

Research Article

Optimizing Product Configuration Problems with Multisourcing Supplier Selections under Both Carbon Cap and Carbon Tax Regulations

Jia Li  and Dong Yang 

School of Business and Management, Donghua University, Shanghai 200051, China

Correspondence should be addressed to Dong Yang; yangdong@dhu.edu.cn

Received 13 July 2021; Revised 3 January 2022; Accepted 15 January 2022; Published 8 February 2022

Academic Editor: Wei Zhang

Copyright © 2022 Jia Li and Dong Yang. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Product configuration is a key enabling technology for mass customization production. Previous studies mainly focus on the solving technologies for obtaining a feasible configuration or optimal configuration but ignore environmental factors like CO₂ emissions, especially the environmental regulations from international organizations. In this article, we address product configuration optimization problems considering both carbon cap and carbon tax regulations. A two-stage approach that combines both Particle Swarm Optimization (PSO) and a greedy algorithm is presented to solve the product configuration problems with carbon emission concerns. A case study of a configurable ranger-drilling machine is illustrated to demonstrate the effectiveness of the presented approach. The impacts of carbon cap and carbon tax regulations on configuration results are compared. Further, the models are extended to analyze the effects of single-sourcing and multisourcing on product configuration. The analysis shows that multisourcing in combination with carbon tax regulation can facilitate a manufacturer to provide more green and low-carbon customized products for customers. Numerical experiments are conducted to validate the effectiveness of the model and the efficiency of the proposed solution method.

1. Introduction

With the fierce market competition, modern enterprises have been striving to improve their product design to meet the personalized needs of customers. Module-based product design has proved to be an effective means to balance production efficiency and individual needs of customers and thus acts as a foundation for implementing mass customization [1–3]. By selecting components from a predefined component catalog with both configuration rules and customer requirements satisfied, a customized product can be configured [4, 5]. Consequently, the product configuration is an enabling technology for mass customization and personalization production [6]. For example, Dell offers customers an online computer configuration system to select options according to their personalized requirements. For the past several decades, product configuration has attracted much attention from both academia and industry. For

representing structure relations between components, various representation methods have been proposed, such as objected-oriented modeling [7] and ontology-based knowledge modeling [8]. Further, substantial achievements have been made in developing the solution methods for obtaining a feasible configuration such that configuration rules owing to both market and technical considerations are observed. Main solution methods for obtaining a valid configuration include logic-based approach [9], rule-based reasoning [10], case-based reasoning [11], and constraint satisfaction problem (CSP) [12, 13]. Moreover, in the case that numerous valid configurations exist for a configuration problem, some researchers apply optimization algorithms to find the optimal configuration in accordance with some objective such as minimal configuration cost. The algorithms contain mathematical programming [14] and metaheuristic methods [15–17], such as the Genetic Algorithm (GA). With the study extending to the supply chain domain, product

configuration decision is closely connected with supply chain network design and thus entails joint optimization of both product configuration decisions and supply chain decisions [18]. However, the aforementioned studies do not take the environmental impact into account during product configuration processes.

With an increasing awareness of global climate change, almost all countries around the world agree to reduce their amount of greenhouse gases such as CO₂. As a consequence, carbon emission regulations or policies are drawn up by international agreements such as the UNFCCC (United Nations Framework Convention on Climate Change) and the Kyoto Protocol [19]. Typical carbon emission regulations include carbon cap, carbon tax, carbon cap and trade, and carbon offset [20]. For carbon cap, a manufacturer is allocated a permitted amount of carbon emissions which restricts its total amount of carbon emissions during production activities. For carbon tax, an extra fee must be paid according to a given tax rate based on how much carbon a manufacturer emits. Under carbon cap-and-trade regulation, the insufficient amount of carbon emissions can be purchased from the carbon-trading market if the needed amount of carbon emissions exceeds the permitted limitation (cap), and the excessive amount can be sold in the market if the total amount of carbon emissions is less than the limitation. Similar to the carbon cap and trade, in the carbon offset regulation, the excessive amount cannot be sold in the carbon market. Although carbon emissions have been considered in inventory control [21], vehicle routing problems [22], port scheduling [23], low-carbon product structure design [24], and product remanufacturing [25], only limited studies considered carbon emissions during the configuration processes. Tang et al. [26] investigated a multiobjective product configuration problem where one of the objectives is to minimize the amount of carbon emissions in a configured product. However, carbon emission regulations such as carbon cap and carbon tax are not handled in their study. In our previous study, the impacts of carbon regulations on product configuration are analyzed [27]. Nevertheless, suppliers for providing modules of a product are not taken into account in the previous research. To take the previous study further, in this article, we address the product configuration problem with multisourcing supplier selections under carbon regulations. The mathematical models under both carbon emission regulations are built considering configuration rules, customer requirements, and carbon emission constraints simultaneously. Our contributions in this article are twofold. On the one hand, a multisourcing strategy for module purchase is proposed in product configuration to avoid the disruption risk of the supply chain. On the other hand, combining the multisourcing strategy and the carbon tax regulation can help a manufacturer reduce its carbon emissions and offer a more green-friendly customized product for customers. This article is organized as follows. Related work is reviewed in Section 2. In Section 3, the product configuration problem under carbon emission regulations is described. The optimization model for the configuration problem is formulated in Section 4. In Section 5, a two-stage solution algorithm

combining PSO and the greedy algorithm is presented to solve the configuration optimization model. A case study of a configurable ranger-drilling machine is illustrated in Section 6. The effects of both carbon cap and carbon tax rate on the configuration results are analyzed. Further analysis is performed by comparing single-source vs. multisourcing strategies. In Section 7, numerical experiments are carried out to validate the effectiveness and efficiencies of the proposed two-stage algorithms. Conclusions and future research are summarized in Section 8.

