symmetry

Article

Study on an Adaptive Co-Evolutionary ACO Algorithm for Complex Optimization Problems

Huimin Zhao [1,5,6], Weitong Gao [1], Wu Deng [1,2,3,4,5,6,*] and Meng Sun [1]

1 Software Institute, Dalian Jiaotong University, Dalian 116028, China; hm_zhao1977@126.com (H.Z.); gao1174750291@163.com (W.G.); fukaifang@yeah.net (M.S.)
2 Key Laboratory of Guangxi High Schools Complex System and Computational Intelligence, Nanning 530006, China
3 Guangxi Key Lab of Multi-Source Information Mining & Security, Guangxi Normal University, Guilin 541004, China
4 Guangxi Key Laboratory of Hybrid Computation and IC Design Analysis, Guangxi University for Nationalities, Nanning 530006, China
5 Traction Power State Key Laboratory, Southwest Jiaotong University, Chengdu 610031, China
6 Liaoning Key Laboratory of Welding and Reliability of Rail Transportation Equipment, Dalian Jiaotong University, Dalian 116028, China
* Correspondence: dw7689@djtu.edu.cn; Tel.: +86-411-8410-5386

Received: 24 March 2018; Accepted: 9 April 2018; Published: 11 April 2018

Abstract: The ant colony optimization (ACO) algorithm has the characteristics of positive feedback, essential parallelism, and global convergence, but it has the shortcomings of premature convergence and slow convergence speed. The co-evolutionary algorithm (CEA) emphasizes the existing interaction among different sub-populations, but it is overly formal, and does not form a very strict and unified definition. Therefore, a new adaptive co-evolutionary ant colony optimization (SCEACO) algorithm based on the complementary advantages and hybrid mechanism is proposed in this paper. Firstly, the pheromone update formula is improved and the pheromone range of the ACO algorithm is limited in order to achieve the adaptive update of the pheromone. The elitist strategy and co-evolutionary idea are used for reference, the symbiotic mechanism and hybrid mechanism are introduced to better utilize the advantages of the CEA and ACO. Then the multi-objective optimization problem is divided into several sub-problems, each sub-problem corresponds to one population. Each ant colony is divided into multiple sub-populations in a common search space, and each sub-population performs the search activity and pheromone updating strategy. The elitist strategy is used to retain the elitist individuals within the population and the min-max ant strategy is used to set pheromone concentration for each path. Next, the selection, crossover, and mutation operations of individuals are introduced to adaptively adjust the parameters and implement the information sharing of the population and the co-evolution. Finally, the gate assignment problem of a hub airport is selected to verify the optimization performance of the SCEACO algorithm. The experiment results show that the SCEACO algorithm can effectively solve the gate assignment problem of a hub airport and obtain the effective assignment result. The SCEACO algorithm improves the convergence speed, and enhances the local search ability and global search capability.

Keywords: co-evolution; ant colony optimization (ACO); multi-strategies; hybrid mechanism; multi-objective optimization model; gate assignment

1. Introduction

A large number of problems in the fields of industry, agriculture, national defense, information, transportation, economy, management, and so on, can be transformed into optimization problems [1].

However, optimization problems are some of the more popular and difficult problems in the world. At present, there are no mature theories and methods to effectively solve these optimization problems. As an important branch of scientific research, the optimization methods have made a great impact on the development of many disciplines [2]. They have received extensive attention and have been widely promoted and applied in many fields. The optimization method has become an indispensable tool in different fields. With the rapid development of new computer and information technologies, the applications of the optimization theories and methods have become more and more widely applicable in a large number of fields. Many of the latest developments depend on the improvement and innovation of numerical techniques for calculating optimization problems, and these methods have become popular research topics at home and abroad [3–6]. Therefore, the new optimization theory and method are studied, which not only have important theoretical significance, but also have wide application value.

As the most effective methods in solving optimization problems, the ant colony optimization (ACO) algorithm and co-evolution algorithm have received extensive attention and research. The ACO algorithm is a heuristic evolution algorithm based on population, which is inspired by the research results of the collective behavior of real ants in nature. Dorigo et al. [7] proposed the ACO algorithm based on making full use of the similarity between the process of an ant colony searching food and the famous traveling salesman problem (TSP). Since then, many experts and scholars have devoted themselves to the research of the ACO algorithm. However, with the increasing complexity of various large-scale optimization problems, the ACO algorithm has some inherent shortcomings. In recent years, many experts have been trying to improve the ACO algorithm, and some improved ACO algorithms are proposed to improve the performance of the ACO algorithm [8,9]. However, the inherent shortcomings of the ACO algorithm have not been fundamentally solved. The co-evolutionary algorithm (CEA) is a new algorithm based on co-evolution theory [10]. It emphasizes the coordination between the population and the environment, the population, and the population in the process of continuous evolution. The CEA can reasonably divide the optimization problem space, and can effectively jump out of the local optimum value and find the best optimal solution for a large-scale optimization problem. It emphasizes the existing interaction among different sub-populations, affecting each other and coevolving together [11,12]. Numerical calculation and optimization problems are the original models for practical engineering problems, so the study of numerical computation and optimization have certain representativeness. However, the CEA is an optimization method for engineering applications, so it is overly formal, and does not form a very strict and unified definition, which will hinder further research and development of the CEA. Therefore, it is necessary to study deeply the combination of the CEA with other algorithms in order to improve the computational efficiency and promote the research and development of the CEA.

The ACO algorithm has the characteristics of positive feedback, essential parallelism, and global convergence, but it has the shortcomings of premature convergence and slow convergence speed. The CEA emphasizes the existing interaction among different sub-populations, but it is overly formal, and does not form a very strict and unified definition. In order to realize the complementary advantages, the elitist strategy, the min-max ant strategy, co-evolutionary idea, the symbiotic mechanism, and the hybrid mechanism are introduced into the CEA and ACO algorithms in order to propose a new adaptive co-evolutionary ant colony optimization (SCEACO) algorithm. Then, the theories of the SCEACO algorithm are studied and analyzed in detail. The practical application case is selected to verify the optimization performance of the SCEACO algorithm. Therefore, this study has important theoretical significance and practical application value.

The remainder of the paper is organized as follows: The related works are comprehensively analyzed in Section 2. The basic methods are introduced in Section 3. In Section 4, an adaptive co-evolutionary ACO (SCEACO) algorithm based on the elitist strategy, co-evolutionary idea, the symbiotic mechanism, and the hybrid mechanism is proposed in detail. In Section 5, the SCEACO algorithm is applied in solving the gate assignment problem. In Section 6, data analysis from

Guangzhou Baiyun airport of China is introduced in detail. Finally, the conclusions are offered and future research direction is discussed in Section 7.

