processes

MDPI

Article

Improved Seagull Optimization Algorithm Combined with an Unequal Division Method to Solve Dynamic Optimization Problems

Le Xu [1], Yuanbin Mo [1,2,*], Yanyue Lu [3] and Jiang Li [1]

[1] Institute of Artificial Intelligence, Guangxi University for Nationalities, Nanning 530006, China; lexu_stu@163.com (L.X.); ljzhumeng@163.com (J.L.)
[2] Guangxi Key Laboratory of Hybrid Computation and IC Design Analysis, Guangxi University for Nationalities, Nanning 530006, China
[3] School of Chemistry and Chemical Engineering, Guangxi University for Nationalities, Nanning 530006, China; luyanyue@163.com
* Correspondence: moyuanbin2020@gxun.edu.cn

Abstract: The numerical solution of the dynamic optimization problem is often sought for chemical processes, but the discretization of control variables is a difficult problem. Firstly, based on the analysis of the seagull optimization algorithm, this paper introduces the cognitive part in the process of a seagull's attack behavior to make the group approach the best position. Secondly, the algorithm adds the mechanism of natural selection, where the fitness value is used to sort the population, and the best half is used to replace the worst half, so as to find out the optimal solution. Finally, the improved seagull optimization algorithm (ISOA) is combined with the unequal division method to solve dynamic optimization problems. The feasibility of the method is verified by three practical examples of dynamic optimization in chemical industry.

Keywords: chemical processes; chemical dynamic optimization problem; numerical solution; seagull optimization algorithm

check for updates

Citation: Xu, L.; Mo, Y.; Lu, Y.; Li, J. Improved Seagull Optimization Algorithm Combined with an Unequal Division Method to Solve Dynamic Optimization Problems. *Processes* **2021**, *9*, 1037. https://doi.org/10.3390/pr9061037

Academic Editor: Jean-Pierre Corriou

Received: 16 April 2021
Accepted: 8 June 2021
Published: 14 June 2021

1. Introduction

With the continuous development of society, the optimization of chemical processes, such as batch reaction process and tubular reaction process, has always been a research hotspot. Improving chemical efficiency and reducing energy consumption are the goals of chemical industry. However, as the chemical engineering model becomes more and more complex and the number of control variables increases, the stability of the model becomes lower, and inputs and outputs are related by nonlinear models. Therefore, we need to propose a new method to optimize it. Chemical dynamic optimization problems are usually complicated, which seeks one or more control variables to optimize the performance index of the objective function. At present, direct methods [1] and indirect methods [2] are often used to solve the optimal control problem. The direct method is to transform the continuous optimal control problem into a finite-dimensional nonlinear programming problem for solution through discretization. The indirect method mainly solves the optimality condition (necessary condition) of the original problem, thereby indirectly obtaining the optimal solution of the original problem. So far, the optimization methods for chemical engineering problems mainly include dynamic programming (DP) [3], control variable parameterization (CVP) [4], and also swarm intelligence algorithm [5,6].

Dynamic programming does not require gradient information, but it faces the difficult question of dimensionality. In 1989, Luus [7] proposed the iterative dynamic programming method, which mainly included the introduction of area reduction processing and the iterative application of dynamic programming. At the same time, the iterative dynamic programming algorithm needs to simultaneously discretize the two dimensions of time and space, and obtain the optimal solution through iterative calculation. However, if the dimension of time and space are relatively large, the implementation process of this method is very complicated.

In 1970, Sargent et al. [8] proposed the CVP algorithm for the first time. The main idea of the algorithm is the subsection control approximation strategy. At the same time, in 1979, Sargent et al. [9] used the CVP algorithm to solve dynamic optimization problems for the first time, and many people knew the algorithm. In recent years, the CVP algorithm has been improved by many people. Xu et al. [10] proposed a dynamic optimization problem grid reconstruction strategy based on pseudo Wigner–Ville analysis, using Wigner–Ville to perform grid reconstruction combined with variable time node CVP method to achieve switching between time nodes. Li et al. [11] proposed a variable time node control vector parameterization method to optimize the control variable and time node. Li et al. [12] proposed a control parameterization method with variable time nodes, using a multi-time grid control vector parameterization method, so that each control variable had a corresponding time grid. Binder et al. [13] introduced wavelet analysis to further fine-tune the grid divided by the CVP method, so that the dynamic optimization trajectory was closer to the true value.

In recent years, intelligent algorithms have been widely used to solve dynamic optimization cases. The algorithm has the advantages of simplicity, ease of implementation, and strong search capabilities, so it is also a hot topic for scholars to study. Based on particle swarm algorithm (PSO) and control variable parameterization, Shi [14] et al. proposed to solve the dynamic optimization of the chemical process. First, the particle swarm was used for the first optimization, and then the optimized value was passed to the control variable parameter method for the second optimization. Xu et al. [15] proposed a biogeographic learning particle swarm algorithm to improve the efficiency of learning. Sarkar et al. [16] used genetic algorithms to optimize feed bioreactors: multiple control variables. Anand et al. [17] proposed a tabu search method to dynamically optimize the copolymerization reactor. Schluter et al. [18] proposed the design of an integrated process and control system based on the ant colony optimization algorithm. Egea et al. [19] proposed an enhanced scatter search method for the global dynamic optimization of nonlinear processes. Nikumbh [20] proposed biogeography-based optimization for dynamic optimization of chemical reactors.

In this study, we propose a new improvement strategy for the seagull optimization algorithm, which mainly includes the improvement strategy of the cognitive part and the natural selection mechanism. Finally, by combining the improved seagull optimization algorithm (ISOA) with the unequal segmentation method, three typical chemical examples are solved, and different methods are compared and analyzed. Each chemical case has a good performance index, and the results show the optimization ability of the algorithm.

The main contributions and innovations of this research are as follows: (1) The cognitive part strategy is introduced to prevent the algorithm from entering the local optimum. (2) The use of the natural selection mechanism further enhances the optimization performance of the seagull optimization algorithm, so as to find the optimal solution faster. (3) The improved seagull optimization algorithm combined with the unequal division method is used to solve dynamic optimization problems. (4) In order to verify the feasibility of the improved algorithm, this study introduces three chemical engineering cases for testing. At the same time, we analyze the performance indicators of the three cases and compare them with other methods. Finally, we also compare the execution efficiency of the equal division method and the unequal division method.