2. Literature Review

This study is related to both product configuration problems and carbon emissions in productions and operations. Below we first addressed the studies regarding the product configuration problems. Then, the literature about carbon emissions in productions and operations is described.

2.1. Product Configuration. Product configuration research can be dated to the early 1980s, where the configuration system R1 was developed to configure a computer system using rule-based reasoning [10]. Since then, a great deal of effort has been devoted to developing the methods for representing and solving product configuration problems. To obtain a feasible configuration, various problem-solving technologies such as rule-based reasoning [10], case-based reasoning [11], logic-based reasoning [9], and constraint satisfaction problem (CSP) [12, 13] are employed to infer a configuration solution such that both configuration rules and customer requirements are met. For example, Tseng et al. [11] applied the CBR in obtaining a valid configuration by considering the similarity between previous configuration cases and current configuration requirements. Stumptner et al. [12] and Aldanondo and Vareilles [13] employed CSP to infer a valid configuration by transforming configuration concepts such as attributes, modules, and rules into the elements like variables and domains in the CSP. To enable the reuse of the configuration knowledge, on the other hand, object-oriented modeling such as UML (unified modeling language) [7] and ontology-based knowledge modeling [8] is utilized to formally represent the structural relations among modules, components, and products. Furthermore, some researchers explored the problem of product configuration optimization where a considerable number of feasible solutions exist. For instance, Frutos et al. [14] proposed an integer linear programming model for product configuration optimization problem and solved the model using LINGO solver. Hong et al. [15] utilized the Genetic Algorithm to obtain optimal configuration given the AND-OR graph for representing a configurable product. Some recent studies concentrated on eliciting customer preferences by using the KANO approach [28], concurrent product and process configuration [29], and configuring products using online review data [30].

2.2. Carbon Emissions in Production and Manufacturing. The importance of carbon emissions in production and manufacturing has been gradually recognized recently.

Firstly, some researchers focus on how to measure the amount of carbon emissions in a product. Song and Lee [31] presented an approach to estimating the GHG (greenhouse gas) emissions of a product by identifying the bill of material (BOM) of the product [17]. Zhang et al. [24] developed a method of calculating the carbon footprint of a product by analyzing the connection characteristics between components. Further, reducing carbon emissions is handled in the studies related to the manufacturing processes of a product. Yi et al. [32] explored the optimization of process parameters to minimize carbon emissions during the machining processes for low-carbon manufacturing. Kwak and Kim [25] built a green-profit model with a target of environmental-impact savings as a constraint for a line of new and remanufactured products. Additionally, carbon emissions are also dealt with in the logistics of products, like inventory control [21], vehicle routing problems [22], and port scheduling [23].

Only a few studies regarding product family and product configuration considered the environmental impacts. Kwak and Kim [33] developed an optimization model for product family design from an end-of-life perspective and the environmental impacts; i.e., the recovery rate of materials is considered in the model. Tang et al. [26] addressed the product configuration problem with the objective of reducing carbon emissions. However, they do not deal with the carbon emission regulations proposed by UNFCCC and the Kyoto Protocol, among which carbon cap and carbon tax are two types of mainly applied regulations. Yang et al. [27] analyzed the effects of carbon emissions on product configuration decisions and carbon purchasing decisions. However, the supply strategies of modules are not dealt with in the studies of both Tang et al. [26] and Yang et al. [27]. By contrast, our main contribution is to investigate product configuration problems with a multisourcing strategy under carbon regulations. As a result, the risk of the supply chain may be avoided by adopting the strategy.

3. Problem Statement

A manufacturer that adopts assembly-to-order (ATO) production offers customized products for customers with modular product designs. Namely, its product has a modular structure where modules consist of common module, variant module, and optional modules. Common modules that act as a common product platform exist in all product variants, whereas a variant module has several candidate module instances representing differential performances or characteristics of products. Optional modules provide additional functions or features of products for customers. It is assumed that all module instances are purchased from suppliers and only assembling the modules into a product is carried out by the manufacturer under the ATO paradigm. Figure 1 depicts a generic product structure model from which multiple customer orders are configured. As shown in this figure, a configurable product consists of common modules M_c and various modules M_1, M_2, \dots, M_j , which implement different functions of products. For a variant module $M_j (j = 1..J)$, there exist candidate module

instances (i.e., components) $a_{j1}, a_{j2}, \dots, a_{jL}$ representing differential performances or characteristics of products and the candidate module instances are exclusively selected in a configuration, namely, XOR relations. Generally, there exist configure rules restricting possible combinations of module instances due to economic, technical and lawful considerations. Two types of configuration rules are considered in this study. One is the selective rule, which specifies that the selection of a module instance must select another module instance in the same configuration. Another is the incompatible rule, which enforces that two module instances cannot exist in the same configuration since they are incompatible. For example, a selective rule says that the instance a_{21} of module M_2 must be selected if the instance a_{11} of module M_1 is selected, as shown in Figure 1. Moreover, the instance a_{1L_1} of module M_1 and the instance a_{j1} of module M_j cannot be simultaneously selected in one configuration, as indicated by a compatible rule in this figure.

Due to the fact that all module instances (i.e., components) are purchased from suppliers, it is crucial to consider the disruption risk of the supply chain when making decisions on component purchasing for a product configuration [34]. A supply chain can suffer from disruption by unexpected natural or man-made disasters such as fires, earthquakes, label strikes, and terrorist attacks [35]. Multisourcing, which means that the same components can be purchased from several suppliers, has proved to be an effective strategy for reducing the risk of supply disruption [36]. Consequently, the multisourcing strategy is adopted in this study to select module suppliers to protect against the risk of the supply chain.

Furthermore, with an increasing concern for environmental protection, the manufacturer decides to comply with the environmental regulations such as carbon tax and carbon cap put forwards by international agreements like UNCC and the Tokyo protocol. As a result, it is imperative to consider CO₂ emissions when configuring customized products for customers. Therefore, the task of product configuration under environmental regulations is to determine the selection of module instances and the purchased amount of the module instances to minimize the total cost subject to both configuration rules and carbon emission regulations. Moreover, two types of carbon emission regulations, i.e., carbon tax and carbon cap, are considered in this study.

