Applied Soft Computing Journal 76 (2019) 193–204

Contents lists available at ScienceDirect

Applied Soft Computing Journal

journal homepage: www.elsevier.com/locate/asoc

Complex network oriented artificial bee colony algorithm for global bi-objective optimization in three-echelon supply chain ☆

Jianhua Jiang [a], Di Wu [a], Yujun Chen [a], Keqin Li [b,*]

[a] School of Management Science and Information Engineering, Jilin University of Finance and Economics, Changchun 130117, China
[b] Department of Computer Science, State University of New York, New Paltz, NY 12561, USA

HIGHLIGHTS

- The complexity of a three-echelon SCM network is reduced greatly.
- The capability of scout bees is enhanced by gradient descent approach.
- The convergence and exploration capability is enhanced by simulated annealing approach.
- A real three-layer Bulldozer supply chain is taken to prove its excellent performance.

ARTICLE INFO

Article history:
Received 9 June 2018
Received in revised form 17 October 2018
Accepted 7 December 2018
Available online 18 December 2018

MSC:
00-01
99-00

Keywords:
Artificial bee colony algorithm
Bi-objective optimization
Complex network
Three-echelon supply chain
Gradient descent
Simulated annealing

ABSTRACT

Finding the best flow patterns (i.e., choices of resources) for a family of products is a key part of supply chain management. It primarily focuses on reasonable selecting suppliers for every component, selecting plants for assembling every sub- or final assembly, and selecting the delivery options to bring products to customers. Different selecting operations form different cost and lead-time. Balancing a trade-off between cost and lead-time is a non-trivial problem in a three-echelon supply chain, which forms a complex network. We focus on finding the best flow patterns in which reasonable selections can be formed together to provide products or services. The objective is to minimize the bi-objective of cost and lead-time for any product. In this paper, we propose a complex network oriented artificial bee colony algorithm, which can be processed in parallel, to tackle the so-called combinatorial problem. Besides, we employ *simulated annealing* and *gradient descent* to find global Pareto optimal solutions in a supply chain network. Extensive experiments on the three-echelon supply chain network demonstrate the superiority of our proposals: (1) the proposed CN-ABC and CN-ABC-SAGD have the capability of discovering global POS in a complex three-echelon SCN; (2) the speed of searching global POS is accelerated to satisfy the requirement of its complexity of a SCN.

1. Introduction

With the development of electronic commerce, the number of suppliers increase rapidly, hence supply chain network (SCN) becomes more and more complex. The supply chain operation management characterized by manufacturing resources, processing and delivering needs to be optimized based on complex network structure [1,2]. It is more and more important to help the decision makers to choose the suppliers, which not only can protect the profits of the enterprises, but also can guarantee the service level of the enterprises. Therefore, finding an efficient optimization approach is necessary to satisfy the emerging complexity requirement of supply chain operation management.

The optimization problem in a SCN refers to discover the best flow patterns (i.e., choices of resources) for a family of products [3,4]. The flow pattern in the SCN management involves selection through which materials (raw materials, work in progress, and finished products) and information (demand data, due date, delivery, assembly cost and lead time) in order to satisfy its multi-objective functions [5–7]. In order to determine an efficient flow pattern for every product in a family, it depends on reasonable select operations: (1) the selection of a supplier (or suppliers) for every component required by the product mix, (2) the selection of a manufacturing plant (or plants) for assembling every sub- or final assembly in the product mix, and (3) the selection of delivery options to customers [4,8]. In a typical SCN, there often exist many

☆ The authors are grateful to the financial support by the National Natural Science Foundation of China (no. 61572225), the Natural Science Foundation of Jilin Province, China (no. 20180101044JC), the Social Science Foundation of Jilin Province, China (no. 2017BS28), the Science Foundation of Education Department in Jilin Province, China (no. JJKH20180465KJ) and Foundation from Jilin University of Finance and Economics, China (no. 2018Z05).

* Corresponding author.
E-mail address: lik@newpaltz.edu (K. Li).

https://doi.org/10.1016/j.asoc.2018.12.006
1568-4946/

suppliers that could supply the same raw materials or components, multiple manufactures that could assemble sub- or final products, and a number of deliveries to bring products to customers. These different combinatorial selections in the flow pattern will generate different cost and lead-time. Hence, the optimization in a SCN is a combinatorial optimization problem, and it is a NP-hard problem.

Henceforth, it is necessary to design a bi-objective optimization algorithm to find the global Pareto optimal solutions (POS) with the objective of minimizing and balancing both cost and lead-time. This is a non-trivial task due to its complexity, which requires simultaneous optimization of cost and time, which are often contradicting with each other. Further complexity of the problem includes the involvements of multiple products of complex hierarchy, sharing common components and sub-assemblies, and the existence of a large number of resource options across a SCN [3,4]. Aggregating all these objectives through weighted sum can transform the bi-objective problem into a single objective problem. Every objective is multiplied by a weighted factor, and the sum of the weighted object is the objective function [7–11]. This alternative method is a collection of different criteria under certainty, which is making for decision-maker to search trade-off solutions and to get global POS.

Existing literatures have been proposed to solve the bi-objective combinatorial optimization problem using various algorithms in supply chain design. For instance, Shaw et al. [12] propose an integrated approach for selecting appropriate suppliers in a SCN, using fuzzy-AHP and fuzzy multi-objective linear programming, that does not consider the structural nature of complex network. Yeh et al. [13] introduce green criteria into the framework of supplier selection criteria, that does not consider optimization with metaheuristics. Some metaheuristic approaches, such as ant colony optimization (ACO [14]), genetic algorithm (GA [15,16]), have been proposed to minimize the total supply chain cost and lead time simultaneously in order to ensure product deliveries without delays. They note that bi-objective can be optimized by ACO, but their study should be improved by its searching speed for the complexity of a SCN. Yuce et al. [17] use the artificial bee colony (ABC) algorithm to deal with the bi-objective supply chain model to search the optimum configuration of a given SCN problem that can minimize the total cost and the total lead-time, but it does not improve the basic ABC for the specific SCN problem.

The majority of studies on SCN have considered only one product and one manufacturing center [18]. But, in reality most optimization problems involve more than one product, have a number of manufacturing centers in different regions, and bring different products to different customers. Henceforth, our study focuses on three-echelon SCN with multiple products, manufacturing centers and customers with the global Pareto optimization with bi-objective of total cost and lead-time.

To solve these above problem, the present paper proposed a complex network oriented artificial bee colony algorithm (CN-ABC) for parallel computing in a SCN. Furthermore, to solve the disadvantage of ABC algorithm with easily falling into local optimal solutions, the idea of *simulated annealing* is adopted to increase the capability of *exploration* that can find global POS in a SCN. To increase the convergence speed of finding global POS, the idea of *gradient descent* is adopted to update its selection probability for each choice in a node for a SCN. These two enhancements forms a complex network oriented artificial bee colony algorithm with simulated annealing and gradient descent (CN-ABC-SAGD). The major contributions of this paper can be highlighted as follows:

(i) Solutions are modelled with complex network structure to satisfy the requirement of the SCN nature;
(ii) The proposed CN-ABC and CN-ABC-SAGD have the capability of discovering global POS in a complex three-echelon SCN;
(iii) The convergence speed is accelerated to satisfy the increasing complexity of a SCN.

The organization of this paper is as follows: materials and methods are depicted in Section 2; the optimized result of the test example is given in Section 3; the simulated results are analysed in Section 4; and the final conclusion is drawn in the last section.

2. Materials and methods

2.1. Model description

This paper simulates a supply chain network which is formed by the operation process of a Bulldozer manufacturing enterprise. The whole network, which includes the initial raw material procurement, semi-finished products and the distribution of the final target market, is a complete supply chain with three layers structure. In order to get closer to the real situation, we simulate the supply chain composed of different groups, each group represents different spare parts. Each group is made up of different nodes, which act as different suppliers of the group, as shown in Fig. 1.

In Fig. 1, a_{22}, a_{26}, a_{35} are three products supplied by the Bulldozer company, such as Wheel Loader (WHL), Track Loader (TRL) and Track-Type Tractor (TTT). $\{a_1, a_2, a_3, \ldots, a_{16}, a_{17}\}$ are the common raw materials and intermediate products of these three products. $a_{18}, a_{19}, a_{20}, a_{21}, a_{25}, a_{31}, a_{32}, a_{33}, a_{34}$ are the respective raw materials and intermediate products of the three products. Each directed line in Fig. 1 represents the transfer of the raw material between groups, that is, the product produced by the directed group requires the product of the above grade group as the raw material. Therefore, the supply chain we simulate consists of 38 groups, and each node has its own cost and lead-time.