2. Problem Description and Unequal Division Method

2.1. Problem Description

A typical dynamic optimization problem (DOP) of a chemical engineering process can be stated as follows:

$$\min J = \Phi\left[x\left(t_f\right)\right] + \int_{t_0}^{t_f} \Psi[x(t), u(t)]dt, \tag{1}$$

$$s.t.\begin{cases} \dot{x} = f[x(t), u(t), t] \\ u_{\min} \le u(t) \le u_{\max} \\ x_{\min} \le x(t) \le x_{\max} \\ x(0) = x_0 \end{cases}, \tag{2}$$

where t_0, t_f are the initial time and terminal time, respectively, $x(t)$ is the state vector; $u(t)$ is the control vector; J is performance index; $\Phi[x(t_f)]$ is terminal value function, and $\Psi[x(t), u(t)]$ represents integral function; $u_{\max}$ and $u_{\min}$ are the lower and upper bounds of the control vector, respectively; $x_{\min}$ and $x_{\max}$ are the minimum and maximum values of the state vector, respectively; $x(0)$ is the initial state, and f is a dynamic system equation.

2.2. Unequal Division Method

The piecewise constant method is the most common method in control variable parameterization (CVP). Within a time domain $[t_0, t_f]$, which is equally divided into n subregions using the equal division method, i.e., the length of each subregion is $d = (t_f - t_0)/n$, the Runge–Kutta method can be used to integrate the ordinary differential equations of the model. At the same time, the corresponding algorithm is used to optimize the control variables, so that the control variables produce a series of trajectories. Compared with the equal division method, the unequal division method may produce a more precise control trajectory, and thus obtain better performance indicators, therefore, we use the unequal division method. The steps for unequal division are as follows:

① We assume that the time domain needs to be divided into n segments, a set of parameters are randomly initialized in $[t_0, t_f]$, namely $\tau_1, \tau_2, \ldots, \tau_n \in [t_0, t_f]$;

② The final time parameter is as follows, t_i represents time parameter.

$$t_i = t_0 + \frac{\left(t_f - t_0\right) \times \sum_{k=1}^{i} \tau_k}{\sum_{i=1}^{n} \tau_i}, i = 1, 2, \ldots, n \tag{3}$$

3. Seagull Optimization Algorithm (SOA)

Inspired by the migration and attack behavior of seagulls, Dhiman et al. [21] proposed a random search algorithm based on swarm intelligence—seagull optimization algorithm (SOA). When using the seagull optimization algorithm, assume that the size of the seagull population is pop, the dimension of the problem space is D, and the position of the seagull is $X_i = ({x_i}^1, {x_i}^2, \ldots, {x_i}^D)$, $i = 1, 2, \ldots, pop$. Finally, the migration and attack behavior of seagulls are the process of individual position updating. The seagull migration and attack process are shown in Figure 1.

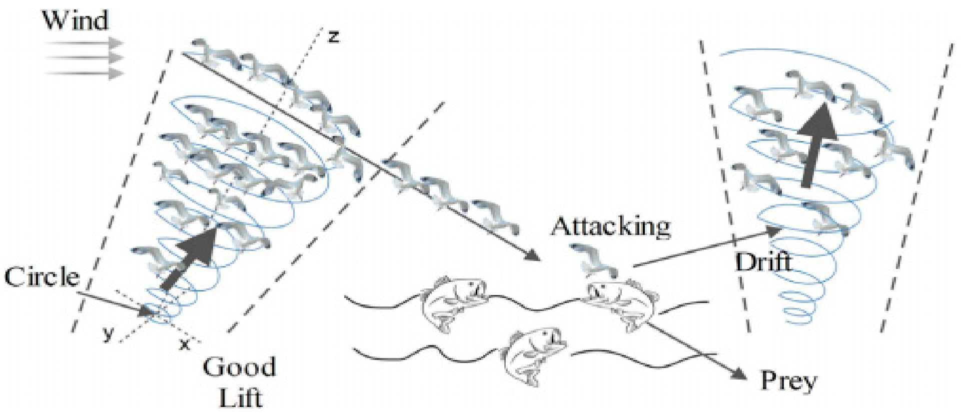

Figure 1. The seagull migration and attack process.

3.1. Migration Behavior

The migration behavior belongs to the global search. Migration refers to the movement of seagulls from one position to another, but three conditions that seagulls should meet during migration are avoiding collisions, direction of the best position, and approaching the best position.

① Avoiding collisions: In order to prevent collisions between adjacent seagulls, an additional variable A is introduced. At the same time, the algorithm uses the variable A to update the position of the seagull in the iterative process.

$$cs(iter) = A \times ps(iter), \tag{4}$$

where $cs(iter)$ is the new position of the seagull after collision avoidance, $iter$ is the current iteration number, $ps(iter)$ is the initial position of the seagull, and A is the motion behavior of the seagull in a given search space. The equation of A is as follows:

$$A = f_c - (iter \times (f_c / Maxiter)), \tag{5}$$

where the value of A is adjusted linearly, the value of f_c decreases linearly to 0, and $Maxiter$ is the maximum number of iterations.

② Direction of the best position: After the seagull is satisfied that it will not collide with other individuals, the seagull will move in the direction of the best position. The equation is as follows:

$$ms(iter) = B \times (zbest(iter) - ps(iter)), \tag{6}$$

$$B = 2 \times A \times A \times rd, \tag{7}$$

where B is a random number that balances global search and local search, $zbest(iter)$ represents the best position in the population, and the value of rd is a random number of (0,1).

③ Approaching the best position: After the seagull moves to a position where it does not collide with other seagulls, it moves in the direction of the best position to reach a new position.

$$ds(iter) = |cs(iter) + ms(iter)|, \tag{8}$$

where $ds(iter)$ represents the best fit search seagull.

3.2. Attack Behavior

The attack behavior belongs to the local search. When seagulls attack their prey, they move in a spiral shape in the air, using x, y, and z to represent their motion behavior. The equation for exercise behavior is as follows:

$$x = r \times \cos(\theta), \tag{9}$$

$$y = r \times \sin(\theta), \tag{10}$$

$$z = r \times \theta, \tag{11}$$

$$r = \mu \times e^{\theta v}, \tag{12}$$

where r is the spiral radius in the motion of the seagull, and μ, v are the correlation constants of the spiral shape. θ represents the angle, which is a random number in $[0, 2\pi]$. The equation for the attacking behavior of seagulls is as follows:

$$ps(iter) = ds(iter) \times x \times y \times z + zbest(iter), \tag{13}$$

where $ps(iter)$ saves the best solution and updates the position of other search seagulls.