2. Related Works

In recent years, many experts and scholars have devoted themselves to the research of the ACO algorithm and the CEA. Many improved ACO algorithms and CEA are proposed to improve the optimization performance. Yang and Cho [13] proposed a strategic coalition to obtain superior adaptive agents and simulate its emergence in a co-evolutionary learning environment. He and Wang [14] proposed a co-evolutionary particle swarm optimization (PSO) approach (CPSO) to solve the constrained optimization problems. Huang et al. [15] proposed a differential evolution approach based on a co-evolution mechanism. Kou et al. [16] proposed a new co-evolutionary PSO algorithm to solve global nonlinear optimization problems. Gu et al. [17] proposed a novel competitive co-evolutionary quantum genetic algorithm (GA). Coelho and Bernert [18] proposed a modified continuous approach based on combining ACO with differential evolution. Xing et al. [19] proposed a multi-population interactive co-evolutionary algorithm based on ACO and GA with different configurations. Li and Wang [20] proposed a quantum ACO algorithm based on a Bloch sphere search. Liao et al. [21] proposed three new hybrid ACO algorithms that are extended from the developed original ACO structure (ACOR). Chandra et al. [22] proposed a new cooperative co-evolution framework based on incorporating a crossover-based local search. Sun et al. [23] proposed a hybrid co-evolutionary cultural algorithm based on PSO and a shared global belief space. Gao and Wang [24] proposed a co-evolutionary algorithm based on PSO and ACO. Fernández-Vargas et al. [25] proposed a new continuous ACO method with feasible region selection. Rizk-Allah et al. [26] proposed a novel hybrid ACO-FA algorithm integrating the ACO and firefly algorithm. Juang et al. [27] proposed a cooperative continuous ACO to address the accuracy-oriented fuzzy systems design problems. Zhao et al. [28] proposed an extension to a previous piece of work on multi-objective cooperative co-evolution algorithms. Ding et al. [29] proposed a more efficient attribute self-adaptive co-evolutionary reduction algorithm. Bu et al. [30] proposed a weighted algorithm to calculate each arc's increment based on its selected probability and provided greater exploration. Jiang et al. [31] presented a co-evolutionary multi-ACO algorithm. Wan et al. [32] proposed a modified binary-coded ACO algorithm. Pan [33] proposed a novel cooperative co-evolutionary ABC algorithm. Lei et al. [34] proposed an enhanced multi-objective co-evolutionary algorithm. Yu et al. [35] proposed a multi-population co-evolutionary genetic programming approach to identify the optimal design. Goran and Tihana [36] proposed a co-evolutionary multi-population genetic program. Nilakantan et al. [37] proposed a multi-objective co-operative co-evolutionary algorithm. Hiew et al. [38] proposed a competitive co-evolution. Yang et al. [39] proposed an adaptive multimodal continuous ACO algorithm with current niching methods. Xu et al. [40] proposed a new heuristic dual population ACO. Ding et al. [41] proposed a co-evolutionary quantum PSO with self-adaptive memeplexes.

Other algorithms and methods are also proposed to solve the complex problems. Xue et al. [42] proposed a self-adaptive artificial bee colony algorithm based on the global best candidate. Yu et al. [43] employed ant colony algorithm to train the nonparametric mode for obtaining the optimal weights based on the force-displacement/velocity data. Deng et al. [44] proposed an improved adaptive PSO algorithm to solve the gate assignment problem.

In recent decades, some improved CEA and ACO algorithms have been proposed to better solve the complex optimization problems. However, the improved CEA still has a slow convergence speed and high computation complexity, and the improved ACO algorithms have long calculation times and are easy to fall into stagnation, etc. Therefore, it is necessary to deeply study the combination of the CEA with ACO algorithms to construct a new algorithm for improving the computation efficiency.

3. Basic Method

3.1. The CEA

Co-evolution is a trait of a species that evolves as a reaction to the character of another species [45]. The character of the latter species is evolved as a response to the character of the former species. The CEA is a new evolutionary algorithm based on co-evolution theory in recent years. It admits the biological diversity, emphasizes a certain dependence between organisms, and between organisms and the environment in the process of evolution. The CEA considers the coordination between populations and the population and environment in the evolution process. The fitness of individuals is determined by the performance of individuals under interaction with other individuals. Although the study of the CEA started recently, more scholars have studied it due to the superiority of the CEA. At present, the CEA has become a hot topic in current evolutionary algorithms.

For solving large-scale optimization problems, the CEA can effectively jump out the local optimum value by using the reasonable population division so as to search for a better optimization solution. As one class of highly-abstract algorithm models, the CEA can flexibly establish the solving model. The basic framework of the CEA is shown in Figure 1.

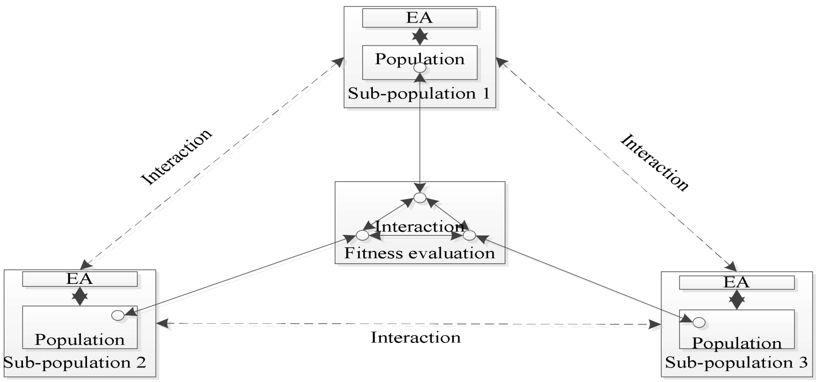

Figure 1. The basic frame of CEA.

The framework of the CEA includes three sub-populations in Figure 1. The individuals in each sub-population implement the corresponding evolutionary operation by using different evolutionary algorithms. There exists the interrelation and interaction among sub-populations, and each sub-population is independently evolved by the evolutionary algorithm in order to search for the optimal solutions.

3.2. The ACO Algorithm

The ACO is a meta-heuristic optimization algorithm [7]. In every iteration, many ants use heuristic information and the collected experiences to establish the complete solutions. The pheromone trail is deposited on the constituent elements of the solution, it is used to represent the collected experiences. Pheromone can be deposited on the components according to the solution of solving problem. The steps of the pheromone update rule is described:

(1) The Transition Rule

In the route, the k^{th} ant starts from city r, the next city s is selected among the unvisited cities memorized in J_r^k according to the following formula:

$$s = \underset{u \in J_r^k}{\arg\max}[\tau_i(r,u)^{\alpha} \cdot \eta(r,u)^{\beta}] \ if \ q \leq q_0(Exploitation) \tag{1}$$

To visit the next city s with the transition probability $p_k(r,s)$:

$$p_k(r,s) = \begin{cases} \frac{\tau(r,s)^\alpha \cdot \eta(r,s)^\beta}{\sum\limits_{u \in J_r^k} \tau(r,u)^\alpha \cdot \eta(r,u)^\beta} & if\ s \in J_r^k \\ 0 & otherwise \end{cases} \tag{2}$$

In Equation (2), $p_k(r,s)$ is the transition probability, $\tau(r,u)$ is the pheromone concentration between city r and city u, $\eta(r,u)$ is the path length from city r to city u, J_r^k is the set of unvisited cities of the k^{th} ant, the parameter α and β are the control parameters, and q is a uniform probability [0, 1].

(2) The Pheromone Update Rule

The pheromone trails need be updated to improve the solution quantity. Trail updating includes local and global updating. The local updating is given:

$$\tau(r,u) = (1-\rho)\tau(r,s) + \sum_{k=1}^{m} \Delta\tau_k(r,s) \tag{3}$$

In Equation (3), $\rho(0 < \rho < 1)$ is the pheromone trial evaporating rate. $\Delta\tau_k(r,s)$ is the amount of pheromone trail added to the edge (r,s) by ant k between time t and $t + \Delta t$ in the tour. It is given:

$$\Delta\tau_k(r,s) = \begin{cases} \frac{Q}{L_k} & (\mathrm{r},s) \in \pi_k \\ 0 & otherwise \end{cases} \tag{4}$$

where Q is a constant parameter, and L_k is the distance of the sequence π_k toured by ant in Δt

3.3. Adaptive ACO Algorithm

An adaptive ACO (SACO) algorithm is proposed to solve the stagnation of the ACO algorithm. In the proposed SACO algorithm, the pheromone updating formula and the updating restraint range of pheromones have been improved.