In this paper, the three layer structure of the supply chain is regarded as a complex network. Groups on the supply chain are represented by $\{a_1, a_2, \ldots, a_n\}$, and a_i is not related to each other. The total cost and total lead-time are chosen as the indicator to measure the profit and service quality as Eq. (1).

$$Z = \omega_1 \times TC + \omega_2 \times LT \tag{1}$$

where ω_1 and ω_2 are the two parameters for balancing cost and lead-time, and the sum of them equals to 1. TC is the **total cost** in supply chain, while LT is the **total lead-time**. For each group a_i, TC_i and LT_i are calculated as Eqs. (2) and (3):

$$TC_i = \mu_i \sum_{j=1}^{N_j} C_{ij} \times y_{ij} \tag{2}$$

In Eq. (2), each node in the group is marked by $a_{ij}(j = 0, 1, 2, \ldots, N_j)$, where N_j is the number of alternative suppliers in a_i, μ_i denotes average quantity demanded per day when calculating cost, C_{ij} is the cost for each selection in a_i, y_{ij} is used as a Boolean variable for judging whether the node is selected. If the node is selected, $y_{ij} = 1$, otherwise $y_{ij} = 0$.

$$LT_i = \sum_{j=1}^{N_j} T_{ij} \times y_{ij} + \max_{k \in S} LT_k \tag{3}$$

where T_{ij} is the lead-time of each selection in the node a_i, N_j and y_{ij} are the same as Eq. (2), S is the set of all child nodes of the node a_i, LT_k is the lead-time of a child node with a_i. Therefore, LT_i is calculated by putting its own lead-time and the maximal lead-time from its child nodes.

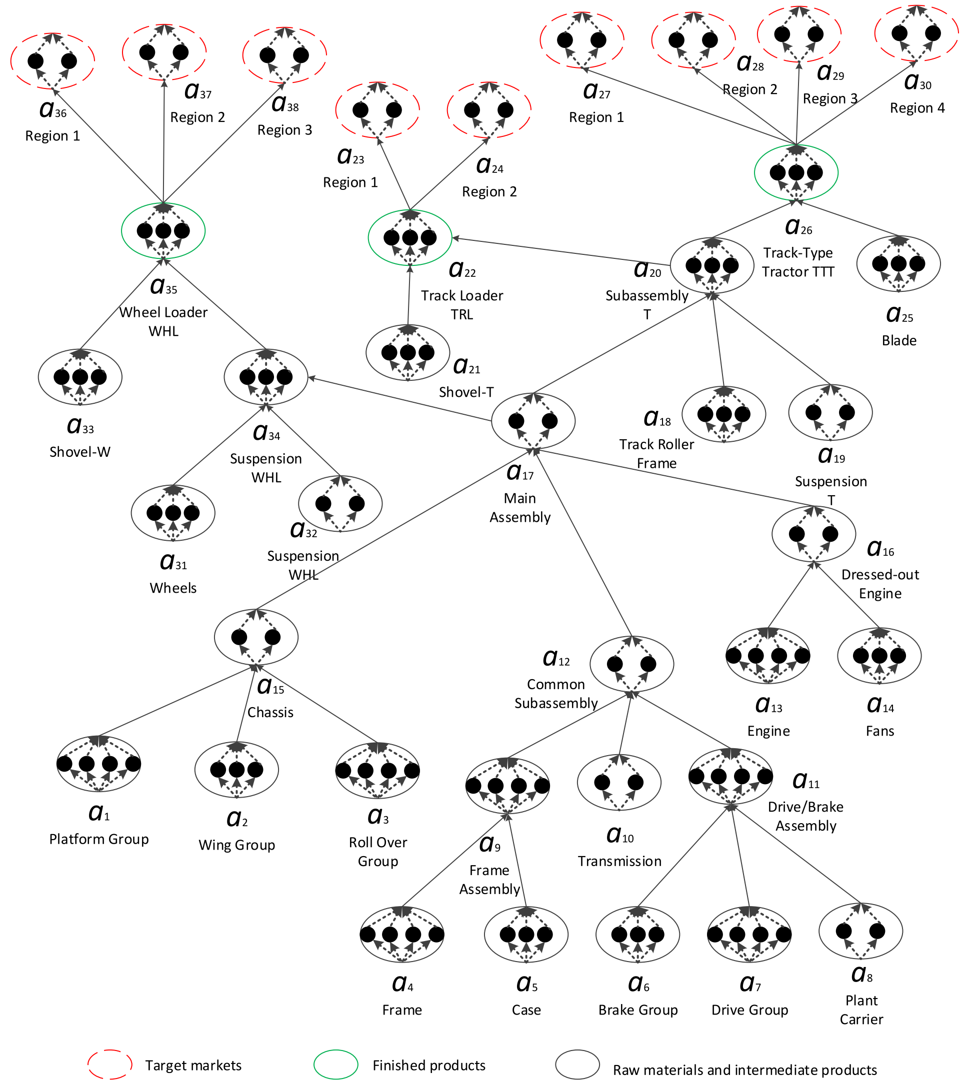


Fig. 1. Supply chain network for Bulldozer manufacturing enterprise.

2.2. Expressed solutions

In the proposed algorithms of CN-ABC and CN-ABC-SAGD, a set of feasible solutions, named as "solution vector", is created. The solution vector is structured as fragments of supply chain network. The supply chain network can be divided into different sub-networks. For instance, as illustrated in Fig. 1, a_{35}, a_{22} and a_{26} are three assembling nodes for WHL, TRL and TTT respectively, and these three products are assembled by a range of common supply nodes, e.g. $\{a_1, a_2, a_3, \ldots, a_{16}, a_{17}\}$. Therefore, this sub-network for all three products is the fragment from node a_1 to a_{17}. The sub-network for both products of TRL and TTT is the fragment from node a_{18} to a_{20}. The sub-network for the unique product WHL is the fragment from node a_{31} to a_{38}, the sub-network for the unique product TRL is the fragment from node a_{21} to a_{24}, and the sub-network for the unique product TTT is the fragment from node a_{25} to a_{30}. For an universal complex network oriented solutions, it can be modelled as Fig. 2. For instance, the value of $m+n+k+t+z$ is 38 in Fig. 1.

In Fig. 2, every Sub-Network can be modelled as feasible solutions as Fig. 3. In Fig. 3, $\{SV_1, SV_2, \ldots, SV_L\}$ represents all detected solution vectors (SV) for the solution space of the SCN problem, $SV_i (i \in L)$ indicates the ith iteration of the maximal iteration of L. For any solution vector SV_i, it has three parts, such as option sequence, fitness value and Bayesian probability sequence. Option sequence of $\{O_1, O_2, \ldots, O_k\}$ is the selection sequence formed by its selection of each node a_i in a Sub-Network, and Bayesian probability sequence of $\{P_1, P_2, \ldots, P_k\}$ is the probability sequence with its initial probability of equal chance for each selection. Fitness value records the evaluation result based on Eq. (1) for its option sequence. After the fitness value evaluation, both option sequence and Bayesian probability sequence will be changed based on the principle of the proposed CN-ABC and CN-ABC-SAGD algorithms.

2.3. Methods

In order to solve the problems mentioned in Section 2.1, the proposed algorithms are mainly based on ABC algorithm, use the related technology of complex network to strengthen the mutual connection between supply chain structure which can simplify the process of supply chain decision-making. And by adding the idea of *gradient descent* and *simulated annealing*, the search result set of the ABC algorithm is greatly enlarged while the convergent speed is speeded up as far as possible.

2.3.1. A brief introduction to basic ABC

Artificial bee colony (ABC) algorithm is a novel optimization method which is proposed by Karaboga [19]. It mainly mimicks the

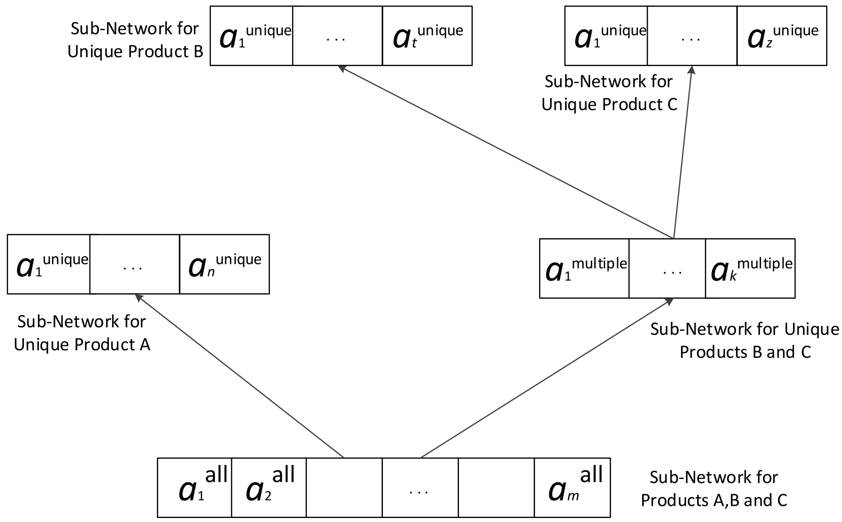


Fig. 2. Different types of sub-networks for a complex network.