4. Optimization Model

Based on the generic product structure, two mathematical models for optimizing the product configuration problem are constructed by considering carbon cap and carbon tax regulations, respectively. The notations, including sets, parameters, and decision variables, are shown in Table 1. In both models, the decision variables are binary variables x_{ijl} , representing whether l th instance of module j is configured for order i and continuous variables y_{kjl} , denoting the amount of l th instance of module j purchased from supplier k .

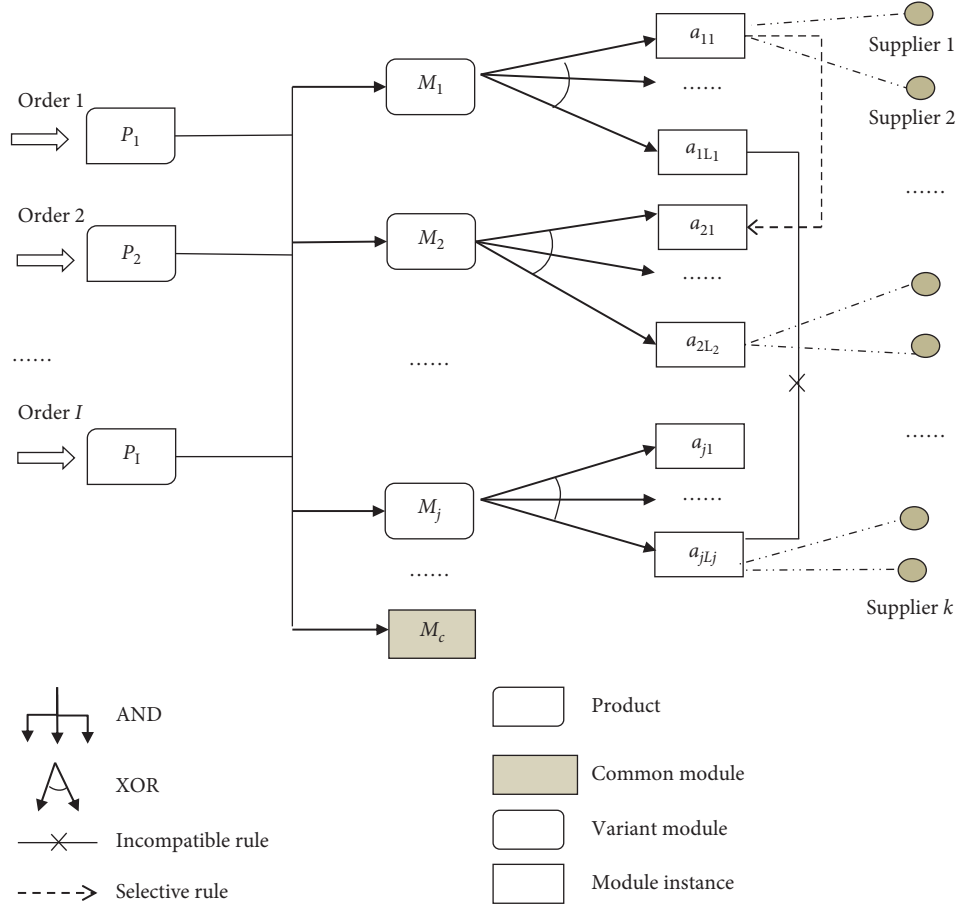


FIGURE 1: A general configurable product structure.

TABLE 1: Notations.

Index and sets	
I	Index of product, $i \in \{1, 2, \dots, I\}$
j, j'	Index of module, $j, j' \in \{1, 2, \dots, J\}$
l, l'	Index of instance, $l, l' \in \{1, 2, \dots, L_j\}$. k index of supplier, $k \in \{1, 2, \dots, K\}$
SEL	Set of selection pair (j_b, j'_l) representing that selection of instance l of module j requires instance l' of module j' in the same configuration
INC	Set of incompatible pair (j_b, j'_l) representing that instance l of module j and instance l' of module j' cannot exist together in the same configuration
$REQI$	Set of pair (i, j, l) representing that a customer needs instance l of module j in product i
$REQN$	Set of pair (i, j, l) representing that a customer does not need instance l of module j in product i
Parameters	
d_i	Annual demand of product i
f_{ijl}	Configuration cost of instance l of module j in product i
u_{ijl}	Number of instances l of module j contained in product i (from bill of material)
h_{kjl}	Per unit purchased cost of instance l of module j from supplier k
e^{cap}	Maximum carbon emission cap
e_{jl}^{ass}	Carbon emissions of instance l of module j during the assembly process
e_{kjl}^{maw}	Carbon emissions of instance l of module j during manufacturing process
α	Carbon tax rate
q_{kjl}	Supply capacity of instance l of module j from supplier k
Decision variables	
x_{ijl}	Binary decision variable, $x_{ijl} = 1$ indicating where instance l of module j exists in a configuration of product i ; otherwise $x_{ijl} = 0$
y_{kjl}	Purchase amount of instance l of module j from supplier k

4.1. Mathematical Model for Carbon Cap Regulation. Under carbon cap regulation, a company is allocated a carbon allowance and the total amount of CO₂ emitted by production activities must be lower or equal to the permitted carbon allowance. If the amount of carbon emissions exceeds the allowance, the production operations of the company will be prevented and correspondingly and the company will be subject to legal punishment. The product configuration problem under the carbon cap can be formulated as a MIP model [PC-CAP].