3.3. The Steps of the Seagull Optimization Algorithm

Step 1: Initialize the size of the seagull population pop, set the parameters mainly including f_c, u, v, θ and $Maxiter$, and initialize the position of pop seagulls randomly.

Step 2: Calculate the fitness value of each seagull, compare the fitness values between individuals, and find global optimal values of the current population.

Step 3: Enter the main loop and use Equations (4)–(8) to calculate the new position of the seagull after migrating.

Step 4: Use Equations (9)–(13) to calculate the final position of the seagull.

Step 5: Compare the fitness between individuals in the current seagull population again to find the global optimal value.

Step 6: Judging that the algorithm reaches the termination condition during the execution process. If it is reached, it ends; otherwise, the calculation goes to Step 4–Step 6 to continue the position update.

Step 7: Output the global optimal position and fitness value of the SOA algorithm.

4. Improved Seagull Optimization Algorithm

4.1. Cognitive Part

The SOA has the advantages of a good optimization effect, simple operation, and few parameter settings. However, the SOA has a premature phenomenon, and it is easy to fall into a local optimum. For Equation (13), it belongs to the local search stage, which is easy to fall into a local extremum, and may not find the global optimal value. Therefore, we introduce the cognitive part into Equation (13) to reduce or jump out of the local extremum. The improvement of the equation is as follows:

$$\begin{aligned} ps(iter) = ds(iter) \times x \times y \times z + zbest(iter) + (zbest(iter) \\ -ps(iter)) \times w, \end{aligned} \tag{14}$$

where w represents the inertia factor, and the value in this paper is 0.95. The role of w is to enhance the ability of each individual seagull to learn from the global optimal individual, so that the algorithm avoids falling into local extrema.

4.2. The Mechanism of Natural Selection

Inspired by genetic algorithms, in order to further improve the optimization performance of the algorithm, this paper introduces the mechanism of natural selection. In order to find the optimal value, keep the good seagulls and eliminate the bad ones. In each iteration, the individuals of the seagull population are sorted by the calculated fitness value, and the best half of the individuals are used to replace the worst half of the individuals, while retaining the historical best value of the original seagull individuals.

The seagull optimization algorithm is easy to fall into local extrema and the solution accuracy is not high. However, the improved seagull optimization algorithm adds the cognitive part and natural selection mechanism, so that the algorithm has better solving ability, and avoids falling into local extrema.

5. Improved Algorithm Performance Test

5.1. Experimental Setup

In order to verify the feasibility and effectiveness of the improved algorithm, this paper introduces three typical chemical engineering cases for testing. The experimentation and algorithms are implemented in MATLAB R2017a (version 9.2.0) software, it was updated and released by MathWorks of the United States in March 2017. The simulations are performed in the environment of Microsoft Windows 10 (64 bit) on Core i5 processor with 2.3 GHz and 8 GB main memory, the system was developed by Microsoft in the United States and was released in Washington in January 2015.

5.2. Algorithm Steps

First, we assume that the number of segments in the time domain is n, and use Equation (3) for unequal division. Then, we express the control variable u as a constant function in each interval, that is, $u = [u_1, u_2, \ldots, u_n]$ represents optimizing the solution of the objective function. Finally, this paper uses the Runge–Kutta method to integrate the dynamic model and the algorithm to solve each interval. Each case is run independently 20 times, and the best value of the case is recorded.

Step 1: Setting parameters and initializing the seagull population, use Equation (3) to divide the time domain unequally, and then apply the Runge–Kutta method to solve differential equations and find the best value based on fitness.

Step 2: Iterative optimization. Use Equations (4)–(12) and (14) to update the position of the seagull, and then apply the Runge–Kutta method to calculate each interval differential equation. Finally, find the best value based on fitness, and add the mechanism of natural selection before the next iteration of the algorithm.

Step 3: Determine whether the algorithm meets the end condition. If yes, jump out of the loop and output the best result; if not, return to Step 2 to continue optimization.

Step 4: End the program. Table 1 shows the setting of experimental parameters, and Algorithm 1 is the algorithm flow chart.

Table 1. Setting of experimental parameters.

Parameter	Value
f_c	0.1
μ	0.001
v	(0, 0.5)
w	0.95

Algorithm 1: The algorithm flow chart.

```
Input: Seagull population $ps(iter)$
Output: Optimal search agent $zbest(iter)$
Procedure ISOA
    Initialize positions $ps = [x_1, x_2, \ldots, x_{pop}]^T$ and parameters, mainly including $f_c$, $\mu$, $v$, $w$, $Maxiter$, where $pop$ is the number of the population.
    Use Equation (3) to divide the time domain unequally.
    Apply CalculateFitness function to calculate fitness value.
    Use FindZbest function to find zbest and zbestValue.
    for $iter$ = 1:$Maxiter$
        for $i$ = 1:$pop$
            Update position by Equations (4)–(12) and (14).
            Determine whether the position is out of bounds.
            Apply CalculateFitness function to calculate fitness value.
            Use FindZbest function to assess the fitness of all individuals.
            Store the global best individual.
            Introduce the natural selection mechanism before the next iteration to update the position.
        End for
    End for
    Output global optimal value.
End procedure
Procedure CalculateFitness
    for $i \leftarrow 1$ to $pop$ do
        for $j \leftarrow 1$ to $D$ do/* Here, $D$ represents the dimension of problem*/
            $J = Runge - Kutta(ps_i)$ /*Runge-Kutta method to calculate fitness*/
        End for
        $fit(i) = \max J$
    End for
    return $fit$
End procedure
Procedure FindZbest
    zbestValue = $fit(1)$
    for $k \leftarrow 1$ to $pop$ do
        if (zbestValue < $fit(k)$)
            zbestValue = $fit(k)$
            $zbest$ = $x_k$
        End if
    End for
    return $zbest$, zbestValue
End Procedure
```

5.3. Case 1: Batch Reactor

The batch reactor is widely used in the process industries. The batch reactor is used for solid dissolution, dissolution, product mixing, chemical reaction, crystallization, etc. A typical batch reactor consists of a tank with an agitator and an integrated heating/cooling system. Liquids and solids are usually charged through a connection port on the top cover of the reactor, vapor and gas are discharged from the top, and liquids and solids are usually discharged from the bottom. In this paper, the reaction process of the batch reactor is $A \rightarrow B \rightarrow C$. A is the reaction product, B is the intermediate product, and C is the product. After the reaction is completed, the concentration of the intermediate product B reaches the optimal value. The mathematical model of batch reactor [22] is as follows:

$$\max J = C_B\left(t_f\right),$$

$$\begin{cases} \frac{dC_A}{dt} = -k_1 {C_A}^2 \\ \frac{dC_B}{dt} = k_1 {C_A}^2 - k_2 C_B \\ 298 \leq T \leq 398, C_A(0) = 1, C_B(0) = 0, t_f = 1\ \text{h} \\ k_1 = 4 \times 10^3 \times e^{-2500/T}, k_2 = 6.2 \times 10^5 \times e^{-5000/T}, \end{cases} \tag{15}$$

where C_A is the concentration of reactants, mol/L; C_B is the concentration of intermediate products, mol/L, and t_f is the time at the end of the reaction, h; T is the control temperature, K, and J is the performance index, mol/L. In the experiment, we set the pop to 600, and the $Maxiter$ is 200.