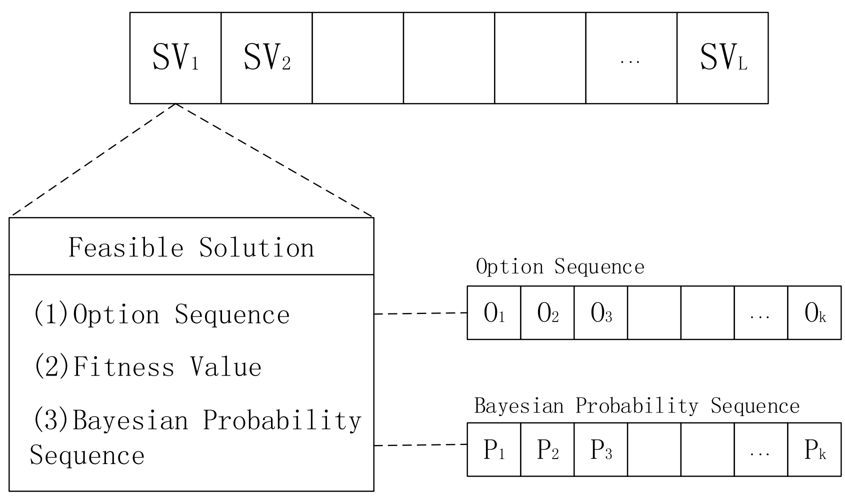


Fig. 3. Feasible solution for sub-networks of a SCN.

behaviour of honey collection by bees, through the optimization of the behaviour of the individual bees and ultimately making the global optimal value in the population, has a faster convergence rate. Karaboga et al. [20] have shown that ABC has been widely applied for feature selection [21], classification [22,23], real-parameter optimization [24], job scheduling [25], travelling salesman problem [26], and combinatorial problems [27]. The previous literature [17,27] has proved that the ABC algorithm has a very good effect on combinatorial optimization problems, such as supply chain optimization issue. Even more, with the explosive growth in supply chain network, this type of combinatorial optimization that is known to be NP-hard. Therefore, using ABC algorithm as the major solution is a good choice.

In the basic ABC algorithm, the number of employed bees or the onlooker bees is equal to the number of solutions in the swarm. Each employed bee X_i generates a new candidate solution V_i in the neighbourhood of its present position as Eq. (4).

$$V_{i_k} = X_{i_k} + \phi_{i_k} \times (X_{i_k} - X_{j_k}) \tag{4}$$

where X_j is a randomly selected candidate solution ($i \neq j$), k is a random dimension index selected from the set $\{1, 2, \ldots, n\}$, and ϕ_{i_k} is a random number within $[-1,1]$.

This probabilistic selection is really a roulette wheel selection mechanism which is described as Eq. (5).

$$P_i = \frac{fit_i}{\sum_j^n fit_j} \tag{5}$$

where fit_i is the fitness value of X_i. If a position cannot be improved over a predefined number (called limit) of cycles, then the food source is abandoned. Assume that the abandoned source is X_i, and then the scout bee discovers a new food source to be replaced with X_i as Eq. (6).

$$X_{i_k} = lb_j + rand(0, 1) \times (ub_j - lb_j) \tag{6}$$

where $rand(0, 1)$ is a random number within [0,1] based on a normal distribution, while lb_j and ub_j are lower and upper boundaries of the jth dimension respectively.

2.3.2. A brief introduction to basic gradient descent algorithm

Gradient descent algorithm (GA) [28] is an optimization algorithm to solve the minimum or maximum value along the direction of gradient descent. This optimization method converges quickly in the initial stage, but the convergence speed will slow down near the optimal value. GA is an iterative optimization algorithm for optimization problems: $minf(w)$. The equation of gradient descent

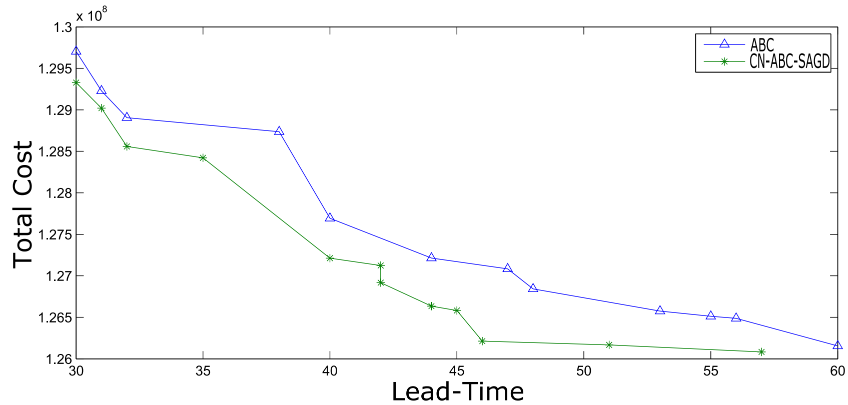
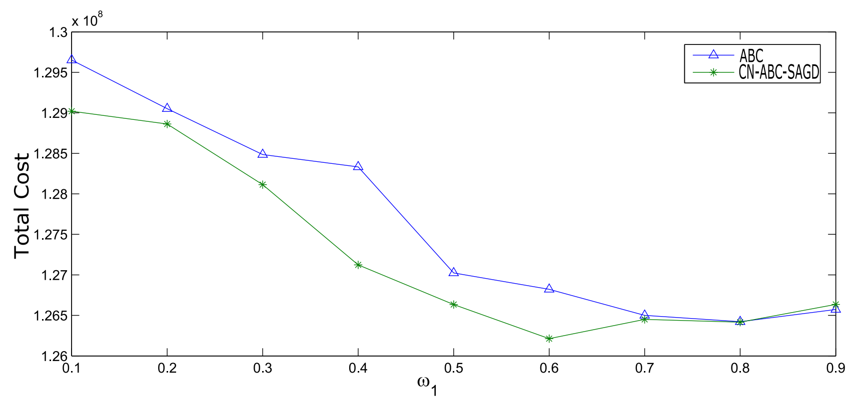
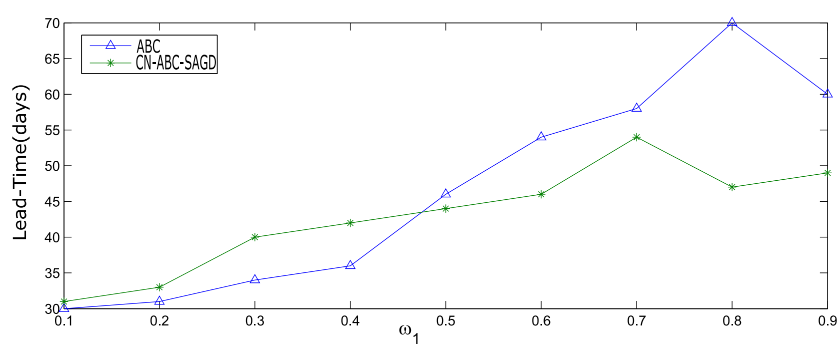


Fig. 4. Pareto fronts for ABC and CN-ABC-SAGD.



(a) Global optimal solutions of TC(in \$) with different ω_1



(b) Global optimal solutions of LT(in days) with different ω_1

Fig. 5. Global optimal solutions with ABC and CN-ABC-SAGD.

direction is shown in Eq. (7). Finally, GA can find the optimal solution by updating formula shown in Eq. (8).

$$d_i = -\frac{\partial}{\partial w} f(w)|_{w_i} \tag{7}$$

$$w_{i+1} = w_i + \rho \times d_i \tag{8}$$

where ρ is the step length.