4.1.1. [PC-CAP].

$$\min \sum_{i=1}^I \sum_{j=1}^J \sum_{l=1}^{L_j} x_{ijl} f_{ijl} u_{ijl} + \sum_{k=1}^K \sum_{j=1}^J \sum_{l=1}^{L_j} y_{kjl} h_{kjl}, \quad (1)$$

s.t.

$$\sum_{l=1}^{L_j} x_{ijl} = 1 \forall i \forall j, \quad (2)$$

$$x_{ijl} \leq x_{i'j'l'} \forall i \forall (j_l, j'_l) \in SEL, \quad (3)$$

$$x_{ijl} + x_{i'j'l'} \leq 1 \forall i \forall (j_l, j'_l) \in INC, \quad (4)$$

$$x_{ijl} = 1 \forall (i, j, l) \in REQI, \quad (5)$$

$$x_{ijl} = 0 \forall (i, j, l) \in REQN, \quad (6)$$

$$\sum_{i=1}^I d_i u_{ijl} x_{ijl} \leq \sum_{k=1}^K y_{kjl} \forall j \forall l, \quad (7)$$

$$y_{kjl} \leq q_{kjl} \forall k \forall j \forall l, \quad (8)$$

$$\sum_{k=1}^K \sum_{j=1}^J \sum_{l=1}^{L_j} (y_{kjl} e_{jl}^{ass} + y_{kjl} e_{kjl}^{raw}) \leq e^{cap}, \quad (9)$$

$$x_{ijl} \in \{0, 1\} \forall i \forall j \forall l, \quad (10)$$

$$y_{kjl} \geq 0 \forall k \forall j \forall l. \quad (11)$$

The objective (1) is to minimize the total cost, including the cost for configuring and assembling modules and the cost for purchasing module instances from suppliers. Constraints (2) ensure that candidate module instances of a module are exclusively selected, namely, XOR structural restriction. Constraints (3) and (4) enforce selective rules and incompatible rules between modules, respectively. Customer requirements are specified on module instances by mapping functional requirements into module requirements since a function of a product may be implemented by one or several modules. The customer requirements for module instances are denoted by constraints (5) and (6), respectively. Constraints (7) impose that the required amount of module instances should be less than or equal to

that amount purchased from suppliers. The supply capacities of suppliers are ensured by constraints (8). Constraint (9) denotes the carbon cap regulation, namely, that the total amount of CO₂ emissions generated in the assembly processes cannot exceed the allocated carbon allowance. Constraints (10) and (11) restrict decision variables.

4.2. Mathematical Model for Carbon Tax Regulation. In addition to carbon cap regulation, the carbon tax is further considered in this study. The carbon tax regulation means that an extra fee must be paid for the amount of CO₂ emission without any restriction on the amount. With the tax rate given as α , the product configuration problem under carbon tax regulation can be formulated as a MIP model [PC-TAX].

4.2.1. [PC-TAX].

$$\min \sum_{i=1}^I \sum_{j=1}^J \sum_{l=1}^{L_j} x_{ijl} f_{ijl} u_{ijl} + \sum_{i=1}^I \sum_{j=1}^J \sum_{l=1}^{L_j} (y_{kjl} h_{kjl}) + \mu \alpha \sum_{k=1}^K \sum_{j=1}^J \sum_{l=1}^{L_j} (y_{kjl} e_{jl}^{ass} + y_{kjl} e_{kjl}^{raw}), \quad (12)$$

s.t.

Constraints (2)–(8).

Constraints (10)–(11).

Objective (12) is to minimize the total cost, including module configuration cost, purchase cost, and carbon tax cost. μ is a penalty factor with the aim of avoiding purchasing too much carbon emissions from the market. The constraints are the same as those in the model [PC-CAP] except that constraints (9) are excluded.

Both models [PC-CAP] and [PC-TAX] belong to combinatorial optimization problems; thus, the models are NP-hard problems. Exact algorithms, such as branch-and-bound only, can solve small- or medium-sized problems and cannot solve large-sized one. Metaheuristic algorithms like Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) are effective approaches to solving combinatorial optimization problems. Therefore, a two-stage algorithm that combines gPSO [37] and the greedy algorithm is employed in this study to solve both models [PC-CAP] and [PC-TAX]. In the first stage, gPSO (“g” stands for “GA”), which incorporates the crossover operator in GA into PSO to solve binary optimization problems, is utilized to select the combination of module instances and thus obtain a valid configuration. In the second stage, the greedy algorithm is designed to obtain the optimal purchase amount of the selected modules from suppliers. The presented two-stage algorithm is elaborated as follows.

5. Solution Algorithm

To solve both the models [PC-CAP] and [PC-TAX] in Section 4, we present a two-stage approach where module selection decisions are solved in the first stage using the

gPSO algorithm, and in the second, the purchase decisions of modules are solved by using the greedy algorithm. In the following, we first address gPSO and then the greedy algorithm is described.

5.1. gPSO. PSO as a swarm-based algorithm mimics the behaviors of bird flocks [38]. In PSO, the problem space is searched by moving a group of particles, representing solutions to the concerned problem. The position and speed of a particle are determined by considering the current position, personal best position (*pbest*), and global best position (*gbest*) simultaneously, as shown in equations (13) and (14). In these equations, v^t and x^t represent the speed and position of a particle in the t th generation, respectively, whereas w , c_1 , c_2 are constant parameters and *rand* is the random number in $[0, 1]$. The processes in which particles are moved through the search space according to the equations are iterated until some stop criteria, such as the maximal number of iterations, are met. Then, the optimal solution corresponding to the best particle with the optimal objective is derived for the interested problem:

$$v^{t+1} = w * v^t + c_1 * rand * (pbest - x^t) + c_2 * rand * (gbest - x^t), \quad (13)$$

$$x^{t+1} = x^t + v^{t+1}. \quad (14)$$

Since (14) is only suitable for handling continuous variables, Kennedy and Eberhart [39] further presented Binary Particle Swarm Optimization (BPSO) algorithm to handle the binary discrete optimization problems. In BPSO, a transformation function, for example, the logistic function, is used to transform a continuous value derived according to (13) into a binary value using the following rule regarding the position update of particles:

$$x^{t+1} = \begin{cases} 1 & \frac{1}{1 + e^{-t}} \geq rand \\ 0 & \text{otherwise} \end{cases}. \quad (15)$$

where *rand* is a random number from a uniform distribution in $[0, 1]$.