(1) The Improved Pheromone Updating Formula

In order to reduce the optimizing influence of the ACO algorithm by the poor path, the difference of pheromone quantity between the best path and other paths is increased in a short time in order to guide the ACO algorithm to converge to the optimal path. At the same time, the convergence speed of the ACO algorithm is accelerated. Therefore, the pheromone updating formula is changed:

$$\Delta\tau_{ij}^k(t) = \frac{Q}{2^{L_k(t) - BestSolution}} \tag{5}$$

The ants in the ACO algorithm have significant differences in the searching path length for the different problems, so it is difficult to control the $L_k(t) - BestSolution$ in a certain range. Too large or too small values of the $L_k(t) - BestSolution$ will result in the excessive or excessive increment of pheromone, which will affect the optimizing ability of ants and fall into a local optimum value. Therefore, the pheromone updating Equation (5) is changed:

$$\Delta\tau_{ij}^k(t) = \frac{Q}{2^{coef}} \tag{6}$$

where:

$$coef = \begin{cases} -1 & L_k(t) - BestSolution < 0 \\ 1 & L_k(t) - BestSolution \geq 0 \end{cases} \tag{7}$$

The improved pheromone updating formula can quickly increase the amount of pheromone on the best path, while the increased amount of pheromone on the bad path is not obvious. There will be a certain difference in the pheromone amount on the best path and the poor path after multiple iterations, which is beneficial to eliminate the interference of the poor path, which greatly accelerates the optimization speed, and quickly converges to the optimal value.

(2) The updating restraint range of the pheromone:

$$coef = \begin{cases} -1 & L_k(t) - BestSolution < 0 \\ 1 & L_k(t) - BestSolution \geq 0 \end{cases} \tag{8}$$

When the pheromone is updated, the pheromone concentration is limited to the interval $[\tau_{\min}, \tau_{\max}]$. If $\tau_{ij} < \tau_{\min}$, the $\tau_{ij} = \tau_{\min}$ is set. If $\tau_{ij} > \tau_{\max}$, the $\tau_{ij} = \tau_{\min}$ is set [46–48]. The initial pheromone is set as the upper bound of the value range, that is, $\tau_{ij}(0) = \tau_{\max}$.

4. Adaptive Co-Evolutionary ACO (SCEACO) Algorithm

4.1. The Idea of the SCEACO Algorithm

In recent years, the application of the ACO algorithm has extended from a combinatorial optimization problem to a function optimization problem, from the unconstrained problem to the constrained problem, from a single objective optimization problem to a multi-objective optimization problem. The ACO algorithm has the characteristics of positive feedback, essential parallelism, and global convergence in solving optimization problems, but it has shortcomings, such as undetermined parameters, premature stagnation, and so on. The CEA uses biological co-evolution theory to construct the competition or cooperation relation by two or more populations for improving the performance by the interaction of multiple populations. It emphasizes the existing interaction among different sub-populations, and affects each other and coevolves together. However, it is overly formal, and does not form a very strict and unified definition. Therefore, the idea of co-evolution, the elitist strategy, the min-max ant strategy, the symbiosis mechanism, and the hybrid mechanism are introduced into the CEA and ACO algorithms in order to form the complementation of advantages and propose an adaptive co-evolutionary ant colony optimization (SCEACO) algorithm in this paper. In the proposed SCEACO algorithm, the pheromone update formula is improved and the pheromone range of the ACO algorithm is limited to achieve the adaptive update of pheromones. The symbiotic mechanism and hybrid mechanism are introduced to better utilize the advantages of the CEA and ACO. The multi-population strategy is used to divide the population into multiple sub-populations in a common search space for improving the search activity and pheromone updating strategy. The elitist strategy is used to retain the elitist individuals within the population. The min-max ant strategy is used to set the pheromone concentration for each path. The selection, crossover, and mutation operations of individuals are used to adaptively adjust the parameters and implement the information sharing of the population and the co-evolution.

4.2. The Model of the SCEACO Algorithm

According to the co-evolution idea, hybrid mechanism, multi-population strategy, elitist strategy, and min-max ant strategy, the flow of the proposed SCEACO algorithm is shown in Figure 2.

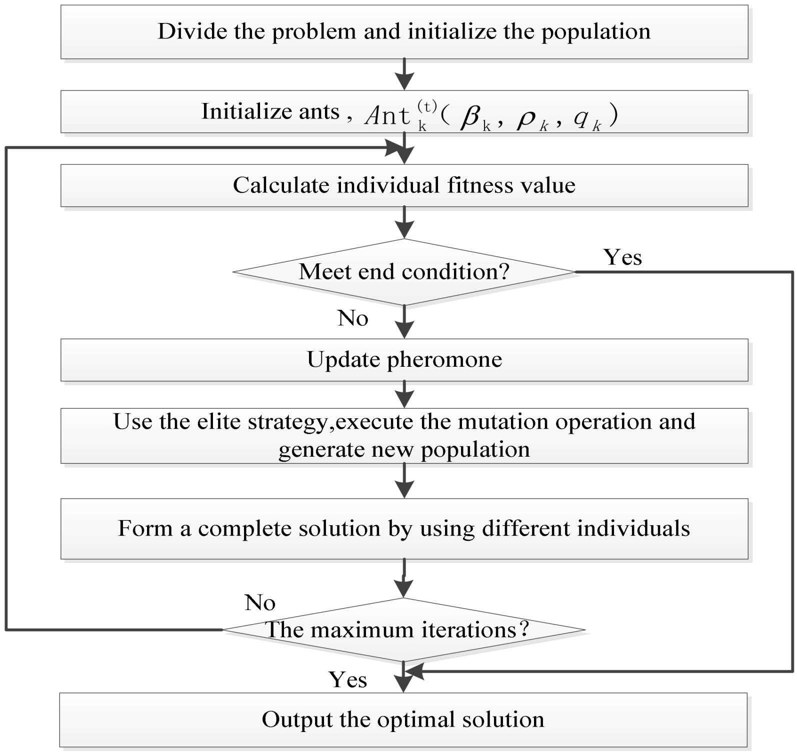

Figure 2. The flowchart of the SCEACO algorithm.

4.3. The Steps of the SCEACO Algorithm

In the proposed SCEACO algorithm, the multiple ant colonies have a common search space. Each ant colony uses the ACO algorithm to execute the search activity and pheromone update strategy. Within the population, the elitist strategy is used to retain some elitist individuals. The min-max ant strategy is used to set the pheromone concentration for each path. The traditional evolutionary algorithm is used as the evolutionary mechanism. The key of the SCEACO algorithm is the information exchange among sub-populations, and its strategy, content, and frequency after some iterations are executed. These are directly related to the efficiency and solution quality of the SCEACO algorithm. The steps of the SCEACO algorithm are described.

Step 1. The ant colony is divided into multiple sub-populations in a common search space, each sub-population performs the search activity and pheromone updating strategy. The multi-objective optimization problem is divided into several sub-optimization problems, then each sub optimization problem corresponds to one sub-population.

Step 2. Initialize the parameters of the SCEACO algorithm. These parameters include the control parameters α and β, ant size m, the pheromone trial evaporation rate ρ, the maximum iteration times $T_{\max}$, and the iteration algebraic counter $t = 0$. For the initialized number of ants, each ant stores these parameters in the form of $Ant_{\mathrm{k}}^{(t)}(\beta_{\mathrm{k}}, \rho_k, q_k)$.

