it is worth mentioning that, except BPSOGSA and PSOOGSA, all the other algorithms converged to relatively poorer solutions. A similar trend is observed in Figure 3b, where convergence characteristics of the tested algorithm in the power consumption minimization scenario are reported. However, here, the DE, DEGA, BPSOGSA and PSOOGSA all were able to locate the best-known optimum, which may be the global optimum.

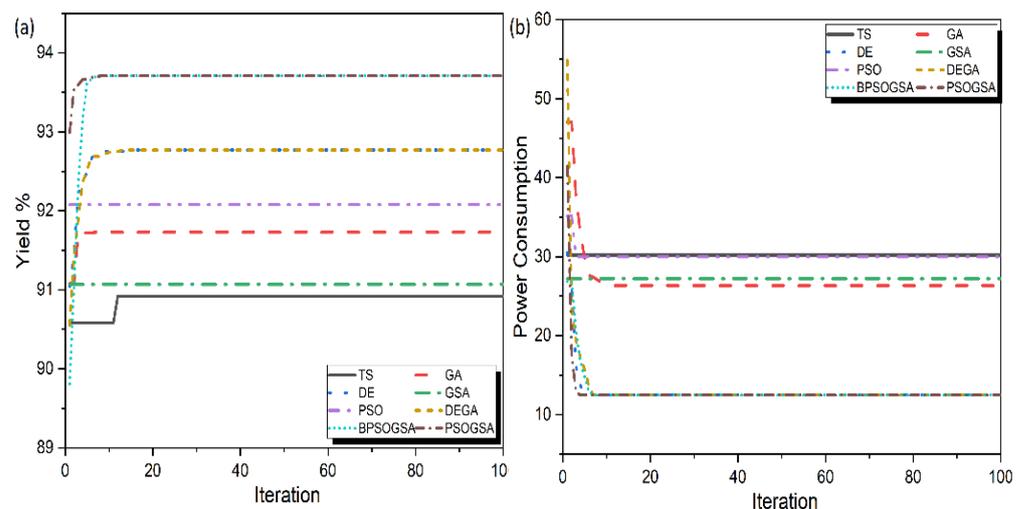


Figure 3. Convergence curve of various metaheuristics while optimizing (a) yield % (b) power consumption.

Scatter of the 3000 function evaluation values for TS, GA, DE, GSA, PSO, DEGA, BPSOGSA and PSOOGSA is presented in Figure 4. It is seen that the distribution pattern of DE, PSO, DEGA, BPSOGSA and PSOOGSA is somewhat similar. Major accumulations of function evaluation values are seen to be clustered in the higher yield % zone. On the other hand, the other three algorithms, i.e., TS, GA and GSA, showed throughout the distribution of the function evaluation values. Similar observations are made from Figure 4b, where GA, DE, GSA, DEGA, BPSOGSA and PSOOGSA showed good concentrations of solutions in the lower power consumption zone, indicating that the algorithms spent a relatively smaller number of function evaluations to locate the potential global optimal zone.

Table 3 presents the optimal process parameters and the achieved best value for yield % optimization by the eight algorithms. It is observed that both BPSOGSA and PSOOGSA were able to achieve the maximum yield % (i.e., 93.715). All the algorithms were able to locate good solutions for yield % maximization. Similarly, from Table 4, it is seen that all the three hybrid algorithms, i.e., DEGA, BPSOGSA and PSOOGSA, as well as DE were able to locate the best-known minimum power consumption value (i.e., 12.48).