Analysis of the experimental results of Case 1: Table 2 presents results obtained by our ISOA and other methods. The "–" in Table 3 means that the reference did not provide a value. Shi et al. [14] used the PSO-CVP optimization strategy to find the optimal value of 0.6105359. Rajesh et al. [22] used the ant colony structure optimization (ACSO) to obtain the value of 0.61045. Zhang et al. [23] used an iterative ant-colony algorithm (IACA) to obtain values of 0.61 (N = 10) and 0.6104 (N = 20). Logsdon et al. [24] used SQP to obtain the value of 0.610775 (N = 80), which is the best value reported. Renfro et al. [25] used OC to obtain the value of 0.6100. Zhou et al. [26] used a control parameterized particle swarm optimization (CP-PSO) method to obtain the value of 0.6107847, and proposed a control parameterized adaptive particle swarm optimization (CP-APSO) to obtain the value of 0.6107850. Liu et al. [27] proposed improved knowledge-based cultural algorithm (IKBCA) to obtain values of 0.6101 (N = 10), 0.610454 (N = 20) and 0.610779–0.610787 (N = 100). Sun et al. [28] applied hybrid improved genetic algorithm (HIGA) to obtain values of 0.61007 (N = 10), 0.61046 (N = 20). Peng et al. [29] proposed an improved knowledge evolution algorithm (IKEA) to get values of 0.6101 (N = 10), 0.610426 (N = 20), and 0.610781–0.610789 (N = 100), and applied genetic algorithm (GA) to obtain a value of 0.61072. Dadebo et al. [30] used IDP to obtain a value of 0.610775 (N = 80). In this paper, the optimal value of using equal division method is 0.61059223 (N = 30), and the optimal value of unequal division method is 0.610794203 (N = 30). In terms of method execution efficiency, the solution speed of ISOA combined with the equal division method is 272 s, but the solution speed of ISOA combined with the unequal division method is 118 s, and the solution efficiency of the latter is higher. Figures 2–4 are the optimal temperature control sequences. Figure 5 is the optimal trajectory of state variables.

Table 2. Comparison of batch reactor methods.

Methods	N	J/(mol/L)
PSO-CVP [14]	–	0.6105359
ACSO [22]	–	0.61045
IACA [23]	10	0.6100
IACA [23]	20	0.6104
SQP [24]	80	0.610775
OC [25]	–	0.61
CP-PSO [26]	–	0.6107847
CP-APSO [26]	–	0.6107850
IKBCA [27]	10	0.6101
IKBCA [27]	20	0.610454
IKBCA [27]	100	0.610779–0.610787
HIGA [28]	10	0.61007
HIGA [28]	20	0.61046
GA [29]	–	0.61072
IKEA [29]	10	0.6101
IKEA [29]	20	0.610426
IKEA [29]	100	0.610781–0.610789
IDP [30]	80	0.610775
equal division (ISOA)	30	0.61059223
unequal division (ISOA)	30	0.610794203

Table 3. Comparison of methods for parallel reaction problems in tubular reactors.

Methods	*N*	*J*/(mol/L)
ACSO [22]	4	0.57284
CP-PSO [26]	–	0.573543
CP-APSO [26]	–	0.573544
modified collocation-based NLP [30]	–	0.57353
nested simultaneous strategy [31]	–	0.5738
control parameterization [32]	–	0.57353
CVP [33]	–	0.56910
CVI [33]	–	0.57322
equal division (ISOA)	40	0.573073
unequal division (ISOA)	40	0.573535

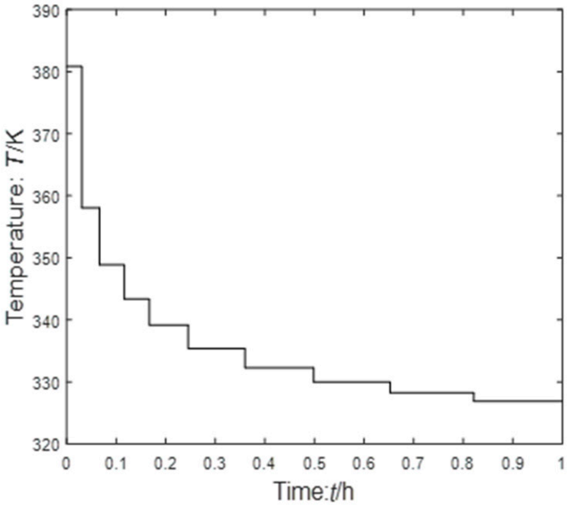

Figure 2. Optimal temperature control trajectory (N = 10).

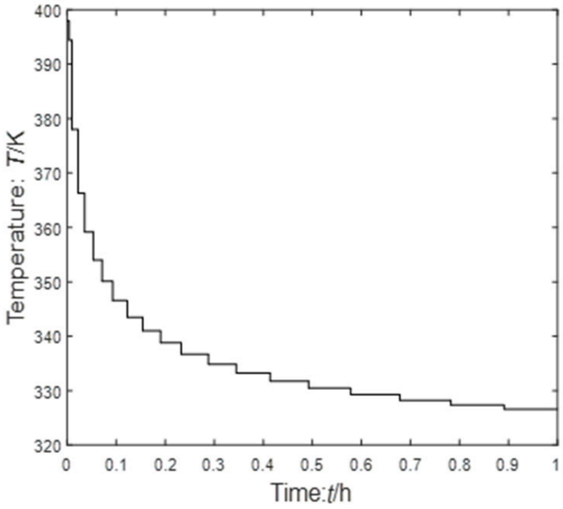

Figure 3. Optimal temperature control trajectory (N = 20).