Step 3. Calculate the fitness value of each individual in each sub-population, determine whether the result meets the end condition. If the result meets the end condition, then the result is output. Otherwise go to Step 4.

Step 4. The pheromone is updated for each individual according to the improved pheromone updating Equations (3)–(6).

Step 5. In each sub-population, the elitist strategy is used to retain some elitist individuals. The other ants are evolved to generate a new population.

Step 6. Each sub-population selects the current optimal individual, which is used to form a complete solution with the individual of different sub-population in order to complete the information interaction among these sub-populations.

Step 7. The min-max ant strategy is used to set pheromone concentration for each path

The pheromone concentration for each path is limited in the range $[\tau_{\min}, \tau_{\max}]$. The value of $\tau_{\max}$ can avoid the pheromone amount of one path with a much larger concentration than the other path in order to prevent concentrating all pheromones on the same path. The value of the $\tau_{\min}$ can effectively avoid the stagnation of the SCEACO algorithm.

Step 8. Determine whether the maximum number of iterations is reached. If the number of iterations is reached, then the result is output. Otherwise go to Step 3.

5. Application of the SCEACO Algorithm in Gate Assignment

5.1. Construct the Optimization Model of Gate Assignment

The gate assignment problem is a combinatorial optimization problem with multi-objective properties. Better gate assignment scheduling is beneficial to the perfect combination of safety and efficiency. In this paper, the most balanced idle time, the shortest walk distances, and the fewest number of flights at the apron are selected as the optimization objectives in order to respectively establish the corresponding objective functions. The objective function of the most balanced idle time is $F_1 = \min[\sum_{i=1}^{n}\sum_{k=1}^{m} S_{ik}^2 + \sum_{k=1}^{m} SS_k^2]$, the objective function of the shortest walk distances is $F_2 = \min\sum_{i=1}^{n}\sum_{k=1}^{m} q_{ik} f_k y_{ik}$, and the objective function of the fewest number of flights at the apron is $F_3 = \min\sum_{k=1}^{m} g_{\mathrm{i}}$.

For the objective functions of F_1, F_2, and F_3, the values of three different objective functions are not easily determined, and likely to be very different. Therefore, it is difficult to simply solve and obtain a very satisfactory optimal feasible solution by reasonably adjusting the weight factor. Thus, the non-quantized processing must be carried out here. Set the function $U = \sum_{q=1}^{3} W_q F_q$ and $F_q^0 = \max F_q(q = 1, 2, 3, F_q^0 \neq 0)$, the normalized objective function is $U' = \sum_{q=1}^{3} \frac{W_q F_q}{F_q^0}$. In the actual process, it is often difficult to simply determine F_1^0, F_2^0, and F_3^0. Therefore, the values of F_1, F_2, and F_3 need to be modified. Set $\mu_1 = \frac{W_1}{F_1^0}$, $\mu_2 = \frac{W_2}{F_2^0}$, and $\mu_3 = \frac{W_3}{F_3^0}$, and the normalized objective function is described as follows:

$$F = \mu_1[\sum_{i=1}^{n}\sum_{k=1}^{m} S_{ik}^2 + \sum_{k=1}^{m} SS_k^2] + \mu_2\sum_{i=1}^{n}\sum_{k=1}^{m} q_{ik} f_k y_{ik} + \mu_3\sum_{k=1}^{m} g_{\mathrm{i}} \quad (9)$$

5.2. Gate Assignment Method by Using the SCEACO Algorithm

The proposed SCEACO algorithm with global optimization ability is used to solve the multi-objective gate assignment model of a hub airport. The solving flowchart is shown in Figure 3.

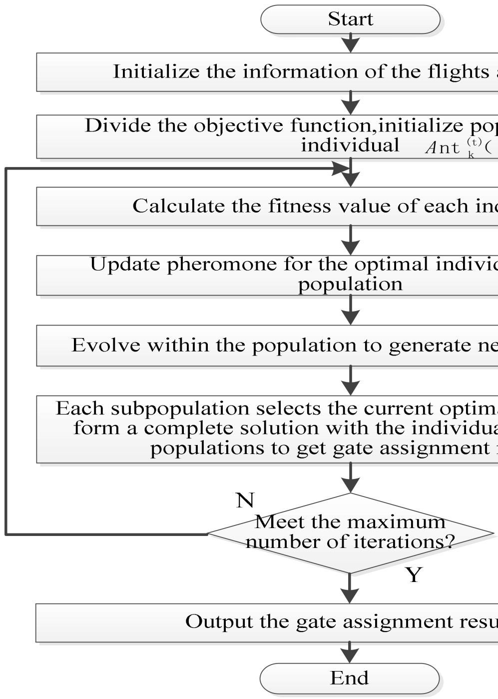

Figure 3. The solving flowchart of the multi-objective gate assignment model.

6. Case Analysis

6.1. Data Source and Experimental Environment

The experimental data came from Guangzhou Baiyun airport of China on 26 July 2015. There are 30 available gates and 201 flights, which are selected to verify the effectiveness of the gate assignment method based on the SCEACO algorithm. The gates are divided into large gates, medium gates, and small gates according to the size of the available aircraft, and the flights are also divided into large flights, medium flights, and small flights. The large gates can park all kinds of flights, the medium gate can park medium flights and small flights, and the small gate can only park small flights. The gates are divided into near gates and far gates according to the passenger's walking distance. If the passenger's walking distance is less than 950 m, the gate is a near gate. If the passenger's walking distance is more than 950 m, the gate is far. All flights that are not assigned to the gates can only be allocated to the apron. The information of gates are described in Table 1. The information of flights are described in Table 2.

Table 1. The information of the gates.

Gate	Type	Walking Distance (m)	Gate	Type	Walking Distance (m)
G1	M	190	G16	L	115
G2	M	975	G17	M	215
G3	L	400	G18	S	535
G4	M	333	G19	M	1050
G5	L	260	G20	M	170
G6	S	135	G21	L	585
G7	M	1100	G22	M	1250
G8	M	150	G23	L	500
G9	L	384	G24	L	920
G10	M	960	G25	L	270
G11	S	1000	G26	M	230
G12	L	235	G27	L	265
G13	S	1200	G28	L	450
G14	M	580	G29	M	1300
G15	M	440	G30	L	426

Table 2. The information of the flights.

Flight	Arrival Time	Departure Time	Walking Distance (m)	Type
F1	26 July 2015 0:05:00	26 July 2015 1:15:00	482	M
F2	26 July 2015 0:05:00	26 July 2015 1:45:00	273	S
F3	26 July 2015 0:10:00	26 July 2015 1:30:00	261	S
F4	26 July 2015 0:15:00	26 July 2015 1:30:00	116	S
F5	26 July 2015 0:15:00	26 July 2015 3:15:00	244	S
F6	26 July 2015 0:20:00	26 July 2015 1:30:00	312	M
F7	26 July 2015 0:25:00	26 July 2015 2:40:00	340	M
F8	26 July 2015 0:30:00	26 July 2015 1:00:00	198	S
F9	26 July 2015 0:35:00	26 July 2015 8:10:00	184	S
F10	26 July 2015 0:35:00	26 July 2015 10:55:00	494	M
F11	26 July 2015 0:40:00	26 July 2015 7:00:00	19	L
F12	26 July 2015 0:45:00	26 July 2015 6:40:00	443	L
⋮	⋮	⋮	⋮	⋮
F200	26 July 2015 19:30:00	26 July 2015 20:25:00	252	S
F201	26 July 2015 19:35:00	26 July 2015 20:25:00	378	M

6.2. Experimental Result

The SCEACO algorithm is used to solve the gate assignment model of a hub airport. The experiments were continuously carried out 20 times for solving the gate assignment model. The best time of the 20 experimental times was selected to analyze the effectiveness of the SCEACO algorithm and gate assignment model of a hub airport. The optimal objective value is 0.3160. The obtained assignment result for each gate is shown in Table 3 and Figure 4.