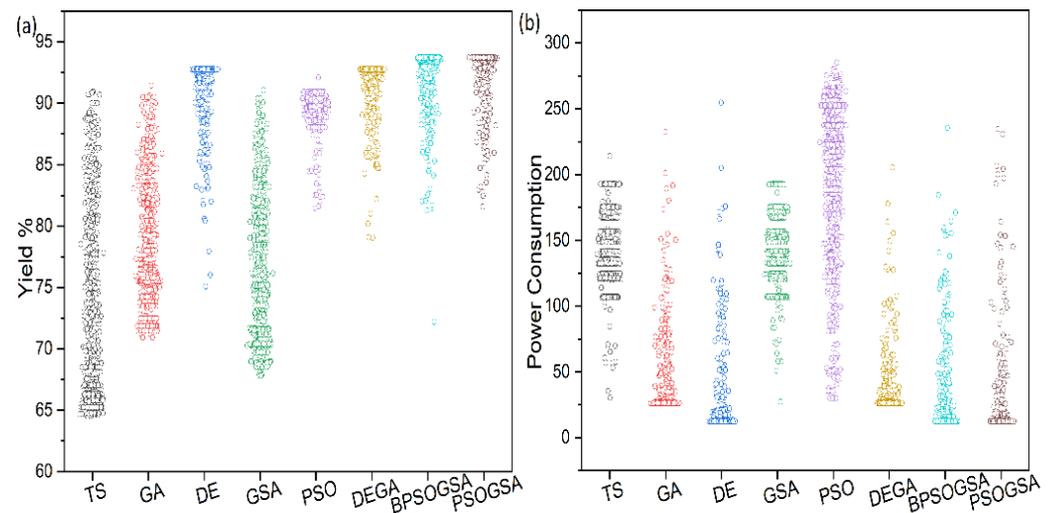


Figure 4. Distribution of total evaluated functions for various metaheuristics while optimizing (a) yield %, (b) power consumption.

Table 3. Optimal process parameter for optimization of yield %.

Optimizer	x_1	x_2	x_3	x_4	x_5	Response
TS	77.25	79.4	31.36	57.8	43.2	90.938
GA	84.6	58.4	23.38	59.2	44	91.731
DE	90	64	29.33	80.8	80	92.768
GSA	87	71.6	26.32	72.8	47.7	91.07
PSO	86.55	43.4	32.41	73.8	47.7	92.08
DEGA	90	64	29.33	80.8	80	92.768
BPSOGSA	90	48.8	30.66	62.6	40	93.715
PSOGSA	90	48.8	30.66	62.6	40	93.715

Table 4. Optimal process parameter for optimization of power consumption.

Optimizer	x_1	x_2	x_3	x_4	x_5	Response
TS	86.7	22.6	17.85	30.4	62.1	30.223
GA	30	20	14	20	80	26.34
DE	90	20	14	20	40	12.48
GSA	76.65	25.4	14.14	22.4	42.9	27.225
PSO	85.65	29.4	17.08	42.2	44.2	29.982
DEGA	90	20	14	20	40	12.48
BPSOGSA	90	20	14	20	40	12.48
PSOGSA	90	20	14	20	40	12.48

The ability of the hybrid algorithms to tackle multi-objective optimization problems is also assessed by considering a weighted sum multi-objective optimization wherein yield % and power consumption are simultaneously optimized. The composite objective function (Z) is described as

$$\text{Min } Z = \frac{0.5 * R_2}{(R_2)_{\min}} - \frac{0.5 * R_1}{(R_1)_{\max}} \quad (16)$$

where $(R_1)_{\max}$ and $(R_2)_{\min}$ are the best-known values of yield % and power consumption single-objective optimization.

The convergence of multi-objective function with respect to iteration and distribution scatter of 3000 function evaluations for TS, GA, DE, GSA, PSO, DEGA, BPSOGSA and PSOOGSA is presented in Figures 5 and 6, respectively.

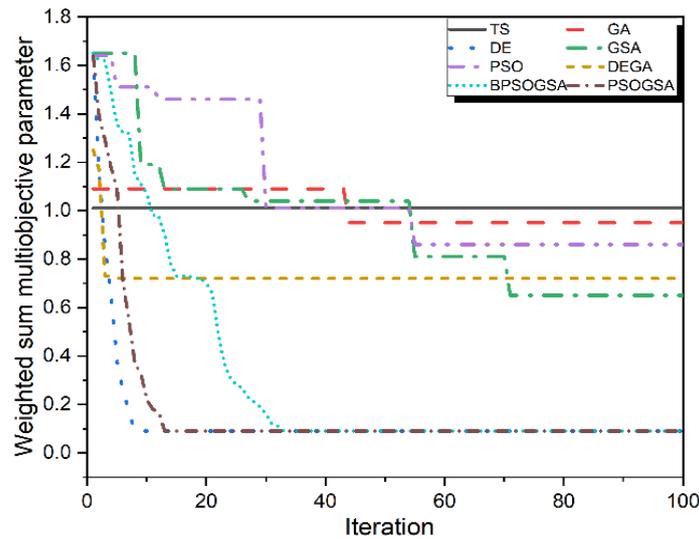


Figure 5. Convergence curve of algorithms for multi-objective optimization.