However, the indirect transformation makes the selection of the transformation functions problematic since the function is independent of the concrete problem; thus, it may crucially affect the solution quality. By incorporating the crossover operators in GA into PSO, Ozsoydan and Baykasoglu [37] developed gPSO (“*g*” means GA) that takes full advantage of both GA and PSO and makes the direct transformation from a continuous value to a binary value possible. More importantly, one can design the crossover operator, which is problem-related and thus can make full use of domain knowledge. Instead of using equations (13) and (14), gPSO updates the position of a particle using a uniform-based crossover operator in GA. The position of a new particle in gPSO is uniformly determined from the current position, particle best position, and global best

position according to a probability scale, as shown in the following equation:

$$x^{t+1} = \begin{cases} x^t & r < p_1 \\ pbest^t & p_1 \leq r < p_2 \\ gbest^t & r \geq p_2 \end{cases}. \quad (16)$$

Here, $p_1 \in [0, 1]$ and $p_2 \in [0, 1]$ are the parameters set by the decision-maker and $r \in [0, 1]$ is a uniform random number. If the generated random number r is less than p_1 , the new particle takes the value of the current particle. If r is greater than or equal to p_1 and is less than p_2 , the new particle takes the value of the personal best particle (*pbest*). If r is greater than or equal to p_2 , the new particle takes the value of the global best particle (*gbest*).

To avoid premature convergence in PSO, the mutation operation in GA is further adopted by gPSO after the new particles are obtained according to equation (16). Randomly chosen bits in a particle are changed either from 0 to 1 or from 1 to 0 in this mutation operation.

5.2. Particle Representation. In gPSO, each particle corresponds to a candidate solution of the concerned problem. Thus, each particle in this study represents both module selection decisions and purchase decisions using a vector representation. An integer-based approach, not binary encoding, is employed to represent module selection decision in a particle to ensure that only one candidate instance of a module is selected, i.e., XOR structural restriction, whereas the real number is utilized to represent the purchase decisions for the selected modules. Figure 2 shows an illustrative example for a particle. For module selection decisions, there are two products (orders) needed to be configured; i.e., $i = 1, 2$, each of which consists of three modules $j \in [1, 3]$. It is assumed that there exist five candidate module instances for each module; i.e., $l \in [1, 5]$. Therefore, the value of corresponding bits for module selection takes an integer between 1 and 5; i.e., $x_{ijl} \in [1, 5]$. From this figure, it can be seen that for product 1, the first module instance is selected for module 1, the fourth instance selected for module 2, and the fifth instance for module 3. For product 2, the second instance is selected for module 1, the first instance for module 2, and the fourth instance for module 3. For purchase decisions, modules are purchased from two suppliers ($k = 1, 2$). The optimal purchase amounts can be obtained using the greedy algorithm, which will be described later.

5.3. Particle Position Update and Mutation. gPSO in this study updates the position of a particle by incorporating the crossover operator in GA according to equation (16). To avoid premature convergence and trap in local optimum, the mutation is employed in gPSO after a particle has been updated. Figure 3 shows the mutation operation for the particle in Figure 2. As shown in the figure, the fourth instance of module 2 in product 1 is changed to the second instance, namely, that value 4 is replaced by value 2 for this module. For module 1 in product 2, the selected instance is changed from 2 to 5. However, the fragment for purchase decisions in this

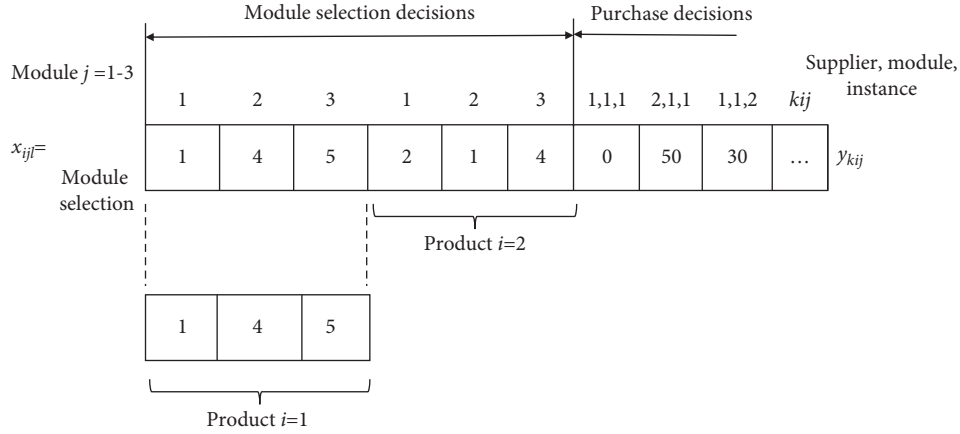


FIGURE 2: Example of a particle.

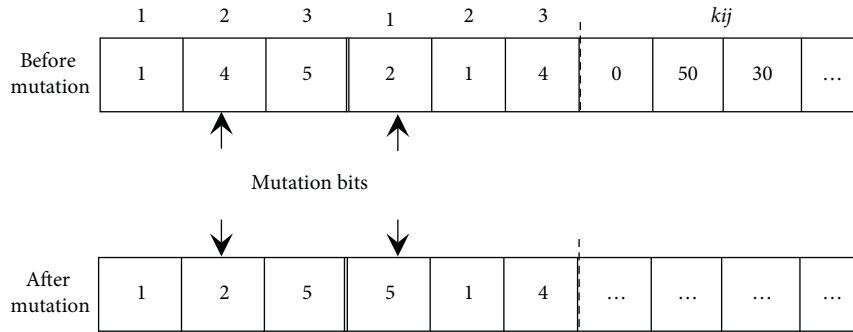


FIGURE 3: Mutation of a particle.

particle is not mutated. The values for purchase decisions are obtained using the developed greedy algorithm.