Table 3. The gate assignment results.

Gate	Flights										Total
G1	F38	F59	F78	F96	F140	F172					6
G2	F36	F116	F135	F159							4
G3	F34	F74	F95	F107	F134	F148					6
G4	F8	F32	F101	F110	F147	F173					6
G5	F15	F41	F56	F71	F94	F100	F113	F129	F157		9
G6	F31	F51	F126	F141	F169						5
G7	F17	F65	F89	F124	F179						5
G8	F1	F30	F46	F60	F164						5
G9	F10	F29	F40	F58	F70	F88	F139	F146	F156		9
G10	F16	F48	F90	F161							4
G11	F13	F37	F53	F145	F171						5
G12	F14	F61	F72	F92	F106	F115	F131	F144	F155	F193	10
G13	F3	F133	F151	F160	F189						5
G14	F28	F39	F50	F132	F152	F163	F198				7
G15	F18	F44	F68	F166	F185	F192					6
G16	F26	F42	F108	F119	F123	F162	F184				7
G17	F6	F45	F55	F67	F109	F150	F195				7
G18	F25	F82	F86	F167	F199						5
G19	F4	F112	F197								3
G20	F24	F43	F57	F69	F187						5
G21	FF23	F87	F105	F118	F125	F158	F188				7
G22	F22	F177	F183								3
G23	F9	F21	F33	F47	F52	F64	F73	F168	F181	F191	10
G24	F2	F99	F111	F122	F130	F149	F170	F190			8
G25	F11	F35	F54	F63	F83	F194					6
G26	F12	F27	F117	F128	F138	F176	F182				7
G27	F20	F49	F62	F80	F85	F153	F196				7
G28	F7	F75	F91	F104	F120	F127	F137	F201			8
G29	F19	F66	F154	F200							4
G30	F5	F93	F114	F121	F136	F165	F186				7
Total											186

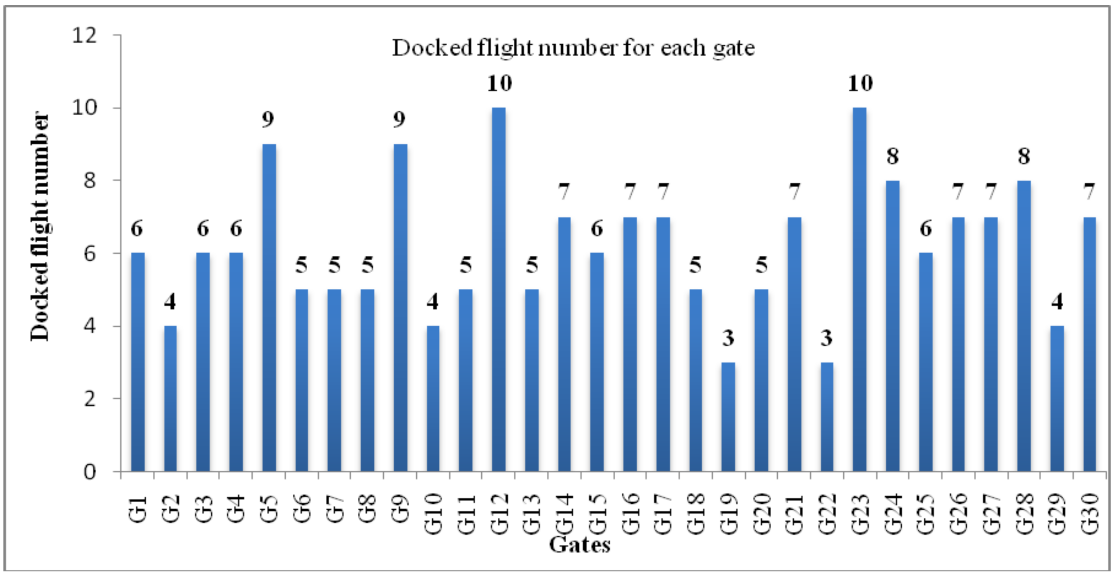

Figure 4. The assigned flights for each gate.

As can be seen from Table 3 and Figure 4, there are 186 flights, which are assigned to 30 gates. The minimum interval time is set T = 5 min between two adjacent flights for the same gate to avoid a conflict. There are 15 flights, which are allocated to the apron. The assigned efficiency for 201 flights reaches 92.5%. From the number of assigned results for each gate, the number of flights is more balanced for each gate. Twenty-two gates have been assigned to five flights or more than five flights. Gate 12 and Gate 23 can park 10 flights, Gate 5 and Gate 9 can park nine flights, and Gate 24 and Gate 28 can park eight flights. There are seven flights allocated to Gate 14, Gate 16, Gate 17, Gate 21, Gate 26, Gate 27, and Gate 30. There are six flights allocated to Gate 1, Gate 3, Gate 4, Gate 15, and Gate 25. There are five flights allocated to Gate 6, Gate 7, Gate 8, Gate 13, Gate 18, and Gate 20. There are four flights allocated to Gate 2, Gate 10, and Gate 29. There are three flights allocated to Gate 19 and Gate 22. Gates 2, 7, 10, 13, 19, 22, and 29 are farther from the check-in of all the gates, so the passengers need more time to arrive at these gates. Thus, these gates are assigned fewer flights. In general, the closer gates have a higher utilization rate. However, the excessive utilization to these gates will damage equipment and easily cause equipment failure. Therefore, the balanced utilization gate needs to be considered in the gate assignment model of a hub airport. As can be seen in the comprehensive assignment result, the gate assignment model of a hub airport based on the most balanced idle time, the shortest walk distances, and the fewest number of flights at the apron can improve the utilization efficiency and balance the rate of gates to a satisfactory degree for passengers. The proposed SCEACO algorithm can solve the gate assignment model of a hub airport faster and obtain the ideal assignment result of gates. It offers a better optimization performance in solving the gate assignment problem.

6.3. Comparison and Analysis of the Experimental Results

In order to further testify to the optimization performance of the SCEACO algorithm, the SCEACO algorithm is compared with the ACO algorithm and SACO algorithm. The parameters of the ACO algorithm and SACO algorithm are set the same as the SCEACO algorithm. The experiments were continuously carried out 20 times. The results are shown in Table 4 and Figure 5.

Table 4. The experimental results of the three algorithms.