2.3.3. A brief introduction to basic simulated annealing algorithm

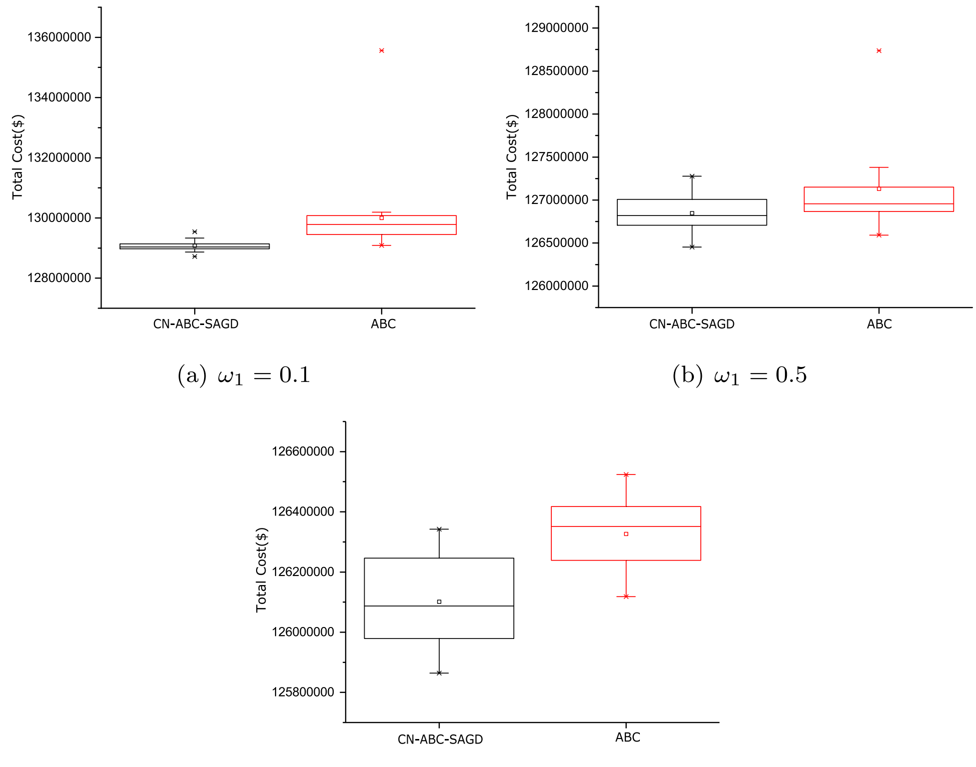
Simulated annealing (SA) [29] is a general probability algorithm for finding the optimal solution in a large search space. The SA algorithm is derived from the simulation of the annealing process in thermodynamics. In SA, T is an initial temperature, S is the initial solution, S' is a new solution. The incremental temperature $\triangle t'$ is defined by Eq. (9).

$$\triangle t' = C(S') - C(S) \tag{9}$$

where $C(S)$ is an evaluation function, S is replaced with S' if $\triangle t' < 0$, otherwise, S is replaced with S' according to probability formula of $exp(-\frac{\triangle t'}{T})$. The optimal solution so far will be found through L iterations in polynomial time.

2.3.4. The proposed CN-ABC-SAGD algorithm

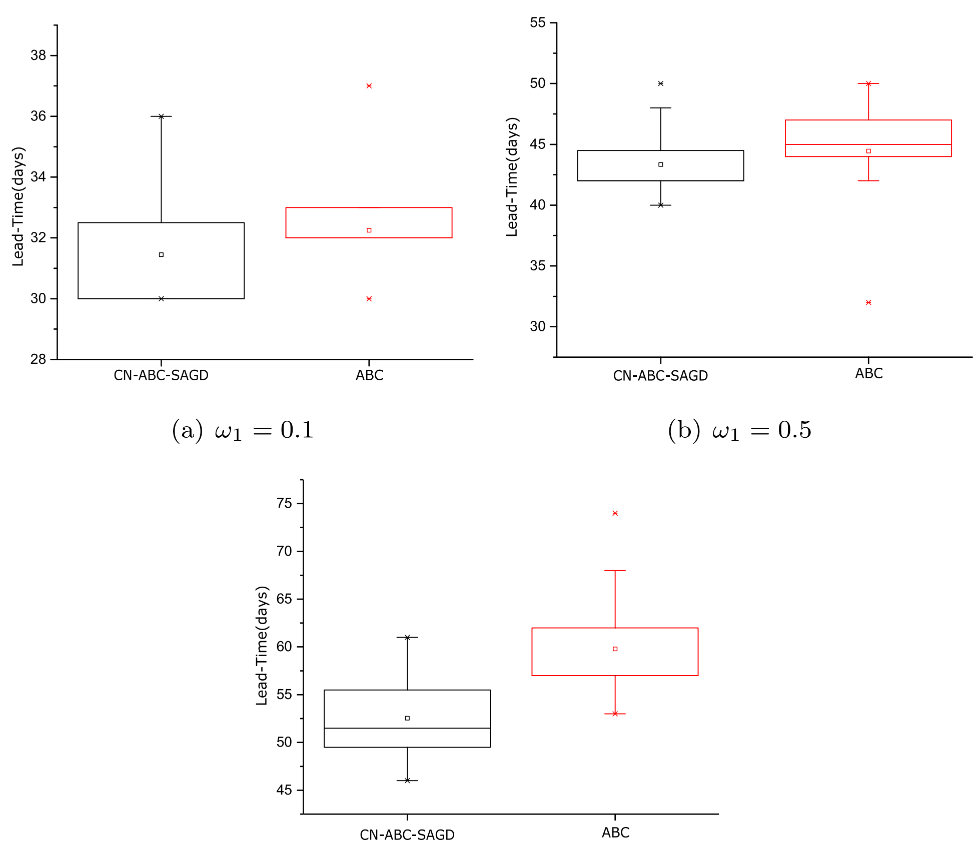
We propose a CN-ABC-SAGD algorithm on the basis of the advantages and disadvantages of the algorithms mentioned in the above Sections 2.3.1, 2.3.2 and 2.3.3. Firstly, we combine the gradient descent algorithm to make the scout bees move in the direction of the most probable optimal solution with increasing



(a) $\omega_1 = 0.1$ (b) $\omega_1 = 0.5$

(c) $\omega_1 = 0.9$

Fig. 6. Global optimal solutions with ABC and CN-ABC-SAGD.



(a) $\omega_1 = 0.1$ (b) $\omega_1 = 0.5$

(c) $\omega_1 = 0.9$

Fig. 7. Box plots of the LT solution obtained by ABC and CN-ABC-SAGD.

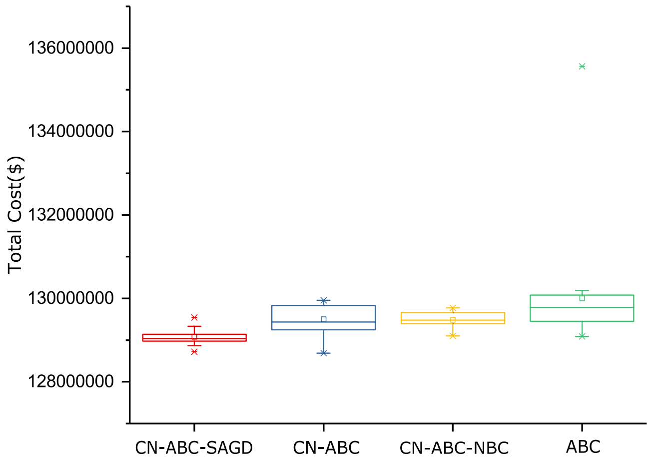


Fig. 8. Box diagram of *TC* solution set obtained when $\omega_1 = 0.1$.

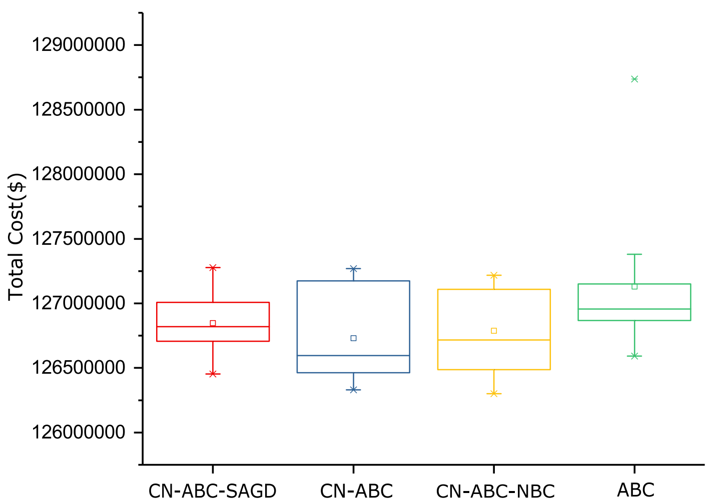


Fig. 9. Box diagram of *TC* solution set obtained when $\omega_1 = 0.5$.