5.4. Greedy Algorithm for Module Purchase Decisions. The presented two-stage algorithm integrates the gPSO for module selections and the greedy algorithm for purchase decisions. The purchase decisions in the product configuration problem belong to order assignment problems [40]. To speed up the computation time, a greedy algorithm is developed to obtain optimal-near decisions on module purchases. A pseudoutility ratio γ_{kjl} is introduced in this algorithm and is defined as the inverse of multiplication of purchase prices and carbon emissions, i.e., $1/(h_{kjl} \times e_{kjl}^{raw})$. The greater the ration, the higher the chance that a supplier may be selected for offering the module. Consequently, the suppliers that have lower cost or/and lower carbon emissions are preferred to be selected according to the ratio. This greedy algorithm runs as follows.

Step 1. For the selected module instance, its suppliers are sorted in descending order according to the pseudoutility ratio γ_{kjl} .

Step 2. Decode the fragment of a particle representation for module decisions to obtain a configuration solution. Calculate the total amount of the selected module instances according to equation (7), i.e., $\xi_{jl} = \sum_{i=1}^I d_i u_{ijl} x_{ijl}$.

Step 3. For each selected instance, the assignment of purchase orders to suppliers is carried out in accordance with the sorted values mentioned in Step 1. If the amount δ of a module instance that has been assigned is less than the requirement ξ_{jl} , the size of the next order to be assigned is $\min\{q_k, \xi - \delta\}$, where q_k is the capacity for the module instance of supplier k . The process for order assignment repeats until δ is equal to or greater than ξ_{jl} .

Step 4. After all the required module instances have been assigned to suppliers, the values, i.e., sizes of orders, are the encoding values in the fragment of a particle representation for purchase decisions.

5.5. Fitness Evaluation and Constraint Handling. Each particle is assigned a fitness value indicating the survival capability of this particle such that the swarm evolution is guided by both the global best particle and personal best particles. The objective value of the product configuration problem is used to evaluate the fitness value of a feasible particle. However, not all derived particles during the iteration process satisfy the constraints in the models [PC-CAP] and [PC-TAX] due to the mutation operation in the first-stage algorithm, i.e., gPSO. As a result, constraint violation might occur and thus must be handled. There exist two kinds of constraint handling approaches in the metaheuristic algorithm [41]. One is

the repair approach, where the violated particle is transformed to a feasible one according to the repair rules. The other is the penalty approach, where a penalty term is added to the objective function to avoid invalid particles entering into swarm evolution. The penalty approach is employed in this study due to its simplicity and easy implementation. Based on the penalty approach, we design a penalty function for an infeasible particle as follows:

$$G_j(x) = \begin{cases} \max(0, |h_j(x)|) \\ \max(0, l_j(x)) \end{cases}. \quad (17)$$

In the function, $h_j(x)$ represents the left-hand expression of the equality constraint of the form " $h_j(x) = 0$ ". $l_j(x)$ denotes the left-hand expression of the inequality constraints with the form " ≤ 0 ". It is assumed that all constraints in the models [PC-CAP] and [PC-TAX] are transformed into the standard forms of the constraints with " $= 0$ " or " ≤ 0 ." Therefore, the fitness value of a particle is evaluated according to the following equation:

$$\text{fitness} = f(x) + M \sum_j^S G_j(x). \quad (18)$$

Here, $f(x)$ represents the objective value of the model [PC-CAP] or [PC-TAX] in Sections 4.1 and 4.2. M is a penalty coefficient and S is the number of the constraints in this model.

5.6. Two-Stage Solution Algorithm. With the mechanisms for particle update, mutation, and constraint handling, our presented two-stage algorithm combining g-PSO with the greedy algorithm can be summarized in Figure 4. This algorithm begins with the initialization of some parameters such as swarm size ($nPop$), maximal number of iteration (T), and mutation probability. The initial swarm of particles is randomly generated and the personal best particles ($pbest$) and global best particle ($gbest$) in this swarm are derived according to their fitness values. Then, the position of a new particle is determined in accordance with equation (16). The newly generated particles for the module decision fragments are mutated in the given probability. Further, the fragment of purchase decisions in a particle is determined by using the greedy algorithm. If all constraints are not satisfied by the mutated particles, the penalty-based approach will be adopted to evaluate their fitness values according to equation (18). After the fitness values of all particles in the swarm have been evaluated, the personal best particles and global best particles can be obtained in terms of the sorting of the fitness values of particles in the swarm. The processes are repeated until the stop condition is met; namely, the maximal number of iterations is reached. Finally, the algorithm ends with an output of both the optimal configuration of modules and the purchase amount of modules.

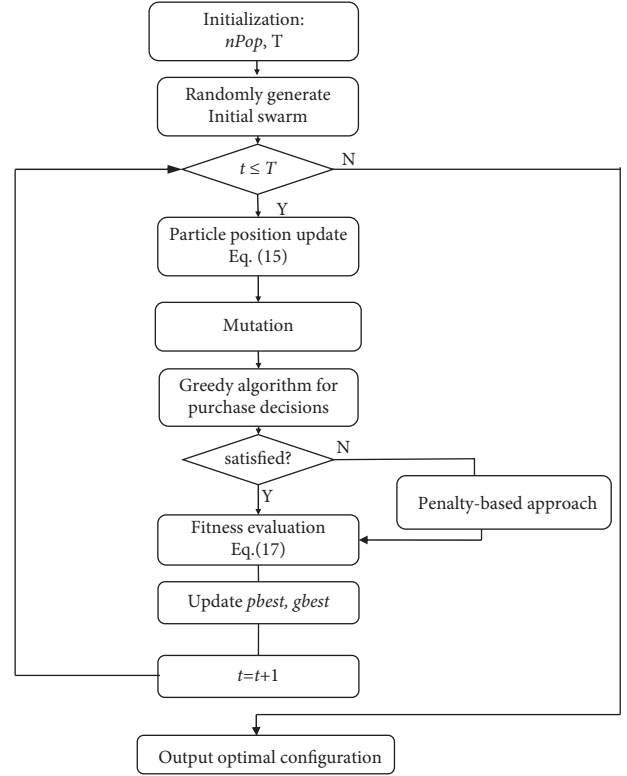


FIGURE 4: The two-stage algorithm for product configuration.