Algorithms	ACO Algorithm			SACO Algorithm			SCEACO Algorithm		
Times	Iterations	Optimal Value	Flights	Iterations	Optimal Value	Flights	Iterations	Optimal Value	Flights
1	103	0.4010	155	32	0.3665	165	76	0.3216	181
2	126	0.3941	158	93	0.3686	161	83	0.3243	174
3	115	0.3962	153	98	0.3553	156	60	0.3269	171
4	58	0.3755	162	85	0.3662	151	21	0.322	177
5	64	0.3847	166	79	0.3689	170	58	0.3182	182
6	149	0.4070	155	61	0.3658	168	33	0.3266	173
7	121	0.3998	157	82	0.3764	162	64	0.3160	186
8	158	0.3938	166	114	0.3689	180	88	0.3225	175
9	74	0.3855	163	77	0.3613	162	18	0.3246	173
10	143	0.3987	167	83	0.3648	153	96	0.3185	183
11	69	0.3891	162	105	0.3565	167	10	0.3165	181
12	79	0.3973	154	100	0.3698	162	20	0.3227	180
13	156	0.3826	159	39	0.3471	167	37	0.3168	175
14	37	0.3990	163	74	0.3609	164	43	0.3169	176
15	103	0.3805	163	168	0.3742	164	88	0.3198	177
16	138	0.4016	155	41	0.3676	155	80	0.3207	183
17	65	0.3984	161	80	0.3556	163	76	0.3244	182
18	97	0.3988	157	73	0.3718	158	34	0.3217	183
19	141	0.3793	166	11	0.3632	155	28	0.3200	183
20	109	0.3974	158	73	0.3689	167	44	0.3244	173
Average	105.5	0.3930	160	78.4	0.3649	162.5	52.85	0.3213	178.4

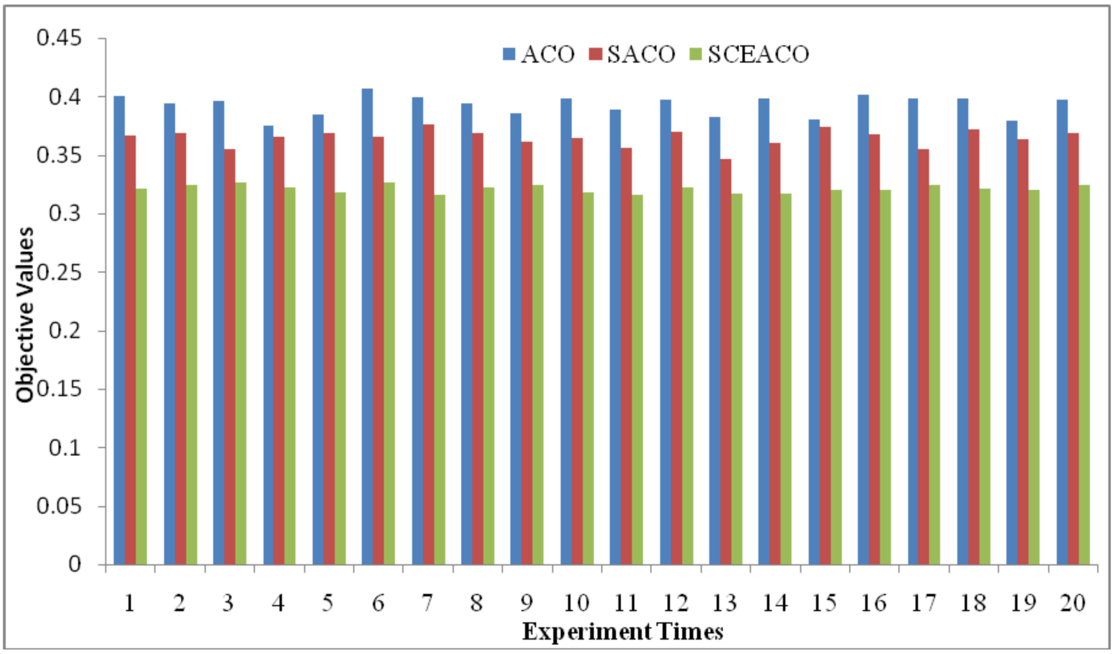

Figure 5. The comparison of optimal values for each time.

As can be seen from Table 4 and Figure 5, in the results of 20 experimental times, the optimal value by using the SECACO algorithm, 0.3160, is better than the ACO algorithm, 0.3755, and SACO algorithm, 0.3471, and the worst value by using the SECACO algorithm, 0.3269, is better than the ACO algorithm, 0.4070, and SACO algorithm, 0.3764. For the average value for the results of 20 experimental times, the average optimal value by using SECACO algorithm, 0.3213 is better than the ACO algorithm, 0.3930, and SACO algorithm, 0.3649. Therefore, the experiment results show that the proposed SECACO algorithm takes on a more outstanding optimization ability and better performance in solving the gate assignment problem. As can be seen from the number of iterations in Table 4, the average number of iterations by using the SECACO algorithm, 52.85, is less than the ACO algorithm, 105.5, and SACO algorithm, 78.4. Therefore, the experiment results show that the SECACO algorithm can obtain the optimal objective value faster than the ACO algorithm and SACO algorithm in solving the constructed gate assignment model of a hub airport. That is to say, the solution quality by using the SECACO algorithm is the best. The SECACO algorithm can effectively improve the comprehensive service ability of the gate assignment and takes on a more outstanding searching ability and faster optimization ability. The SECACO algorithm provides a new method for solving the complex optimization problem.

The change curve of the optimal values by using the ACO, SACO, and SECACO algorithms for solving the gate assignment model is shown in Figure 6.

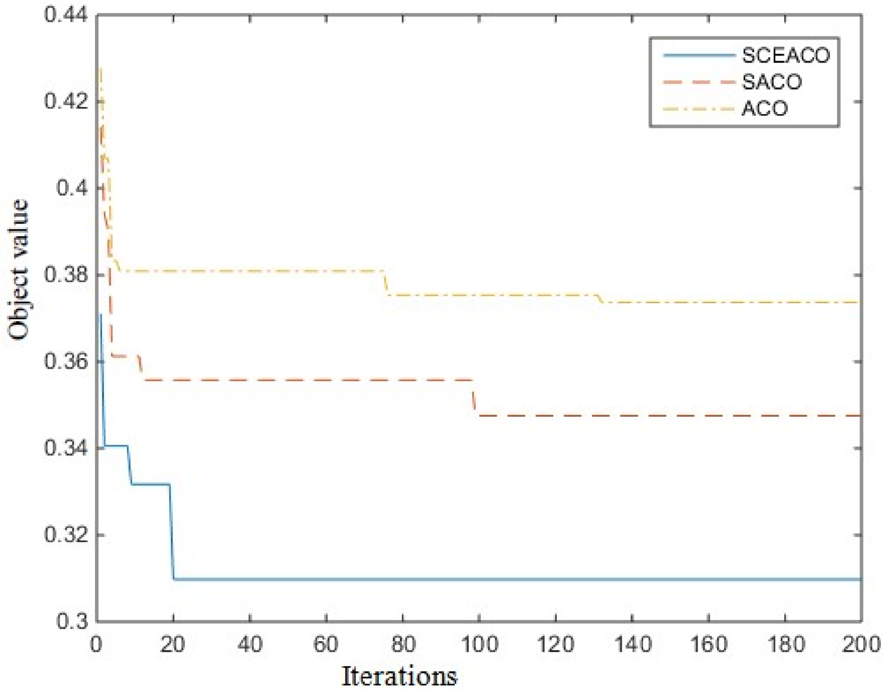

Figure 6. The change curve of the optimal values.