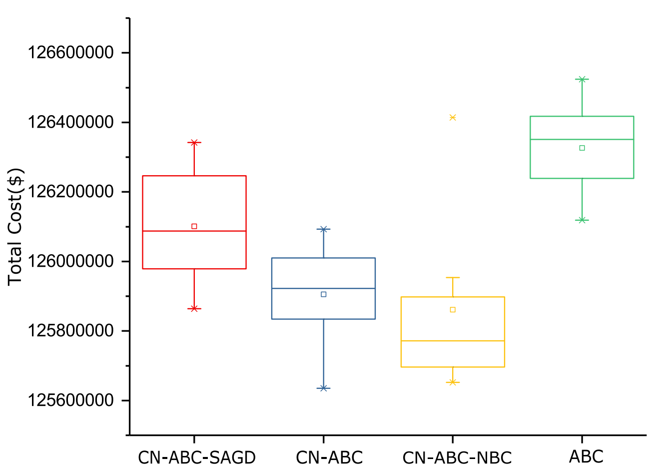


Fig. 10. Box diagram of *TC* solution set obtained when $\omega_1 = 0.9$.

the capability of exploitation for ABC, and it will accelerate the convergence speed of the algorithm. Secondly, the temperature (T) of the simulated annealing (SA) is used to adjust the number of times of the algorithm. When the $T < T\_min$ is terminated, the outer loop is terminated, and the Metropolis criterion in SA can be used to determine whether a new solution is accepted dynamically.

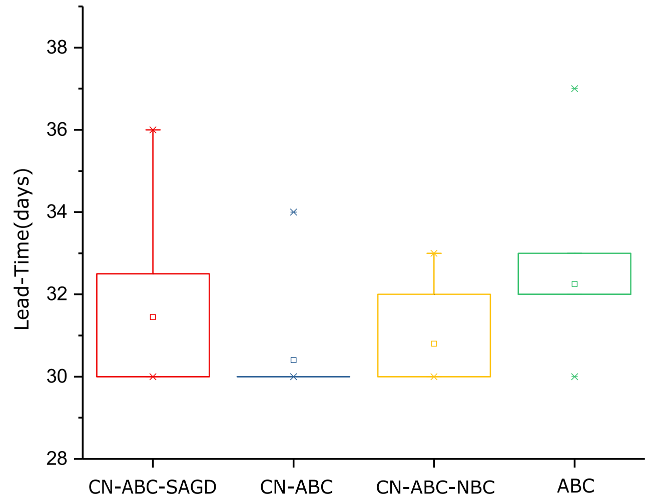


Fig. 11. Box diagram of *LT* solution set obtained when $\omega_1 = 0.1$.

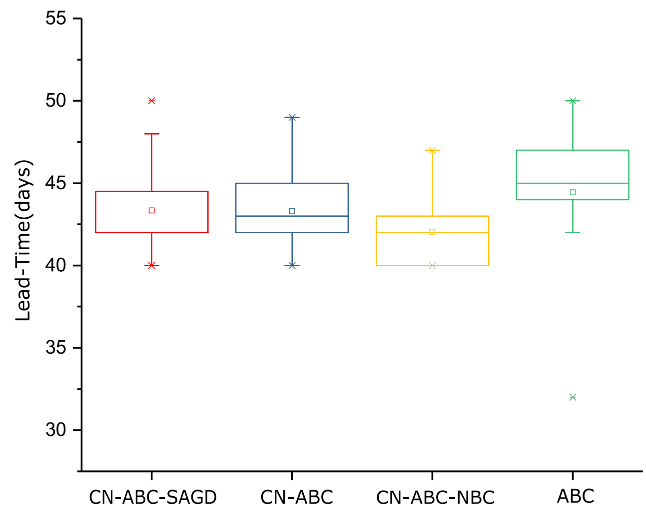


Fig. 12. Box diagram of *LT* solution set obtained when $\omega_1 = 0.5$.

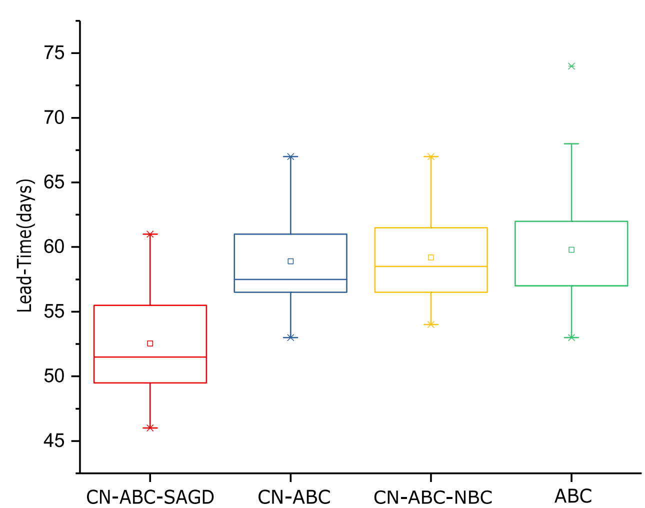


Fig. 13. Box diagram of *LT* solution set obtained when $\omega_1 = 0.9$.

It changes random walk of a scout bee to avoid trapping in a local optimal solution for improving the capability of exploration.

Furthermore, the reference of naive Bayesian probability can maximize the utility of every a randomly obtained solutions, enhance the heuristic performance of the basic ABC algorithm and share the information of the potential direction of the global optimal solution selection in each solution, which will increase the convergence speed. The proposed CN-ABC-SAGD is illustrated as Algorithm 1.

Definition 1. Out degree of node a_i. $OutDegree(a_i)$ equals the number of arrow lines that start from node a_i.

Definition 2. Uni-assembly node. The node a_i belongs to a uni-assembly node when its $OutDegree(a_i) = 1$.

Definition 3. Multi-assembly node. The node a_i belongs to a multi-assembly node when its $OutDegree(a_i) = k$ to support assembling k products.

Definition 4. Final-assembly node. The node a_i belongs to a final-assembly node when its child nodes are all delivering nodes.

Algorithm 1 CN-ABC-SAGD algorithm

Data Collection of all alternative data sets, including their numbers of groups, numbers, lead-time, cost and probability of nodes
Result *GlobalParams*, *GlobalBestCost*, *GlobalBestLeadtime*
Step 1 Extract Sub-Networks from a SCN
Calculate $OutDegree(a_i)$ for each node a_i based on Definition 1
Classify node types based on Definitions 2–4
Extract Sub-Networks based on these three type of nodes as Fig. 2
Form the solution vector of Sub-Network as Fig. 3.
Step 2 Search the global POS using the ABC framework
for $\forall Sub-Network \in SCN$ **do**
  Step 2.1 Send employed bees to find a nectar
  Generate initial solution vector randomly at an average probability
  Evaluate the value of the objective function based on Eq. (1)
  Calculate the fitness value based on $Z(SV_{t+1}) - Z(SV_t)$
  Record to solution vector
  Update Bayesian probability table
  Step 2.2 Send scout bees to search neighbourhoods of the nectar
  for $T < T\_min$ **do**
    Generate new nectars based on Bayesian probability table
    Evaluate the objective function based on Eq. (1)
    Calculate the fitness value based on $Z(SV_{t+1}) - Z(SV_t)$
    Record to solution vector
    Record the minimal solution so far
    if $(Z(SV_{t+1}) - Z(SV_t) < 0 \mid \exp(-(Z(SV_{t+1}) - Z(SV_t))/T) <= rand(0, 1))$ **then**
      Accept the best solution so far and update the nectar
      Update the probability table with Bayesian principle
    end if
  end for
end for

3. Results and analysis

In order to evaluate the performance of the proposed CN-ABC and CN-ABC-SAGD algorithms, a test example is used and we put forward the following situations:

- A total of three products have to be made in the production of Bulldozer SCN, [Wheel Loader (WHL), Track Loader (TRL) and Track-Type Tractor (TTT)]. This SCN covers the whole process from raw materials to manufacturers and distributors.
- Each product can be assembled with a flow pattern that requires the selection of a supplier (or suppliers) for every component used by the product mix.
- Each node has a number of alternative options, out of that one of them represents an single decision that has its own cost and lead-time for assembling this product.
- All the experimental data are under ideal conditions with high stability. There is no other external factors, such as the shortage of raw materials, bad weather for its uncertainty.