6. Case Study

6.1. Case Description. A manufacturer offers a configurable ranger-drilling machine family with a modular product structure for customers. The structure of this product is depicted in Figure 5 and it consists of two common modules and seven variant modules: extra fuel pump module (M_4), drilling attachment module (M_7), rock driller module (M_6), track module (M_{10}), tank module (M_2), air conditioner module (M_3), and engine module (M_1). Each variant module is composed of several alternative module instances, which have the same or similar functions and differ in the levels of performance or characteristics of products. For example, the extra fuel pump module has two alternative instances: high-capacity pump (HP) and normal-capacity pump (NP). The rock driller module has three types of instances to be selected, i.e., HL500, HL600, and HL700, which differ in the size of drilled holes. The configuration rules restricting the possible combination of module instances are shown in Table 2. The manufacturer adopts the ATO paradigm, which means that all modules are purchased from suppliers and only assembly activities are performed by the manufacturer. The multisourcing strategy is employed in this case to avoid the risk of supply disruption; namely, the module instance can be purchased from multiple suppliers.

Now it is assumed that three customized orders are received and the size of each order is the same, i.e., 100. Table 3 shows the customer requirements for these orders.

TABLE 4: Configuration data about module instances.

Module	Instance	Purchase cost		LD time		Em#1		Em#2	Asm# cost
		S1	S2	S1	S2	S1	S2		
Extra fuel pump	HP	139.6	120.8	11	19	17.4	18	10.5	7.5
	NP	51.2	22.5	6	24	25.2	39		3.2
Drilling attachment	LDA	131.4	45.4	13	14	46.2	51	14.6	8.2
	HDA	144.6	91.4	10	11	28.2	33.6		8.3
Rock driller	HL500	144	110.8	14	15	6.6	21	18	1.3
	HL600	129.6	73.9	14	18	10.8	27		10.3
	HL700	119.3	45.6	19	20	49.2	51.6		10.9
Track module	N	41.3	32.6	11	12	10.8	21.6	16.6	8.3
	T	147.4	87.3	8	18	3.6	3.6		12.3
Tank module	USA	97.5	70.4	7	15	10.8	20.4	13.6	3
	AUS	76.9	29.8	12	24	20.4	40.2		0.6
	EURO	60.2	19.9	9	17	31.8	54		0.7
Air conditioner	AC5	145	107.2	10	22	7.2	12.6	9.9	1.8
	AC6	137.2	93.3	15	19	54.6	59.4		2.7
Engine	S	102	44.1	23	24	55.2	60	12.8	4.1
	W	145.5	89.3	8	16	28.2	32.4		7.1
	R	119	77.9	8	10	40.8	42.6		1.8

Note: (1) "S1" and "S2" represent supplier 1 and supplier 2, respectively. (2) "Em#1" and "Em#2" denote CO₂ amount emitted in manufacturing and assembly activities, respectively. (3) "LD time" and "Asm# cost" mean purchase lead-time and assembly cost, respectively.

per ton of CO₂e are shown in Table 5, respectively. Obviously, due to strict restrictions for carbon emissions, the model [PC-CAP] based on carbon cap regulation has fewer carbon emissions but higher cost, compared with the model [PC-TAX]. However, the opposite situation can occur if we relax the carbon cap (limits) or the carbon tax rate increases.

6.3. Impact of Carbon Emission Regulations

6.3.1. Impact of Carbon Cap. The effects of carbon caps on the model [PC-CAP] are further analyzed using the case of the ranger-drilling machine. Figure 6 shows the effect of different carbon caps on both configuration cost and total carbon emissions. As shown in this figure, total configuration cost increases with a decrease in the carbon cap, namely, when the carbon cap becomes stringent. However, the total amount of CO₂ emissions decreases in the case of more strict carbon caps. It indicates that the adoption of a more strict carbon cap regulation will lead to an increase in production cost for customizing products and a reduction in total CO₂ emissions. Moreover, it tends to select the module instances with low-carbon emissions but high costs when carbon restriction is strict. Nevertheless, when the carbon cap is lower than 76380 kgCO₂-e (i.e., 7.63×10^4), the model becomes infeasible and any feasible configuration cannot be found since any combination of module instances exceeds the permitted carbon cap. On the other hand, when carbon cap is larger than 113500 kgCO₂-e (i.e., 11.35×10^4), both the total configuration cost and total carbon emissions do not change because the carbon restriction is too loose such that any configuration of module combination satisfies the cap. Consequently, setting a reasonable carbon cap for product configuration under the carbon cap model is crucial for decision-makers to make a trade-off between configuration cost and the reduction of CO₂ emissions.

6.3.2. Impact of Carbon Tax. The effect of different carbon tax policies on the model [PC-TAX] is shown in Figure 7. As shown in this figure, the configuration cost increases and total carbon emission reduces when the carbon tax rate increases. This indicates that carbon tax policy has a significant impact on production costs and a higher carbon tax rate can facilitate the reduction of CO₂ emissions. Moreover, an increase in the carbon tax rate will not lead to an infeasible solution, in contrast to the carbon cap regulation. Therefore, the carbon tax is more flexible and acceptable for a manufacturer to manage its customization production.