As can be seen from Figure 6, for solving the gate assignment model of a hub airport, the target value of the first iteration by using the SCEACO algorithm is superior to the ACO algorithm and SACO algorithm. When the ACO algorithm and SACO algorithm are used to solve the gate assignment model of a hub airport, the multi-objective function is established according to the gate assignment model of the hub airport in order to obtain the fitness value of each individual. When the fitness function is solved, the weights of each objective function also need to be considered. In this study, the SCEACO algorithm divides the optimization problem into several sub-problems in order to avoid the effect of the optimization problem. As can be seen from Figure 6, the SCEACO algorithm can obtain the optimal value in nearly 20 iterations. The convergence speed is obviously faster than the ACO algorithm and SACO algorithm. The mutation within the population can adjust the combination of parameters of the SCEACO algorithm in order to obtain a greater chance to search for a better path, which also reflects that the SCEACO algorithm has a more effective searching ability. Therefore, the SECACO algorithm can effectively solve the complex optimization problem and takes on a more outstanding searching ability and faster optimization ability. It provides a valuable reference for solving the complex optimization problem.

7. Conclusions

The ACO algorithm takes on the positive feedback, essential parallelism, and global convergence in solving optimization problems, but it has undetermined parameters, premature stagnation, and slow convergence speed. The CEA emphasizes the existing interaction among different sub-populations, but it is overly formal, and does not form a very strict and unified definition. In order to make use of the advantages of the elitist strategy, min-max ant strategy, co-evolutionary idea, symbiotic mechanism, and hybrid mechanism, a new adaptive co-evolutionary ant colony optimization (SCEACO) algorithm based on the complementary advantages is proposed in this paper. In the SCEACO algorithm, the pheromone update formula is improved and the update range of the pheromone is limited in order to realize the adaptive update of the pheromone. The multi-objective optimization problem is divided into several sub-problems with a corresponding population to perform the search activity and pheromone updating strategy. The elitist strategy is used to retain the elitist individuals within the population and the min-max ant strategy is used to set the pheromone concentration for each path. The selection, crossover, and mutation operations of the individual are used to adaptively adjust the parameters of the algorithm and implement the information sharing of the population and the co-evolution. The experiment results show that the SCEACO algorithm can assign 186 flights to 30 gates and the assigned efficiency of flights reaches 92.5%. The optimal value by using the SECACO algorithm, 0.3160, is better than the ACO algorithm and SACO algorithm. The SCEACO algorithm can better obtain the effective assignment result of the gates. It takes on a faster convergence speed and global search capability. This study provides a new idea for solving the complex optimization problem.

Due to the SCEACO algorithm having a higher time complexity in solving the gate assignment problem, the SCEACO algorithm needs to be studied further in order to reduce the time complexity. In future work, the SCEACO algorithm will be studied in depth.

Acknowledgments: The authors would like to thank all the reviewers for their constructive comments. This paper is supported by the National Natural Science Foundation of China (61771087, 51475065, 51605068), the Research Fund of the Guangxi Key Lab of Multi-source Information Mining and Security (MIMS17-03), the Open Project Program of Key Laboratory of Guangxi High Schools Complex System and Computational Intelligence, (2017CSCI03), the Open Project Program of the Guangxi Key Laboratory of Hybrid Computation and IC Design Analysis, and the Open Project Program of the Traction Power State Key Laboratory of Southwest Jiaotong University (TPL1705,TPL1803), and the Natural Science Foundation of Liaoning Province (20170540125, 20170540126, 20170540145). The program for the initialization, study, training, and simulation of the proposed algorithm in this article was written with the tool-box of MATLAB 2014a produced by the Math-Works, Inc.

Author Contributions: Huimin Zhao conceived and designed the experiments; Weitong Gao performed the experiments; Wu Deng wrote the paper; and Meng Sun analyzed the data.

Conflicts of Interest: The authors declare no conflict of interest. The founding sponsors had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Villarrubia, G.; De Paz, J.F.; Chamoso, P.; la Prieta, F.D. Artificial neural networks used in optimization problems. *Neurocomputing* **2018**, *272*, 10–16. [CrossRef]
2. Deng, W.; Chen, R.; He, B.; Liu, Y.Q.; Yin, L.F.; Guo, J.H. A novel two-stage hybrid swarm intelligence optimization algorithm and application. *Soft Comput.* **2012**, *16*, 1707–1722. [CrossRef]
3. Yu, Y.; Li, Y.C.; Li, J.C.; Gu, X.Y. Self-adaptive step fruit fly algorithm optimized support vector regression model for dynamic response prediction of magnetorheological elastomer base isolator. *Neurocomputing* **2016**, *211*, 41–52. [CrossRef]
4. Deng, W.; Li, B.; Zhao, H.M. Study on an airport gate reassignment method and its application. *Symmetry* **2017**, *9*, 258. [CrossRef]
5. Li, Y.C.; Zhou, X.Q.; Lu, M.M.; Lin, J.Q.; Sun, J.B. Parameters optimization for machining optical parts of difficult-to-cut materials by genetic algorithm. *Mater. Manuf. Process.* **2013**, *29*, 9–14. [CrossRef]
6. Dackermann, U.; Skinner, B.; Li, J.C. Guided wave-based condition assessment of in situ timber utility poles using machine learning algorithms. *Struct. Health Monit.* **2014**, *13*, 374–388. [CrossRef]
7. Dorigo, M.; Maniezzo, V.; Colorni, A. Ant system: Optimization by a colony of cooperating agents. *IEEE Trans. Syst. Man Cybern. Part B Cybern.* **1996**, *26*, 29–41. [CrossRef] [PubMed]