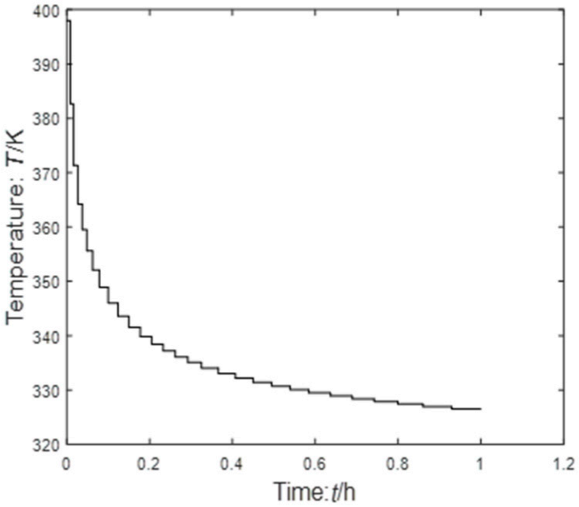

Figure 4. Optimal temperature control trajectory (*N* = 30).

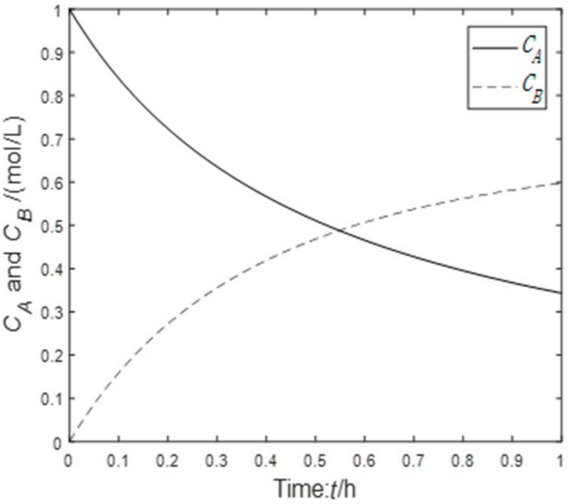

Figure 5. Optimal trajectory of state variables.

5.4. Case 2: Parallel Reaction Problem of Tubular Reactor

Parallel reaction means that reactants can carry out two or more different reactions in parallel to obtain different products. For example, ethanol can undergo two reactions of dehydration and dehydrogenation in parallel. In our paper, we introduce a dynamic optimization problem with saturation characteristics of control variables, which is cited by many scholars. The process of this problem is $A \rightarrow B$ and $A \rightarrow C$. By determining the optimal value of the control variable, the concentration of by-product B is maximized after the reaction is completed. The mathematical model of this reaction is as follows [31]:

$$\max J\left(t_f\right) = x_2\left(t_f\right),$$

$$\begin{cases} \frac{dx_1}{dt} = -\left[u(t) + 0.5u^2\right]x_1(t) \\ \frac{dx_2}{dt} = u(t)x_1(t) \\ 0 \le u(t) \le 5 \\ x_1(0) = 1, x_2(0) = 0, t_f = 1\text{ s}, \end{cases} \tag{16}$$

where $x_1(t)$ is the concentration of reactants A, mol/L; $x_2(t)$ is the concentration of by-product B, mol/L, and J is the performance index, mol/L; $u(t)$ represents the control variable, 10^6, and t_f is terminal time, s. In the experiment, we set the *pop* to 800, and the *Maxiter* is 200.

Analysis of the experimental results of Case 2: Table 3 presents results obtained by our ISOA and other methods. The "–" in Table 3 means that the reference did not provide a value. Rajesh et al. [22] used the ant colony structure optimization (ACSO) to obtain the value of 0.57284 (N = 4). Zhou et al. [26] used a control parameterized particle swarm optimization (CP-PSO) method to obtain the value of 0.573543, and proposed a control parameterized adaptive particle swarm optimization (CP-APSO) to obtain the value of 0.573544. Dadebo [30] applied IDP to obtain the value of 0.57353. Tanartkit and Biegler [31] used a nested simultaneous strategy to obtain the value of 0.5738, which is the best reported value. Vassiliadis [32] solved the model with a value of 0.57353. Biegler [33] proposed a combination of continuous quadratic programming and orthogonal collocation method to obtain values of 0.56910 (CVP) and 0.57322 control vector iteration (CVI). In this paper, the optimal value of using equal division method is 0.573073 (N = 40), and the optimal value of unequal division method is 0.573435 (N = 40). In terms of method execution efficiency, the solution speed of ISOA combined with the equal division method is 326 s, but the solution speed of ISOA combined with the unequal division method is 302 s, and the optimization efficiency of the latter is about 10% higher than that of the equal division method. Figures 6–8 are the optimal control trajectories, and Figure 9 is the optimal state variable trajectory.

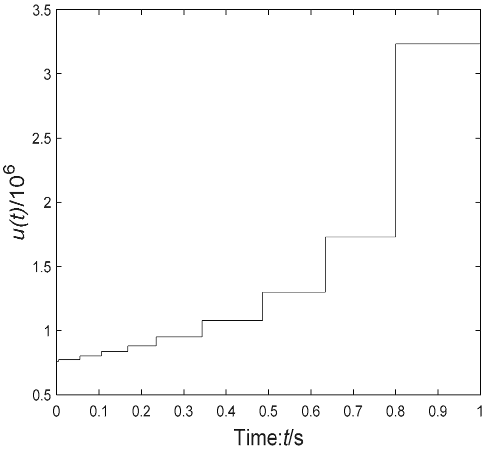

Figure 6. Optimal temperature control trajectory (N = 10).

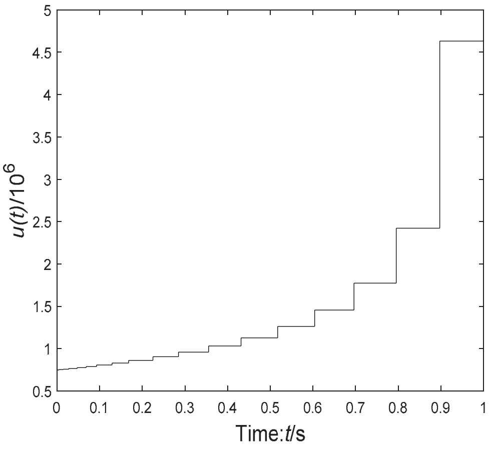

Figure 7. Optimal temperature control trajectory (N = 20).