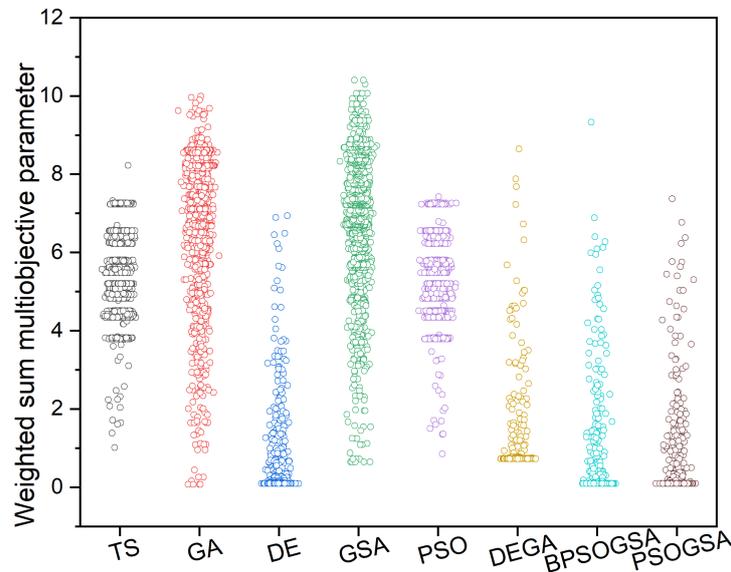


Figure 6. Scatter of total evaluated functions of algorithms for multi-objective optimization.

The performance analysis of the eight optimizers in the weighted sum multi-objective environment is given in Table 5. In multi-objective optimization, the minimum value of the function is obtained using PSOGSA, whereas for DEGA and BPSOGSA, the obtained solution is almost on par with that of PSOGSA.

Table 5. Performance of metaheuristics in multi-objective optimization.

Optimizer	Best	Worst	Mean	Median	Standard Deviation
TS	1.012	1.644	1.454	1.644	0.290
GA	0.952	1.644	1.358	1.447	0.304
DE	0.725	1.250	1.093	1.250	0.241
GSA	0.646	1.250	1.008	1.250	0.296
PSO	0.855	1.644	1.092	0.855	0.316
DEGA	0.091	1.459	0.775	0.775	0.684
BPSOGSA	0.091	0.725	0.281	0.091	0.291
PSOGSA	0.090	0.725	0.217	0.091	0.254

The optimal process parameters and responses are presented in Table 6. The obtained results are compared with the previously published results by Mostafaei et al. [32]. The optimal result is improved about 72.12%, 73.77%, 80.03%, 82.20%, 76.45%, 97.49%, 97.49% and 97.52% using TS, GA, DE, GSA, PSO, DEGA, BPSOGSA and PSO-GSA, respectively. However, it is important to state here that the derived multi-objective results are dependent on the considered weightage of the objectives. The composite objective function (Z) is a compromise solution that takes into account both yield (R_1) and energy consumption (R_2). This is why in the best solution derived by PSO-GSA, the R_1 dropped by about 15.7% whereas the R_2 improved by approximately 88%. The average computational time for each independent run of TS, GA, DE, GSA, PSO, DEGA, BPSOGSA and PSO-GSA is 2.010, 2.351, 2.411, 3.832, 2.407, 3.973, 4.771 and 4.684 s, respectively.

Table 6. Optimal process parameter for multi-objective optimization.