8. Lee, J.W. Heterogeneous-ants-based path planner for global path planning of mobile robot applications. *Int. J. Control Autom. Syst.* **2017**, *15*, 1754–1769. [CrossRef]
9. Liu, J.H.; Yang, J.G.; Li, H.P. An improved ant colony algorithm for robot path planning. *Soft Comput.* **2017**, *21*, 5829–5839. [CrossRef]
10. Jiao, L.C.; Wang, H.D.; Shang, R.H.; Liu, F. A co-evolutionary multi-objective optimization algorithm based on direction vectors. *Inf. Sci.* **2013**, *228*, 90–112. [CrossRef]
11. Deng, W.; Zhao, H.M.; Zou, L.; Li, G.Y.; Yang, X.H.; Wu, D.Q. A novel collaborative optimization algorithm in solving complex optimization problems. *Soft Comput.* **2017**, *21*, 4387–4398. [CrossRef]
12. Mercedes, E.N.; Miquel, A.P. Robust gate assignment procedures from an airport management perspective. *Omega* **2015**, *50*, 82–95.
13. Yang, S.R.; Cho, S.B. Co-evolutionary learning with strategic coalition for multiagents. *Appl. Soft Comput. J.* **2005**, *5*, 193–203. [CrossRef]
14. He, Q.; Wang, L. An effective co-evolutionary particle swarm optimization for constrained engineering design problems. *Eng. Appl. Artif. Intell.* **2007**, *20*, 89–99. [CrossRef]
15. Huang, F.Z.; Wang, L.; He, Q. An effective co-evolutionary differential evolution for constrained optimization. *Appl. Math. Comput.* **2007**, *186*, 340–356. [CrossRef]
16. Kou, X.L.; Liu, S.Y.; Zhang, J.K.; Zheng, W. Co-evolutionary particle swarm optimization to solve constrained optimization problems. *Comput. Math. Appl.* **2009**, *57*, 1776–1784. [CrossRef]
17. Gu, J.W.; Gu, M.Z.; Cao, C.W.; Gu, X.S. A novel competitive co-evolutionary quantum genetic algorithm for stochastic job shop scheduling problem. *Comput. Oper. Res.* **2010**, *37*, 927–937. [CrossRef]
18. Coelho, L.S.; Bernert, D.L.A. A modified ant colony optimization algorithm based on differential evolution for chaotic synchronization. *Expert Syst. Appl.* **2010**, *37*, 4198–4203. [CrossRef]
19. Xing, L.N.; Chen, Y.W.; Yang, K.W. Multi-population interactive coevolutionary algorithm for flexible job shop scheduling problems. *Comput. Optim. Appl.* **2011**, *48*, 139–155. [CrossRef]
20. Li, P.C.; Wang, H.Y. Quantum ant colony optimization algorithm based on bloch spherical search. *Neural Netw. World* **2012**, *22*, 325–341. [CrossRef]
21. Liao, T.W.; Kuo, R.J.; Hu, J.T.L. Hybrid ant colony optimization algorithms for mixed discrete-continuous optimization problems. *Appl. Math. Comput.* **2012**, *219*, 3241–3252. [CrossRef]
22. Chandra, R.; Frean, M.; Zhang, M.J. Crossover-based local search in cooperative co-evolutionary feedforward neural networks. *Appl. Soft Comput. J.* **2012**, *12*, 2924–2932. [CrossRef]
23. Sun, Y.; Zhang, L.B.; Gu, X.S. A hybrid co-evolutionary cultural algorithm based on particle swarm optimization for solving global optimization problems. *Neurocomputing* **2012**, *98*, 76–89. [CrossRef]
24. Gao, Y.L.; Wang, M.M. Hybrid coding PSO-ACO co-evolutionary algorithm for solving mixed-integer programming problems. *Asia-Pac. J. Oper. Res.* **2013**, *30*, 1340001. [CrossRef]
25. Fernández-Vargas, J.A.; Bonilla-Petriciolet, A.; Segovia-Hernández, J.G.S. An improved ant colony optimization method and its application for the thermodynamic modeling of phase equilibrium. *Fluid Phase Equilib.* **2013**, *353*, 121–131. [CrossRef]
26. Rizk-Allah, R.M.; Zaki, E.M.; El-Sawy, A.A. Hybridizing ant colony optimization with firefly algorithm for unconstrained optimization problems. *Appl. Math. Comput.* **2013**, *224*, 473–483. [CrossRef]
27. Juang, C.F.; Hung, C.W.; Hsu, C.H. Rule-based cooperative continuous ant colony optimization to improve the accuracy of fuzzy system design. *IEEE Trans. Fuzzy Syst.* **2014**, *22*, 723–735. [CrossRef]
28. Zhao, W.J.; Alam, S.; Abbass, H.A. MOCCA-II: A multi-objective co-operative co-evolutionary algorithm. *Appl. Soft Comput. J.* **2014**, *23*, 407–416. [CrossRef]
29. Ding, W.P.; Guan, Z.J.; Shi, Q.; Wang, J.D. A more efficient attribute self-adaptive co-evolutionary reduction algorithm by combining quantum elitist frogs and cloud model operators. *Inf. Sci.* **2015**, *293*, 214–234. [CrossRef]
30. Bu, Y.; Li, T.Q.; Zhang, Q. A weighted max-min ant colony algorithm for TSP instances. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.* **2015**, *98*, 894–897. [CrossRef]
31. Jiang, W.Y.; Lin, Y.; Chen, M.; Yu, Y.Y. A co-evolutionary improved multi-ant colony optimization for ship multiple and branch pipe route design. *Ocean Eng.* **2015**, *102*, 63–70. [CrossRef]
32. Wan, Y.C.; Wang, M.W.; Ye, Z.W.; Lai, X.D. A feature selection method based on modified binary coded ant colony optimization algorithm. *Appl. Soft Comput. J.* **2016**, *49*, 248–258. [CrossRef]

33. Pan, Q.K. An effective co-evolutionary artificial bee colony algorithm for steelmaking-continuous casting scheduling. *Eur. J. Oper. Res.* **2016**, *250*, 702–714. [CrossRef]
34. Lei, H.T.; Wang, R.; Zhang, T.; Liu, Y.J.; Zha, Y.B. A multi-objective co-evolutionary algorithm for energy-efficient scheduling on a green data center. *Comput. Oper. Res.* **2016**, *75*, 103–117. [CrossRef]
35. Yu, B.; Zhao, H.; Xue, D.Y. A multi-population co-evolutionary genetic programming approach for optimal mass customisation production. *Int. J. Prod. Res.* **2017**, *55*, 621–641. [CrossRef]
36. Goran, M.; Tihana, G.G. Co-evolutionary multi-population genetic programming for classification in software defect prediction: An empirical case study. *Appl. Soft Comput. J.* **2017**, *55*, 331–351.
37. Nilakantan, J.M.; Li, Z.X.; Tang, Q.H.; Nielsen, P. Multi-objective co-operative co-evolutionary algorithm for minimizing carbon footprint and maximizing line efficiency in robotic assembly line systems. *J. Clean. Prod.* **2017**, *156*, 124–136. [CrossRef]
38. Hiew, B.Y.; Tan, S.C.; Lim, W.S. A double-elimination-tournament-based competitive co-evolutionary artificial neural network classifier. *Neurocomputing* **2017**, *249*, 345–356. [CrossRef]
39. Yang, Q.; Chen, W.N.; Yu, Z.T.; Gu, T.L.; Li, Y.; Zhang, H.X.; Zhang, J. Adaptive multimodal continuous ant colony optimization. *IEEE Trans. Evolut. Comput.* **2017**, *21*, 191–205. [CrossRef]
40. Xu, M.L.; You, X.M.; Liu, S. A novel heuristic communication heterogeneous dual population ant colony optimization algorithm. *IEEE Access* **2017**, *5*, 18506–18515. [CrossRef]
41. Ding, W.P.; Lin, C.T.; Chen, S.B.; Zhang, X.F.; Hu, B. Multiagent-consensus-MapReduce-based attribute reduction using co-evolutionary quantum PSO for big data applications. *Neurocomputing* **2018**, *272*, 136–153. [CrossRef]
42. Xue, Y.; Jiang, J.M.; Zhao, B.P.; Ma, T.H. A self-adaptive artificial bee colony algorithm based on global best for global optimization. *Soft Comput.* **2018**, *22*, 2935–2952. [CrossRef]
43. Yu, Y.; Li, Y.C.; Li, J.C. Nonparametric modeling of magnetorheological elastomer base isolator based on artificial neural network optimized by ant colony algorithm. *J. Intell. Mater. Syst. Struct.* **2015**, *26*, 1789–1798. [CrossRef]
44. Deng, W.; Zhao, H.M.; Yang, X.H.; Xiong, J.X.; Sun, M.; Li, B. Study on an improved adaptive PSO algorithm for solving multi-objective gate assignment. *Appl. Soft Comput.* **2017**, *59*, 288–302. [CrossRef]
45. Omidvar, M.N.; Li, X.D.; Mei, Y.; Yao, X. Cooperative Co-Evolution with differential grouping for large scale optimization. *IEEE Trans. Evolut. Comput.* **2014**, *18*, 378–393. [CrossRef]
46. Stutzle, T.; Dorigo, M. A short convergence proof for a class of ant colony optimization algorithms. *IEEE Trans. Evolut. Comput.* **2000**, *6*, 358–365. [CrossRef]
47. Twomey, C.; Stützle, T.; Dorigo, M.; Manfrin, M.; Birattari, M. An analysis of communication policies for homogeneous multi-colony ACO algorithms. *Inf. Sci.* **2010**, *180*, 2390–2404. [CrossRef]
48. Wong, K.Y.; See, P.C. A new minimum pheromone threshold strategy (MPTS) for max-min ant system. *Appl. Soft Comput.* **2009**, *9*, 882–888. [CrossRef]