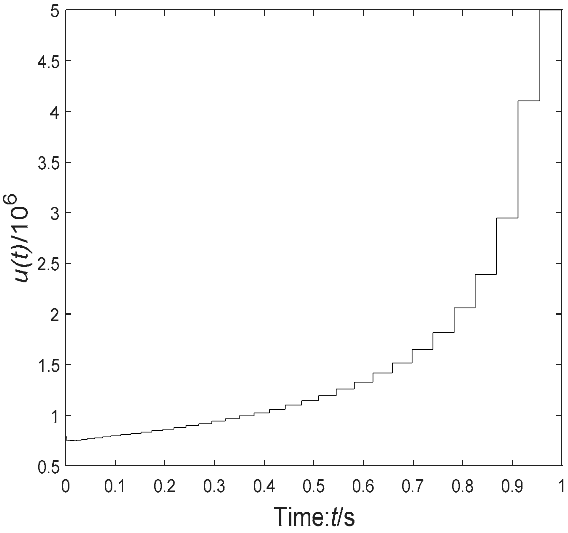

Figure 8. Optimal temperature control trajectory (N = 40).

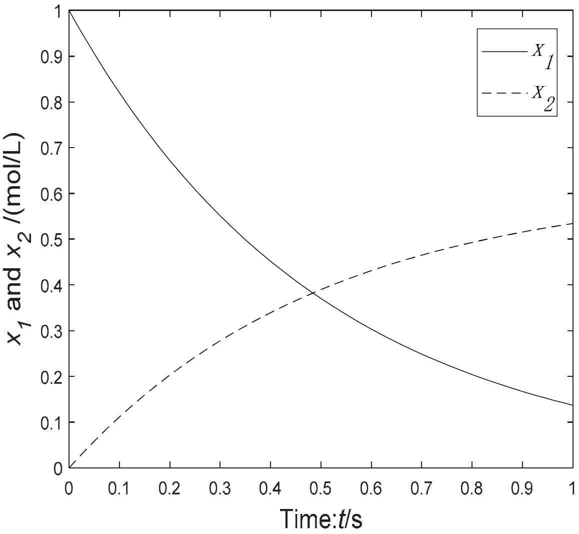

Figure 9. Optimal trajectory of state variables.

5.5. Case 3: Tubular Reactor

The tubular reactor was proposed by some authors [34,35]. The reaction process is as follows: in a certain length of pipeline, the catalysts A and B are mixed, and the target product C is produced after the reaction is completed. The goal of the problem is to distribute the catalyst in the designated pipeline so that the concentration of the target product C reaches the optimal value. The mathematical model of the tubular reactor problem is as follows:

$$\max J\left(z_f\right) = 1 - x_A\left(z_f\right) - x_B\left(z_f\right),$$

$$\begin{cases} \frac{dx_A}{dz} = -u(z)[10 \times x_B(z) - x_A(z)] \\ \frac{dx_B}{dz} = u(z)[10 \times x_B(z) - x_A(z)] \times x_B(z) \\ 0 \le u(z) \le 1 \\ C_A(0) = 1, C_B(0) = 0, z_f = 12 \text{ m}, \end{cases} \tag{17}$$

where $x_A(z)$ is the concentration of catalyst A, mol/L; $x_B(z)$ is the concentration of catalyst B, mol/L; z_f is the pipe length of the tubular reactor, m; $u(z)$ is the mixing fraction of catalyst A, %; J is the performance index of the model, mol/L. In the experiment, we set the pop to 800, and the $Maxiter$ is 200.

Analysis of the experimental results of Case 3: Table 4 presents results obtained by our ISOA and other methods. Rajesh et al. [22] used the ant colony structure optimization (ACSO) to obtain values of 0.47615 (N = 5) and 0.47527 (N = 20). Peng et al. [29] proposed an improved knowledge evolution algorithm (IKEA) to obtain values of 0.4757 (N = 20) and 0.47761–0.47768 (N = 100). Dadebo [30] applied IDP to obtain values of 0.47527 (N = 20) and 0.47695 (N = 40). Chen et al. [36] used nonuniform discretization-based control vector parameterization to obtain the value of 0.47771 (N = 15). The reference of [37] used trigonometric differential evolution (TDE) to obtain values of 0.47527 (N = 20) and 0.47683 (N = 40). The reference of [38] used bees algorithm (BA) to obtain the value of 0.4744 (N = 20). In this paper, the optimal value of using the equal division method is 0.47721 (N = 40), and the optimal value of unequal division method is 0.47770 (N = 40). The value of the latter is better than the former. All in all, the value of the unequal division method is better than the value of other references [22,29,30,37,38]. In terms of method execution efficiency, the solution speed of ISOA combined with the equal division method is 575 s, but the solution speed of ISOA combined with the unequal division method is 285 s. Figures 10–12 are the optimal control trajectories, and Figure 13 is the optimal state variable trajectory.

Table 4. Comparison of methods of tubular reactor.

Methods	***N***	***J*/(mol/L)**
ACSO [22]	5	0.47615
ACSO [22]	20	0.47527
IKEA [29]	20	0.4757
IKEA [29]	100	0.47761–0.47768
IDP [30]	20	0.47527
IDP [30]	40	0.47695
ndCVP-HGPSO [36]	15	0.47771
TDE [37]	20	0.47527
TDE [37]	40	0.47683
BA [38]	20	0.47744
equal division (ISOA)	40	0.47721
unequal division (ISOA)	40	0.47770

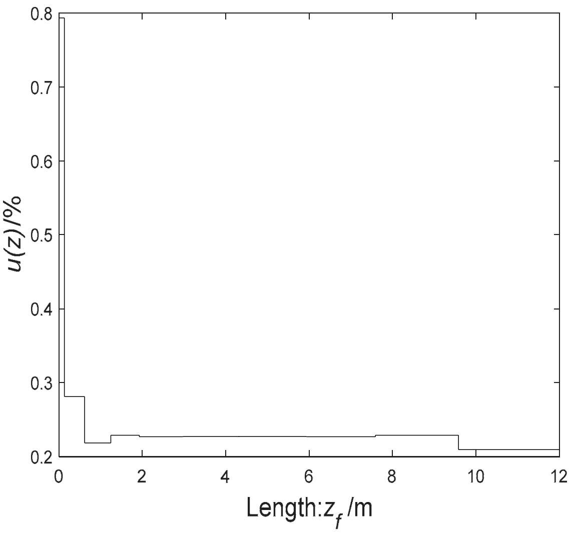

Figure 10. Optimal control trajectory (N = 10).