The network topology which represents the Bulldozer SCN is depicted in Fig. 1. As shown in Fig. 1, the entire SCN has a total of 38 nodes, with a varying number of 2∼4 blackspots (choices) in each node. Each assembly node can represent an enterprise. Among of these nodes, a_{22}, a_{26} and a_{35} are *final-assembly nodes* for assembling the final product, a_{17} and a_{20} are *multi-assembly nodes*, and most nodes are *uni-assembly nodes*. After extracting sub-networks from Fig. 1, the set of $\{a_1, a_2, a_3, \ldots, a_{16}, a_{17}\}$ can be formed as a sub-network to provide public intermediate products of product *WHL*, *TRL* and *TTT*. Nodes of a_{18}, a_{19} and a_{20} can be formed as another sub network to serve the two products of *TRL* and *TTT*. The set of $\{a_{25}, a_{26}, a_{27}, a_{28}, a_{29}, a_{30}\}$ can be formed as a sub-network for serving only one product of *TTT*. The set of $\{a_{21}, a_{22}, a_{23}, a_{24}\}$ can be aggregated as a sub-network for serving the *TRL* product. The set of $\{a_{31}, a_{32}, a_{33}, a_{34}, a_{35}, a_{36}, a_{37}, a_{38}\}$ can be clustered as a sub network for the *WHL* product.

In Table 1, we list 30 nodes of raw materials or intermediate products. Every node has 2∼5 options, and each option has a different cost C_{ij} (in \$) and required lead-time T_{ij} (in days). Table 2 is the set of target markets for the three final products, and each delivery node has 2 options with parameters of lead-time T_{ij} (in days) and delivery cost C_{ij} (in \$).

In order to measure the global optimization effect of the proposed algorithm, we respectively draw the Pareto front line chart of the ABC algorithm, and the CN-ABC-SAGD algorithm, as shown in Fig. 5. In Fig. 5, we can clearly observe the Pareto frontal value of two different methods in the same data set optimization where the abscissa is the waiting time of the customer (LT), and the ordinate is the total cost of the business (TC). With the increase of LT, TC showed a gradual downward trend. Moreover, it is obvious that the curve of Pareto peak obtained by the CN-ABC-SAGD algorithm is lower than which of the ABC algorithm, which proves that the CN-ABC-SAGD algorithm has a stronger advantage than the basic ABC algorithm in the search for the global optimal solution. Therefore, in the same data set, the CN-ABC-SAGD algorithm has a better global optimum than the ABC algorithm, and has a stronger global search capability (see Fig. 4).

We can see the result based on the proposed approach in Table 3. Each approach takes into account the different weights of cost and lead-time. Each weight corresponds to a different solution. When the weight changes from $\omega_1 = 0.1$ to $\omega_1 = 0.9$, the TC (in \$) values are obtained for 129 020 100, 128 861 700, 128 114 220, 127 123 500, 126 633 900, 126 212 700, 126 450 300, 126 416 100 and 126 635 700, and the LT (in days) values are 31, 33, 40, 42, 44, 46, 54, 47 and 49 respectively.

Mastrocinque et al. (2013) use the basic ABC to solve the same problem [17]. Using the same raw data, when the weight of the TC (in \$) changes from $\omega_1 = 0.1$ to $\omega_1 = 0.9$, TC (in \$) values are obtained from 129 652 620, 129 051 300, 128 485 608, 128 332 908, 127 023 480, 126 821 568, 126 500 100, 126 421 236 and 126 572 640, and the LT (in days) values are 30, 31, 34, 36, 46, 54, 58, 70 and 60 respectively. Fig. 5 is a Pareto fronts line chart for both LT (in days) and TC (in \$) under the same condition with different algorithms. It can be illustrated that most of the results obtained by using our proposed algorithms are better than ABC. Fig. 5(a) is a Pareto fronts line chart for TC (in \$) with the changing weight of TC (in \$). When the weight of the objective

Table 1
Data for solving the Bulldozer SCN.

Node(a_i)	Number(i)	Options(j)	Time(T_{ij})	Cost(C_{ij})
Platform group	1	1	11	575
		2	0	690
		3	5	592
		4	3	630
Wing group	2	1	4	575
		2	9	897
		3	0	912
Roll over group	3	1	7	4 459
		2	3	1 161
		3	0	1 167
		4	3	1 150
Frame	4	1	17	609
		2	10	618
		3	19	605
		4	0	622
Case	5	1	12	2 241
		2	7	2 263
		3	15	2 200
Brake group	6	1	11	575
		2	11	575
		3	11	575
Drive group	7	1	8	1 553
		2	3	1 571
		3	9	1 550
		4	5	1 563
Plant carrier	8	1	9	155
		2	1	157
Frame assembly	9	1	5	620
		2	12	612
		3	19	605
		4	0	622
Transmission	10	1	15	7 450
		2	10	7 618
Drive/brake assembly	11	1	4	1 551
		2	0	1 571
		3	9	1 550
		4	3	1 568
Common assembly	12	1	5	8 000
		2	2	8 070
Engine	13	1	6	4 596
		2	3	4 763
		3	0	4 804
		4	5	4 676
		5	7	4 500
Fans	14	1	12	650
		2	0	662
		3	8	659
Chassis	15	1	7	4 320
		2	2	4 395
Dressed-out engine	16	1	10	4 100
		2	3	4 175
Main assembly	17	1	8	12 000
		2	2	12 150
Track roller frame	18	1	6	3 005
		2	10	3 000
		3	2	3 045
Suspension T	19	1	7	3 600
		2	2	3 675
Subassembly T	20	1	4	8 000
		2	1	8 300
		3	3	8 150
Shovel-T	21	1	35	90
		2	20	95
		3	18	93
Track Loader-TRL	22	1	6	725
		2	2	732
		3	5	730
Blade	25	1	35	90
		2	20	95
		3	18	93
Track Type Tractor-TTT	26	1	6	725
		2	2	732
		3	5	730
Wheels	31	1	6	725
		2	2	732
		3	4	730
Suspensions WHL	32	1	7	3 600
		2	2	3 675
Shovel-W	33	1	35	90
		2	20	95
		3	18	93
Subassembly WHL	34	1	4	8 000
		2	1	8 300
		3	3	8 150
Wheel Loader-WHL	35	1	6	725
		2	2	732
		3	5	730

(*continued on next page*)

Table 2
Data for target markets.

Node(a_i)	Number(i)	Options(j)	Time(T_{ij})	Cost(C_{ij})
R1-TRL	23	1	20	300
		2	10	7000
R2-TRL	24	1	1	500
		2	8	1000
R1-TTT	27	1	10	1200
		2	1	2000
R2-TTT	28	1	20	3000
		2	10	7000
R3-TTT	29	1	15	1500
		2	2	3000
R4-TTT	30	1	1	500
		2	8	0
R1-WHL	36	1	10	1200
		2	1	2000
R2-WHL	37	1	15	1500
		2	2	3000
R3-WHL	38	1	1	500
		2	8	1000

function of *TC* (in $) is increasing, the Pareto fronts have a tendency to decrease. Fig. 5(b) is a Pareto fronts line chart for *LT* (in days) with the varying weight of *TC* (in $). When the weight of the *TC* (in $) objective function is increasing, the weight of *LT* (in days) in the objective function is decreasing, and the Pareto fronts value is gradually increasing. The results show that our proposed CN-ABC-SAGD algorithm has achieved a better feasibility and efficiency.

Figs. 6 and 7, we use CN-ABC-SAGD algorithm and ABC algorithm to obtain the optimal solution of 20 box plot drawn in the same weight conditions, the choices of weights were $\omega_1 = 0.1, \omega_2 = 0.9, \omega_1 = \omega_2 = 0.5$ and $\omega_1 = 0.9, \omega_2 = 0.1$.

The black box in Figs. 6 and 7 shows the solution obtained by using the CN-ABC-SAGD algorithm, and the red box represents the solution obtained based on the basic ABC algorithm. Obviously, whether it is *TC* value or *LT* value, the solution drawn by the CN-ABC-SAGD is lower than that of the solution drawn by the ABC, that is, the majority of the solution values are smaller. The experimental results prove that the proposed CN-ABC-SAGD has obvious advantages than the basic ABC algorithm.