Optimizer	x_1	x_2	x_3	x_4	x_5	Z	R_1	R_2	% Improvement in Z
Mostafaei et al. [32]	75	56	28	62	50	3.630	91.6	102.8	-
TS	84	20.2	14.49	78.8	76	1.012	81.179	36.256	72.12%
GA	88.8	56	16.24	46.8	43.3	0.952	88.044	35.451	73.77%
DE	30	20	14	20	80	0.725	61.998	26.34	80.03%
GSA	86.2	27.6	17.57	26.4	44.7	0.646	81.018	26.940	82.20%
PSO	85.5	73.4	14.63	22.4	41.9	0.855	83.013	32.497	76.45%
DEGA	90	20	14	21.6	40	0.091	77.155	12.548	97.49%
BPSOGSA	90	20	14	20	40	0.091	76.638	12.480	97.49%
PSO-GSA	90	20	14	21.6	40	0.090	77.155	12.548	97.52%

5. Conclusions

To improve the performance and capabilities of industrial processes, it is crucial to optimize them. Most industrial processes are combinatorial optimization problems, where the performance or responses can be significantly improved by assigning optimized values to each process parameter. In this study, based on the two independent case studies on biodiesel production optimization, the following conclusions regarding the hybrid algorithms can be drawn.

- For case study 1, the tested algorithms can be ranked as PSO-GSA > DEGA > BPSOGSA > DE > PSO > GSA > TS > GA based on best solution and mean solution. For case study 2, considering all the three scenarios (i.e., yield % maximization, power consumption minimization and weighted sum multi-objective minimization), the ranking is PSO-GSA > BPSOGSA > DEGA > DE > GSA > GA > PSO > TS. Thus, in terms of quality of the optimum obtained, PSO-GSA, BPSOGSA and DEGA are the top three algorithms. However, in general, PSO-GSA is found to be roughly 1–2% better than BPSOGSA and DEGA for the tested case studies.
- Based on variance in the optimum, the algorithms can be ranked from least variance to most variance as DEGA > DE > BPSOGSA > PSO > TS > GSA > PSO-GSA > GA for case study 1. For case study 2, the ranking is DE > PSO-GSA > TS > BPSOGSA > GSA > GA > PSO > DEGA. Using an average ranking methodology and considering both case studies 1 and 2, the top three algorithms (i.e., lesser variance) are DE, BPSOGSA and TS.
- In terms of computational time, the hybrid algorithms are slower than their traditional counterparts. The algorithms can be ranked from fastest to slowest as TS > GA > PSO > DE > GSA > DEGA > PSO-GSA > BPSOGSA. With respect to TS, the GA, DE, GSA, PSO, DEGA, BPSOGSA and PSO-GSA are found to be 17%, 20%, 20%, 91%, 98%, 133% and 137% slower, respectively.
- With regards to the previous best known solution from the literature, the traditional algorithms improved the solution by an average of 13.15%, whereas the hybrid algorithm improved it by 18.09% for case study 1. Similarly, for case study 2, the improvements were seen to be 76.91% and 97.5% for traditional and hybrid algorithms, respectively.

Polynomial regression models are quite common in industrial process optimization research studies and applications. The outcomes of this study will be beneficial to any similar study that relies on second-order models. This study can be further improved by considering the probabilistic and non-probabilistic uncertainties of process parameters. Hybridization of the metaheuristics with MCDMs for multi-objective optimization is also likely to lead toward some interesting research.

Author Contributions: Conceptualization, R.S., N.G., R.Č., R.C.N. and K.K.; data curation, S.P.; formal analysis, S.P. and K.K.; methodology, R.S., N.G., R.Č. and R.C.N.; project administration, R.Č.; resources, R.S., N.G., R.Č. and R.C.N.; software, R.S., N.G. and R.C.N.; supervision, R.Č.; validation, S.P. and K.K.; visualization, S.P. and K.K.; writing—original draft, R.S., N.G., R.C.N. and S.P.; writing—review and editing, R.Č. and K.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available through email upon request to the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Algorithm A1: Pseudocode of PSO

Inputs: Population size (N), t_{max} (Max. iteration number)
 Initialize the random population (N), and initial random velocity (v)
 Evaluate the objective function values of N .
 Assign P_{best} and f_{pbest}
 Select the g_{best} and f_{gbest}
while ($t < t_{max}$)
 for $t = 1$ to N
 Determine the velocity (v_i) of i^{th} particle
 Determine the position (X_i) of i^{th} particle
 Bound X_i
 Evaluate the objective values of i^{th} particle
 if current fitness values are better than the older one
 then update P_{best} and f_{pbest}
 End
 if current fitness value is better among the entire population
 then update g_{best} and f_{gbest}
 End
 End
 $t = t + 1$
End
 Return the best solution