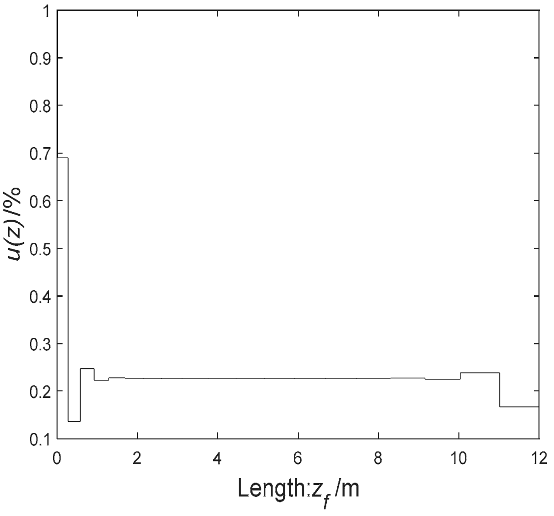

Figure 11. Optimal control trajectory (N = 20).

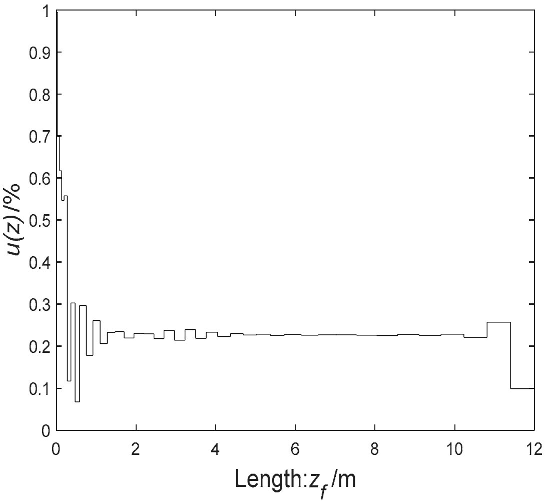

Figure 12. Optimal control trajectory (N = 40).

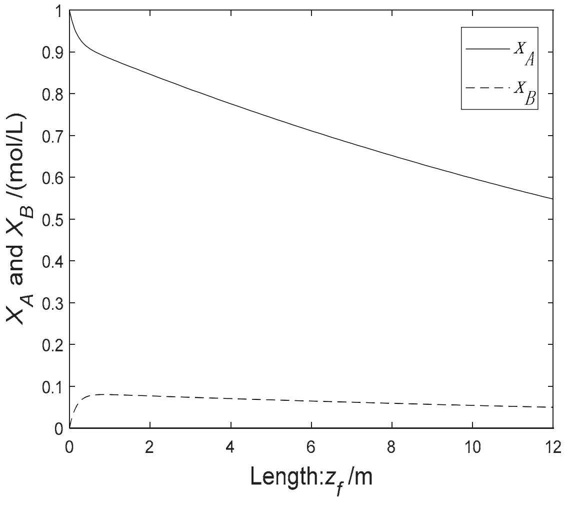

Figure 13. Optimal state variable trajectory.

6. Conclusions

In order to achieve the sustainable development of society and reduce energy consumption, this paper proposes an improved seagull optimization algorithm (ISOA) combined with an unequal division method to solve dynamic optimization problems through the analysis of chemical processes. First of all, we analyzed that the seagull optimization algorithm (SOA) may easily fall into local extrema. In order to improve this shortcoming, this paper introduced the cognitive part and the mechanism of natural selection to improve SOA, thereby effectively avoiding the algorithm from falling into the local optimum, and enhancing the optimization performance of the algorithm. Secondly, chemical dynamic optimization problems usually need to discretize the time domain, use a piecewise function to approach the optimal solution, and convert the problem into NLP for processing. Therefore, this paper used the unequal division method to divide the time domain so that the calculation becomes simple.

Finally, in order to verify the feasibility and effectiveness of improving SOA, the ISOA is combined with an unequal division method to solve dynamic optimization problems. The feasibility of the method is verified by comparing with other algorithms. In this paper, many reference methods use basically the uniform discretization of control variables, and the time domain needs to be divided into multiple segments. At the same time, we use the unequal division method of the control variable, and the time domain is divided into fewer segments, which makes the calculation more simple. For example, the time efficiency of ISOA combined with the unequal division method is higher than that of ISOA combined with the equal division method.

Author Contributions: Conceptualization, L.X. and Y.M.; methodology, L.X.; software, J.L.; validation, Y.M. and Y.L.; writing—original draft preparation, L.X.; writing—review and editing, L.X and Y.M.; supervision, Y.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China, grant numbers 21466008, 21566007, 21968008, and the Guangxi Natural Science Foundation of China, grant number 2019GXNSFAA185017.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: This research was supported by the National Natural Science Foundation of China, grant numbers 21466008, 21566007, 21968008, and the Guangxi Natural Science Foundation of China, grant number 2019GXNSFAA185017.

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

References

1. Peng, H.J.; Gao, Q.; Wu, Z.G.; Zhong, W.X. A mixed variable variational method for optimal control problems with applications in aerospace control. *Zidonghua Xuebao/Acta Autom. Sin.* **2011**, *37*, 1248–1255.
2. Sun, Y.; Zhang, M.R.; Liang, X.L. Improved Gauss Pseudospectral Method for Solving Nonlinear Optimal Control Problem with Complex Constraints. *Acta Autom. Sin.* **2013**, *39*, 672–678. [CrossRef]
3. Sundarlingam, R. Two-step method for dynamic optimization of inequality state constrained systems using iterative dynamic programming. *Ind. Eng. Chem. Res.* **2015**, *54*, 7658–7667. [CrossRef]
4. Szymkat, M.; Korytowski, A.; Turnau, A. Variable Control Parameterization for Time-Optimal Problems. *IFAC Proc. Vol.* **2000**, *33*, 191–196. [CrossRef]
5. Chen, X.; Du, W.; Qian, F. Solving chemical dynamic optimization problems with ranking-based differential evolution algorithms. *Chin. J. Chem. Eng.* **2016**, *24*, 1600–1608. [CrossRef]
6. Fan, Q.; Wang, X.; Yan, X. Harmony search algorithm with differential evolution based control parameter co-evolution and its application in chemical process dynamic optimization. *J. Cent. South Univ.* **2015**, *22*, 2227–2237. [CrossRef]
7. Luus, R. Optimization of fed-batch fermenters by iterative dynamic programming. *Biotechnol. Bioeng.* **1993**, *41*, 599–602. [CrossRef] [PubMed]