4. Comparative experimental study

Jianhua Jiang, Di Wu et al. put forward ABC algorithm based on complex network and naive Bayes classifier in Fast Multi-objective Pareto Optimization on Supply Chain Network, named as CN-ABC-NBC. We compare the performance of the three proposed algorithms: CN-ABC, CN-ABC-NBC and CN-ABC-SAGD. Because the CN-ABC-SAGD algorithm uses parallel solution in the searching process, it is unreasonable to compare the minimum iteration times with other algorithms. Therefore, we only compare the capabilities

Table 3
The global optimal solution obtained based on the proposed **CN-ABC-SAGD**.

Node(a_i)	z^*									
	Index	Option(j)	Option(j)	Option(j)	Option(j)	Option(j)	Option(j)	Option(j)	Option(j)	Option(j)
Platform group	1	3	1	3	4	4	1	1	4	1
Wing group	2	2	2	3	1	3	3	2	2	3
Roll over group	3	4	4	3	4	4	4	4	4	4
Frame	4	4	4	4	4	4	2	4	1	3
Case	5	2	3	1	3	3	3	3	3	3
Brake group	6	3	1	2	1	1	1	3	3	3
Drive group	7	4	3	4	4	1	3	2	1	3
Plant carrier	8	2	1	2	2	2	2	1	2	2
Frame assembly	9	4	4	4	4	4	4	4	4	4
Transmission	10	2	2	2	2	1	1	2	1	2
Drive/brake assembly	11	1	4	1	4	3	4	1	3	1
Common assembly	12	2	2	2	2	2	2	1	2	2
Engine	13	5	5	5	5	5	5	1	5	5
Fans	14	1	3	2	2	3	3	2	2	2
Chassis	15	1	1	1	1	2	2	2	2	2
Dressed-out engine	16	2	2	2	1	1	1	2	2	2
Main assembly	17	2	2	2	2	2	2	1	2	2
Track roller frame	18	2	2	2	2	2	2	2	2	2
Suspension T	19	1	1	1	1	1	1	1	1	1
Subassembly T	20	2	2	2	2	2	1	1	1	1
Shovel-T	21	3	3	3	3	3	3	3	3	3
Track Loader-TRL	22	2	2	2	2	2	2	2	2	2
Blade	23	2	2	2	1	1	1	1	1	1
Travk Type Tractor-TTT	24	1	1	1	1	1	1	1	1	1
Wheels	25	3	3	3	3	3	3	3	3	3
Suspensions WHL	26	2	2	2	2	2	2	2	2	2
Shovel-W	27	1	1	1	1	1	1	1	1	1
Subassembly WHL	28	2	2	1	1	1	1	1	1	1
Wheel Loader-WHL	29	2	2	1	1	1	1	1	1	1
R1-TRL	30	2	2	2	2	2	2	2	2	2
R2-TRL	31	1	1	1	1	1	1	1	1	1
R1-TTT	32	1	1	1	1	1	1	1	1	1
R2-TTT	33	3	3	2	3	3	3	3	3	3
R3-TTT	34	2	2	1	1	1	1	1	1	1
R4-TTT	35	2	2	2	2	2	2	2	2	2
R1-WHL	36	2	2	2	2	1	1	1	1	1
R2-WHL	37	2	2	2	2	1	1	1	1	1
R3-WHL	38	1	1	1	1	1	1	1	1	1
	ω_1	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
	ω_2	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2	0.1
	TC (in $)	**129 020 100**	**128 861 700**	**128 114 220**	**127 123 500**	**126 633 900**	**126 212 700**	**126 450 300**	**126 416 100**	**126 635 700**
	LT (in days)	**31**	**33**	**40**	**42**	**44**	**46**	**54**	**47**	**49**

of the four algorithms in obtaining the optimal solution. In the comparative study, we select three representative weights of *TC*, that is, $\omega_1 = 0.1$ and $\omega_2 = 0.9$, $\omega_1 = \omega_2 = 0.5$, and $\omega_1 = 0.9$ and $\omega_2 = 0.1$. Each algorithm in the weights were randomly obtained 20 optimal solution, and the results of mapping the box diagram as follows.

When $\omega_1 = 0.1$ in Fig. 8, whether it is the solution set of the maximum and minimum value and average value, using the CN-ABC-SAGD algorithm by *TC* solution is obviously superior to the other algorithms used in the solution, and the solution set is more concentrated in a smaller cost value fluctuates near.

When $\omega_1 = 0.5$ in Fig. 9, the maximum value obtained by the CN-ABC-SAGD algorithm is the same as that of the CN-ABC algorithm, which is higher than the maximum of the solution set obtained by the CN-ABC-NBC algorithm, and is lower than the maximum value of the solution set obtained by the ABC algorithm. Its minimum value is higher than the CN-ABC-NBC and CN-ABC algorithm, but less than the ABC algorithm, the solution set is also more concentrated. The maximum and minimum values of the CN-ABC-NBC algorithm are lower than those obtained by the CN-ABC algorithm, and the solution set is relatively concentrated than the CN-ABC algorithm, but the average value is relatively high. The solution set of ABC algorithm is more concentrated, but the maximum, the minimum and the average value are relatively high, and the solution set is also fluctuating near the larger cost value.

When $\omega_1 = 0.9$ in Fig. 10, the maximum, minimum and mean values of the solution set are significantly higher than those obtained by the CN-ABC-NBC and CN-ABC algorithms using the CN-ABC-SAGD algorithm, and the solution sets are relatively dispersed and the fluctuation is quite large. But the solution set obtained by three kinds of algorithms is lower than that obtained by ABC algorithm. The maximum, minimum and mean values of the solution set obtained by the CN-ABC-NBC algorithm are lower than the maximum of the solution set obtained by the CN-ABC algorithm. The solution set of ABC algorithm is more concentrated, but the maximum, the minimum and the average value are relatively high, and the solution set is also fluctuating near the larger cost value.

When $\omega_1 = 0.1$ in Fig. 11, the maximum, minimum and mean values of the solution set are significantly higher than those obtained by the CN-ABC-NBC and CN-ABC algorithms using the CN-ABC-SAGD algorithm, and the solution sets are relatively dispersed and the fluctuation is quite large. But the solution set obtained by three kinds of algorithms is lower than that obtained by ABC algorithm. The maximum value of the solution set obtained by the CN-ABC-NBC algorithm is significantly higher than that of the CN-ABC algorithm, and is relatively dispersed. The solution obtained by the CN-ABC algorithm is very concentrated.

When $\omega_1 = 0.5$ in Fig. 12, the *LT* solution obtained by using the CN-ABC-SAGD algorithm is lower than that obtained by the CN-ABC algorithm, and the maximum and mean value obtained by

CN-ABC-NBC are minimal. The degree of dispersion of the four algorithms is not significant, but the maximum, minimum and mean values of CN-ABC-SAGD, CN-ABC and CN-ABC-NBC algorithms are lower than those obtained by the ABC algorithm.

When $\omega_1 = 0.9$ in Fig. 13, whether it is the maximum, the minimum and the average value of the solution set, the *LT* solution obtained by using the CN-ABC-SAGD algorithm is obviously better than that obtained by using other algorithms. The solutions obtained by CN-ABC-NBC and CN-ABC algorithms are better than those obtained by the ABC algorithm.

Through the above comparison we cannot find that the proposed CN-ABC-SAGD algorithm to the *TC* value is significantly better than other algorithms, but it is better than the ABC algorithm is slightly worse than the CN-ABC-NBC and CN-ABC algorithm when the weight $\omega_1 = 0.1$. When the weight $\omega_1 = 0.5$, the results obtained by CN-ABC-SAGD are not very different from those of the CN-ABC-NBC and CN-ABC algorithms, but also better than the ABC algorithm. When $\omega_1 = 0.9$, the value of the LT obtained by the CN-ABC-SAGD algorithm is obviously better than other algorithms, but the *TC* value is better than the ABC algorithm but is slightly worse than the CN-ABC-NBC and CN-ABC algorithm. This is because the search range of the solution of the CN-ABC-SAGD algorithm in the solution is greater than that of the other algorithms, when the weight of an index is obviously greater than that of another index, the algorithm has a very good result in solving the weighted index value, and the other index value should be relatively poor. When the weights of two indexes are not big, the algorithm is slightly worse than the CN-ABC-NBC and CN-ABC algorithms. Similarly, when the weight of an index is significantly greater than that of another index, the effect of CN-ABC-NBC and CN-ABC algorithm is slightly worse than that of the CN-ABC-SAGD algorithm. However, regardless of the weight, the CN-ABC-SAGD, CN-ABC-NBC and CN-ABC algorithms are obviously better than the ABC algorithm.