Algorithm A2: Pseudocode of GSA

Inputs: Population size (N), Max. iteration number (t_{max})
while ($t < t_{max}$)
 Evaluate the fitness for each agent
 Update the G , **best** and **worst** of the population
 Calculate M and a for each agent
 Update velocity and position
End
 $t = t + 1$
 Return the best solution

References

1. Yong-Jie, M.; Wen-Xia, Y. Research progress of genetic algorithm. *Appl. Res. Comput.* **2012**, *29*, 1201–1206.
2. Eberhart, R.; Kennedy, J. Particle swarm optimization. In Proceedings of the IEEE International Conference on Neural Networks, Perth, WA, Australia, 27 November–1 December 1995; Volume 4, pp. 1942–1948.
3. Karaboga, D. Artificial bee colony algorithm. *Scholarpedia* **2010**, *5*, 6915. [[CrossRef](#)]
4. Yang, X.S.; He, X. Bat algorithm: Literature review and applications. *Int. J. Bio-Inspired Comput.* **2013**, *5*, 141–149. [[CrossRef](#)]
5. Rashedi, E.; Nezamabadi-pour, H.; Saryazdi, S. GSA: A Gravitational Search Algorithm. *Inf. Sci.* **2009**, *179*, 2232–2248. [[CrossRef](#)]
6. Dorigo, M.; Blum, C. Ant colony optimization theory: A survey. *Theor. Comput. Sci.* **2005**, *344*, 243–278. [[CrossRef](#)]
7. Price, K.V. Differential Evolution. In *Handbook of Optimization*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 187–214. [[CrossRef](#)]
8. Mirjalili, S. The ant lion optimizer. *Adv. Eng. Softw.* **2015**, *83*, 80–98. [[CrossRef](#)]
9. Mirjalili, S.; Mirjalili, S.M.; Hatamlou, A. Multi-Verse Optimizer: A nature-inspired algorithm for global optimization. *Neural Comput. Appl.* **2016**, *27*, 495–513. [[CrossRef](#)]
10. Mirjalili, S.; Gandomi, A.H.; Mirjalili, S.Z.; Saremi, S.; Faris, H.; Mirjalili, S.M. Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems. *Adv. Eng. Softw.* **2017**, *114*, 163–191. [[CrossRef](#)]
11. Faris, H.; Aljarah, I.; Al-Betar, M.A.; Mirjalili, S. Grey wolf optimizer: A review of recent variants and applications. *Neural Comput. Appl.* **2018**, *30*, 413–435. [[CrossRef](#)]
12. Mirjalili, S. Dragonfly algorithm: A new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. *Neural Comput. Appl.* **2016**, *27*, 1053–1073. [[CrossRef](#)]
13. Mirjalili, S.; Lewis, A. The Whale Optimization Algorithm. *Adv. Eng. Softw.* **2016**, *95*, 51–67. [[CrossRef](#)]
14. Yang, X.S.; He, X. Firefly algorithm: Recent advances and applications. *Int. J. Swarm Intell.* **2013**, *1*, 36–50. [[CrossRef](#)]
15. Kuo, R.J.; Lin, L.M. Application of a hybrid of genetic algorithm and particle swarm optimization algorithm for order clustering. *Decis. Support Syst.* **2010**, *49*, 451–462. [[CrossRef](#)]
16. Mafarja, M.M.; Mirjalili, S. Hybrid Whale Optimization Algorithm with simulated annealing for feature selection. *Neurocomputing* **2017**, *260*, 302–312. [[CrossRef](#)]
17. Zhang, Z.; Ding, S.; Jia, W. A hybrid optimization algorithm based on cuckoo search and differential evolution for solving constrained engineering problems. *Eng. Appl. Artif. Intell.* **2019**, *85*, 254–268. [[CrossRef](#)]
18. Guo, J.; Sun, Z.; Tang, H.; Jia, X.; Wang, S.; Yan, X.; Ye, G.; Wu, G. Hybrid Optimization Algorithm of Particle Swarm Optimization and Cuckoo Search for Preventive Maintenance Period Optimization. *Discret. Dyn. Nat. Soc.* **2016**, *2016*, 1516271. [[CrossRef](#)]