8. Pollard, J.P.; Sargent, R.W.H. Off line computation of optimum controls for a plate distillation column. *Automatica* **1970**, *6*, 59–76. [CrossRef]
9. Sargent, R.W.H.; Sullivan, G.R. The development of an efficient optimal control package. *Nonlinear Stoch.* **1979**, *13*, 158–168.
10. Xu, W.; Jiang, A.; Wang, H. A grid reconstruction strategy based on pseudo Wigner-Ville analysis for dynamic optimization problem. *CIESC J.* **2019**, *70* (Suppl. 1), 158–167.
11. Li, G.; Liu, X. A variable time nodes control vector parameterization approach for solving optimal control problems. *CIESC J.* **2015**, *66*, 640–646.
12. Li, G.; Liu, P.; Liu, X. A Control Parameterization Approach with Variable Time Nodes for Optimal Control Problems. *Asian J. Control* **2016**, *18*, 976–984. [CrossRef]
13. Binder, T.; Cruse, A.; Cruz, V. Dynamic optimization using a wavelet based adaptive control. *Comput. Chem. Eng.* **2000**, *24*, 1201–1207. [CrossRef]
14. Shi, B.; Yin, Y.; Liu, F. Optimal control strategies combined with PSO and control vector parameterization for batchwise chemical process. *CIESC J.* **2019**, *70*, 979–986.
15. Xu, C.; Mei, C.; Xu, B. Biogeography-based learning particle swarm optimization method for solving dynamic optimization problems in chemical processes. *CIESC J.* **2017**, *68*, 3161–3167.
16. Sarkar, D.; Modak, J.M. Optimization of fed-batch bioreactors using genetic algorithm: Multiple control variables. *Comput. Chem. Eng.* **2004**, *28*, 789–798. [CrossRef]
17. Anand, P.; Rao, M.B.; Venkateswarlu, C. Dynamic optimization of a copolymerization reactor using tabu search. *ISA Trans.* **2015**, *55*, 13–26. [CrossRef] [PubMed]
18. Schluter, M.; Egea, J.A.; Antelo, L.T. An Extended Ant Colony Optimization Algorithm for Integrated Process and Control System Design. *Ind. Eng. Chem. Res.* **2009**, *48*, 6723–6738. [CrossRef]
19. Egea, J.A.; Balsa-canto, E.; Garcia, M.S. Dynamic optimization of nonlinear processes with an enhanced scatter search method. *Ind. Eng. Chem. Res.* **2009**, *48*, 4388–4401. [CrossRef]
20. Nikumbh, S.; Ghosh, S.; Jayaraman, V.K. Biogeography-Based Optimization for Dynamic Optimization of Chemical Reactors. In *Applications of Metaheuristics in Process Engineering*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 201–216.
21. Dhiman, G.; Kumar, V. Seagull optimization algorithm:Theory and its applications for large-scale industrial engineering problems. *Knowl. Based Syst.* **2019**, *165*, 169–196. [CrossRef]
22. Rajesh, J.; Gupta, K.; Kusumakar, H.S. Dynamic optimization of chemical processes using ant colony framework. *Comput. Chem.* **2001**, *25*, 583–595. [CrossRef]
23. Zhang, B.; Chen, D.; Zhao, W. Iterative ant-colony algorithm and its application to dynamic optimization of chemical process. *Comput. Chem. Eng.* **2005**, *29*, 2078–2086. [CrossRef]
24. Logsdon, J.S.; Biegler, L.T. A relaxed reduced space SQP strategyfor dynamic optimization problems. *Comput. Chem. Eng.* **1993**, *17*, 367–372. [CrossRef]
25. Renfro, J.G.; Morshedi, A.M.; Asbjornsen, O.A. Simultaneous optimization and solution of systems described by differential/algebraic equations. *Comput. Chem. Eng.* **1987**, *11*, 503–517. [CrossRef]
26. Liu, X.; Zhou, Y. Control Parameterization-Based Adaptive Particle Swarm Approach for Solving Chemical Dynamic Optimization Problems. *Chem. Eng. Technol.* **2014**, *37*, 692–702.
27. Liu, Z.; Du, W.L.; Qi, R.B. Dynamic optimization in chemical processes using improved knowledge-based cultural algorithm. *CIESC J.* **2010**, *61*, 2889–2895.
28. Sun, F.; Du, W.L.; Qi, R.B. A Hybrid Improved Genetic Algorithm and Its Application in Dynamic Optimization Problems of Chemical Processes. *Chin. J. Chem. Eng.* **2013**, *21*, 144–154. [CrossRef]
29. Peng, X.; Qi, R.B.; Du, W.L.; Qian, F. An improved knowledge evolution algorithm and its application in chemical dynamic optimization. *CIESC J.* **2012**, *63*, 841–850.
30. Dadebo, S.A.; Mcauley, K.B. Dynamic optimization of constrained chemical engineering problems using dynamic programming. *Comput. Chem. Eng.* **1995**, *19*, 513–525. [CrossRef]
31. Tanartkit, P.; Biegler, L.T. A nested, simultaneous approach for dynamic optimization problems—II: The outer problem. *Comput. Chem. Eng.* **1997**, *21*, 735–741. [CrossRef]
32. Vassiliadis, V. Computational Solution of Dynamic Optimization Problems with General Differential-Algebraic Constraints. *J. Guid. Control Dyn.* **1993**, *15*, 457–460.
33. Biegler, L.T. "Solution of dynamic optimization problems by successive quadratic programming and orthogonal collocation". *Comput. Chem. Eng.* **1984**, *8*, 243–248.
34. Jackson, R. Optimization of chemical reactors with respect to flow configuration. *J. Optim. Theory Appl.* **1968**, *2*, 240–259. [CrossRef]
35. Logsdon, J.S.; Biegler, L.T. On the accurate solution of differential-algebraic optimization problems. *Ind. Eng. Chem. Res.* **1989**, *28*, 1628–1639. [CrossRef]

36. Chen, X.; Du, W. Dynamic Optimization of Industrial Processes with Nonuniform Discretization-Based Control Vector Parameterization. *IEEE Trans. Autom. Sci. Eng.* **2017**, *11*, 1289–1299. [CrossRef]
37. Angira, R.; Santosh, A. Optimization of dynamic systems: A trigonometric differential evolution approach. *Comput. Chem. Eng.* **2007**, *31*, 1055–1063. [CrossRef]
38. Pham, D.T.; Pham, Q.T.; Ghanbarzadeh, A. Dynamic Optimisation of Chemical Engineering Processes Using the Bees Algorithm. In Proceedings of the 17th World Congress of the International Federation of Automatic Control (IFAC 2008), Seoul, Korea, 6–11 July 2008.