Based on this, when enterprise decision-makers show obvious tendency to some indexes, using CN-ABC-SAGD algorithm can help them to obtain supply chain decision-making scheme quickly and efficiently. When there is no obvious inclination, CN-ABC-NBC and CN-ABC algorithms can help them to obtain the supply chain decision-making scheme quickly and efficiently.

5. Conclusions

In order to solve the problem of complex supply chain decision-making, the ABC algorithm based on gradient descent and simulated annealing is proposed. The optimal management decision of a three-layer Bulldozer supply chain is taken as an example with a multi products and manufacturing centers. The supply chain consists of 38 nodes and 105 optional supplier enterprises for a total possible solutions of 1.284×10^{16}. In the process of solving the cost and lead-time of two indicators as the bi-objective functions, each of the cost of the corresponding supply chain of the manufacturer (or factory) combination has a unique lead-time. In the process of solving the complex network, gradient descent algorithm changes the random direction of the scout bees, simulated annealing speeds up the convergence of the algorithm, and it avoids falling into the local optimal solution. The experimental results show that the proposed CN-ABC-SAGD algorithm is better than the basic ABC and CN-ABC algorithm to find the optimal solution which is closer to the real, and has higher efficiency.

We hope that using the proposed complex network oriented model and meta-heuristic approach will assist logistics managers in making supply chain operation management decision for their complex SCN that considering various issues, such as multiple raw materials, multiple assembling centers, multiple sub-assembling nodes, multiple delivery nodes. Consideration of the interest rate related with the production procedure with the total cost and the lead-time.

For engineers, this paper proposes another way to make optimization for SCN problems. For a typical SCN optimization problem, the solution is not modelled as a simple selection for all nodes, but it can be modelled with its sub-networks derived from the whole SCN. The proposed approach provides another mechanism that the optimization procedure can be computed in parallel. The approach of extracting sub-networks from a SCN is simple and easy to be implemented for engineers. Because of the strength of the proposed CN-ABC-SAGD algorithm, it is helpful when the SCN is real a complex network.

For industry, the topology of SCN becomes more and more complex, and it increases the difficulty of finding acceptable optimal solutions in a limited time. The proposed CN-ABC-SAGD approach provides an interesting idea with a "divide-and-conquer" mechanism, that is an efficient and effective approach to simplify the complexity of a SCN.

The limitations and shortcomings of this paper can be stated as: because of the complexity of SCN, only two objective functions of total cost and lead-time are considered. For the future researches, we recommend following directions: (1) more objective functions can be investigated, (2) other meta-heuristic approaches can be examined as its extensions, (3) more nodes can be applied into the test example that can increase the complexity of the SCN to evaluate its effectiveness.

References

[1] K.C. Tan, A framework of supply chain management literature, Eur. J. Purchasing Supply Manag. 7 (1) (2001) 39–48.
[2] A.M.F. Fard, M. Hajiaghaei-Keshteli, A tri-level location–allocation model for forward/reverse supply chain, Appl. Soft Comput. 62 (2018) 328–346.
[3] M. Goetschalckx, C.J. Vidal, K. Dogan, Modeling and design of global logistics systems: A review of integrated strategic and tactical models and design algorithms, European J. Oper. Res. 143 (1) (2002) 1–18.
[4] A. Gharaei, F. Jolai, A multi-agent approach to the integrated production scheduling and distribution problem in multi-factory supply chain, Appl. Soft Comput. 65 (2018) 577–582.
[5] J.L. Corner, J. Buchanan, Experimental consideration of preference in decision making under certainty, J. Multi-Criteria Decis. Anal. 4 (2) (1995) 107–121.
[6] P. Schutz, A. Tomasgard, S. Ahmed, Supply chain design under uncertainty using sample average approximation and dual decomposition, European J. Oper. Res. 199 (2) (2009) 409–419.
[7] A. Hassanzadeh, M. Rasti-Barzoki, Minimizing total resource consumption and total tardiness penalty in a resource allocation supply chain scheduling and vehicle routing problem, Appl. Soft Comput. 58 (2017) 307–323.
[8] L.A. Moncayo-Martnez, D.Z. Zhang, Multi-objective ant colony optimisation: A meta-heuristic approach to supply chain design, Int. J. Prod. Econ. 131 (1) (2011) 407–420.
[9] A. Kamali, S.M.T. Fatemi Ghomi, F. Jolai, A multi-objective quantity discount and joint optimization model for coordination of a single-buyer multi-vendor supply chain, Comput. Math. Appl. 62 (8) (2011) 3251–3269.
[10] C. Zhang, Y. Yang, Z. Du, C. Ma, Particle swarm optimization algorithm based on ontology model to support cloud computing applications, J. Ambient Intell. Humanized Comput. 7 (5) (2016) 633–638.
[11] R. Li, S. Hu, Y. Wang, M. Yin, A local search algorithm with tabu strategy and perturbation mechanism for generalized vertex cover problem, Neural Comput. Appl. 28 (7) (2017) 1775–1785.
[12] K. Shaw, R. Shankar, S.S. Yadav, L.S. Thakur, Supplier selection using fuzzy AHP and fuzzy multi-objective linear programming for developing low carbon supply chain, Expert Syst. Appl. 39 (9) (2012) 8182–8192.
[13] W.C. Yeh, M.C. Chuang, Using multi-objective genetic algorithm for partner selection in green supply chain problems, Expert Syst. Appl. 38 (4) (2011) 4244–4253.
[14] L.A. Moncayo-Martnez, D.Z. Zhang, Optimising safety stock placement and lead time in an assembly supply chain using bi-objective max–min ant system, Int. J. Prod. Econ. 145 (1) (2013) 18–28.
[15] N. Demirel, E. Ozceylan, T. Paksoy, H. Gokcen, A genetic algorithm approach for optimising a closed-loop supply chain network with crisp and fuzzy objectives, Int. J. Prod. Res. 52 (12) (2014) 3637–3664.
[16] E. Ozceylan, N. Demirel, C. Cetinkaya, et al., A closed-loop supply chain network design for automotive industry in Turkey, Comput. Ind. Eng. (2017) 727–745.
[17] E. Mastrocinque, B. Yuce, A. Lambiase, Packianather M.S., A multi-objective optimization for supply chain network using the bees algorithm, Int. J. Eng. Bus. Manag. 5 (38(6)) (2013) 1–11.

[18] T. Shu, X. Gao, S. Chen, S. Wang, K. Lai, L. Gan, Weighing efficiency-robustness in supply chain disruption by multi-objective firefly algorithm, Sustainability 8 (3) (2016) 250.
[19] D. Karaboga, B. Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, J. Global Optim. 39 (3) (2007) 459–471.
[20] D. Karaboga, B. Gorkemli, C. Ozturk, N. Karaboga, A comprehensive survey: artificial bee colony (ABC) algorithm and applications, Artif. Intell. Rev. 42 (1) (2014) 21–57.
[21] M. Schiezaro, H. Pedrini, Data feature selection based on artificial bee colony algorithm, EURASIP J. Image Video Process. 2013 (1) (2013) 47.
[22] Y. Zhang, L. Wu, S. Wang, Magnetic resonance brain image classification by an improved artificial bee colony algorithm, Prog. Electromag. Res.-pier 116 (2011) 65–79.
[23] Y. Zhang, L. Wu, Optimal multi-level thresholding based on maximum tsallis entropy via an artificial bee colony approach, Entropy 13 (4) (2011) 841–859.
[24] B. Akay, D. Karaboga, A modified artificial bee colony algorithm for real-parameter optimization, Inform. Sci. 192 (6) (2012) 120–142.
[25] A. Banharnsakun, B. Sirinaovakul, T. Achalakul, Job shop scheduling with the best-so-far ABC, Eng. Appl. Artif. Intell. 25 (3) (2012) 583–593.
[26] W. Yang, Z. Pei, Hybrid ABC/PSO to solve travelling salesman problem, Int. J. Comput. Sci. Math. 4 (3) (2013) 214–221.
[27] U. Taetragool, B. Sirinaovakul, T. Achalakul, NeSS: A modified artificial bee colony approach based on nest site selection behavior, Appl. Soft Comput. 71 (2018) 659–671.
[28] Y. Nesterov, Gradient methods for minimizing composite functions, Math. Program. 140 (1) (2013) 125–161.
[29] D. Bertsimas, J. Tsitsiklis, Simulated annealing, Statist. Sci. 8 (1) (1993) 10–15.