19. Radosavljević, J.; Klimenta, D.; Jevtić, M.; Arsić, N. Optimal Power Flow Using a Hybrid Optimization Algorithm of Particle Swarm Optimization and Gravitational Search Algorithm. *Electr. Power Compon. Syst.* **2015**, *43*, 1958–1970. [[CrossRef](#)]
20. Betiku, E.; Okunsolawo, S.S.; Ajala, S.O.; Odedele, O.S. Performance evaluation of artificial neural network coupled with generic algorithm and response surface methodology in modeling and optimization of biodiesel production process parameters from shea tree (*Vitellaria paradoxa*) nut butter. *Renew. Energy* **2015**, *76*, 408–417. [[CrossRef](#)]
21. Garg, A.; Jain, S. Process parameter optimization of biodiesel production from algal oil by response surface methodology and artificial neural networks. *Fuel* **2020**, *277*, 118254. [[CrossRef](#)]
22. Miraculas, G.A.; Bose, N.; Raj, R.E. Process parameter optimization for biodiesel production from mixed feedstock using empirical model. *Sustain. Energy Technol. Assess.* **2018**, *28*, 54–59. [[CrossRef](#)]
23. Patil, P.D.; Deng, S. Optimization of biodiesel production from edible and non-edible vegetable oils. *Fuel* **2009**, *88*, 1302–1306. [[CrossRef](#)]
24. Outili, N.; Kerras, H.; Nekkab, C.; Merouani, R.; Meniai, A.H. Biodiesel production optimization from waste cooking oil using green chemistry metrics. *Renew. Energy* **2020**, *145*, 2575–2586. [[CrossRef](#)]
25. Ghadai, R.K.; Kalita, K.; Mondal, S.C.; Swain, B.P. Genetically optimized diamond-like carbon thin film coatings. *Mater. Manuf. Process.* **2019**, *34*, 1476–1487. [[CrossRef](#)]
26. Singh, A.; Ghadai, R.K.; Kalita, K.; Chatterjee, P.; Pamučar, D. EDM process parameter optimization for efficient machining of Inconel-718. *Facta Univ. Ser. Mech. Eng.* **2020**, *18*, 473–490. [[CrossRef](#)]
27. Mahmood, M.A.; Popescu, A.C.; Oane, M.; Channa, A.; Mihai, S.; Ristoscu, C.; Mihailescu, I.N. Bridging the analytical and artificial neural network models for keyhole formation with experimental verification in laser melting deposition: A novel approach. *Results Phys.* **2021**, *26*, 104440. [[CrossRef](#)]
28. Mahmood, M.A.; Visan, A.I.; Ristoscu, C.; Mihailescu, I.N. Artificial neural network algorithms for 3D printing. *Materials* **2021**, *14*, 163. [[CrossRef](#)]
29. Mirjalili, S.; Hashim, S.Z.M. A new hybrid PSO-GSA algorithm for function optimization. In Proceedings of the 2010 International Conference on Computer and Information Application, Tianjin, China, 3–5 December 2010; pp. 374–377.
30. Mirjalili, S.; Wang, G.G.; Coelho, L.D.S. Binary optimization using hybrid particle swarm optimization and gravitational search algorithm. *Neural Comput. Appl.* **2014**, *25*, 1423–1435. [[CrossRef](#)]
31. Ayoub, M.; Ullah, S.; Inayat, A.; Bhat, A.H.; Hailegiorgis, S.M. Process Optimization for Biodiesel Production from Waste Frying Oil over Montmorillonite Clay K-30. *Procedia Eng.* **2016**, *148*, 742–749. [[CrossRef](#)]
32. Mostafaei, M.; Ghobadian, B.; Barzegar, M.; Banakar, A. Optimization of ultrasonic assisted continuous production of biodiesel using response surface methodology. *Ultrason. Sonochem.* **2015**, *27*, 54–61. [[CrossRef](#)]