6.4. Comparisons between Single-Sourcing and Multisourcing Strategies. For the model [PC-CAP] in Section 4, the multisourcing strategy is adopted in the model to combat against the risk of supply disruption, namely, that a module instance can be offered by several suppliers. In the following, we will modify this model to accommodate the single-sourcing strategy, namely, that a module instance is only provided by one supplier. Therefore, the effect of two types of purchase strategies on configuration results can be compared. Based on the model [PC-CAP], a binary decision variable z_{kjl} is introduced to represent whether supplier k is selected to offer the instance l of module j ($=1$, selected; $=0$, not selected). The resulting model considering the single-sourcing strategy is formulated as follows.

$$y_{kjl} \leq Mz_{kjl} \forall k \forall j \forall l, \quad (19)$$

$$\sum_{k=1}^K z_{kjl} \leq 1 \forall j \forall l, \quad (20)$$

$$z_{kjl} \in \{0, 1\} \forall k \forall j \forall l. \quad (21)$$

TABLE 5: Configuration results.

Module	Instance	Configuration (carbon cap)					Configuration (carbon tax)				
		Module selection			Purchase amount		Module selection			Purchase amount	
		P1	P2	P3	S1	S2	P1	P2	P3	S1	S2
Extra fuel pump	HP	+			0	100	+			0	100
	NP		+	+	51	149		+	+	50	150
Drill attachment	LDA		+		0	100		+	+	50	150
	HDA	+		+	50	150	+			0	100
Rock drill	HL500			+	0	100			+	0	100
	HL600	+			0	100	+			0	100
	HL700		+		0	100		+		0	100
Track module	N	+		+	200	200	+		+	200	200
	T		+		0	200		+		0	200
Tank	USA		+		0	100		+		0	100
	EURO			+	0	100			+	0	100
	AUS	+			61	39	+			0	100
Air conditioner	AC5	+	+	+	150	150	+	+	+	150	150
	AC6				0	0				0	0
Engine	S				0	0		+		0	100
	W	+		+	50	150	+		+	50	150
	R		+		0	100				0	0
Total configuration cost (\$)					179696.7					170871.9	
Carbon emission amount (kgCO ₂ -e)					94992					99870	

Note: (1) “P1,” “P2,” and “P3” represent the configured products for order 1, order 2, and order 3, respectively. (2) “S1” and “S2” represent supplier 1 and supplier 2, respectively. (3) The symbol “+” denotes the selection of a module instance in a configuration.

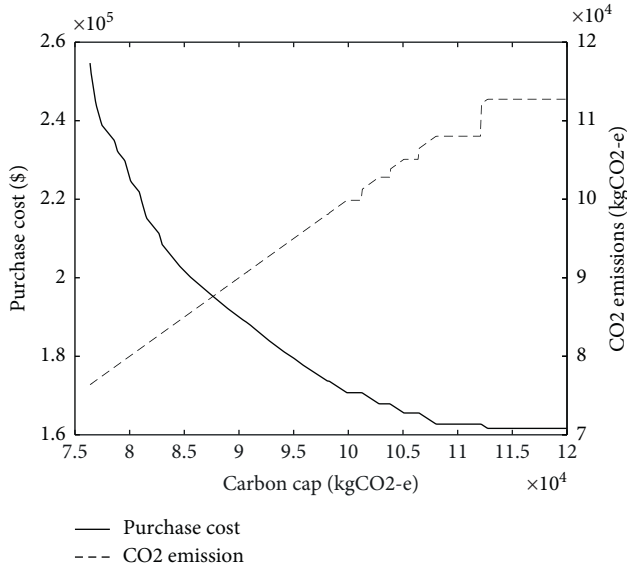


FIGURE 6: Purchase cost and CO₂ emissions with respect to carbon cap.

Constraints (19) enforce that a module instance can only be purchased from a supplier when the supplier is selected. Constraints (20) guarantee the single-sourcing strategy, namely, that a module instance can only be purchased from one supplier. Constraints (21) restrict the decision variables z_{kjl} .

The configuration results based on single-sourcing and multisourcing models under the carbon cap regulation are compared in terms of module selection and purchase

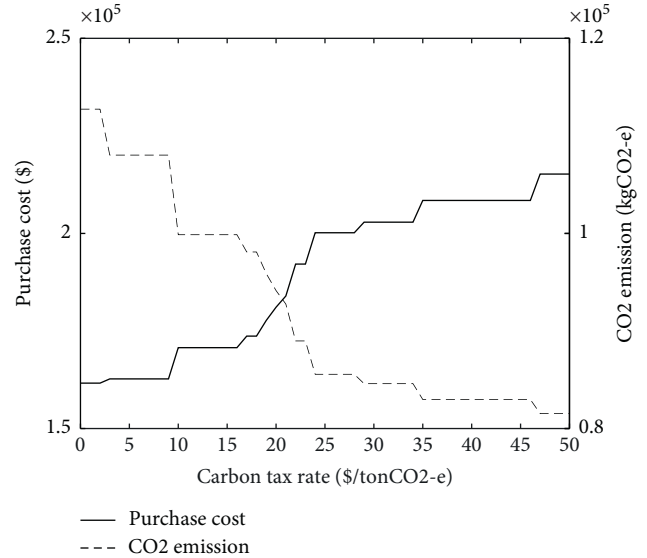


FIGURE 7: Purchase cost and CO₂ emissions with respect to the carbon tax rate.

amount, as shown in Figure 8. As shown in Figure 8(a), a module instance is supplied by only one supplier if it is selected. For example, 300 units of the module instance *LDA* are purchased from supplier *S2*. By comparison, for the multisourcing model, 50 and 150 units of the instance *HDA* are purchased from suppliers *S1* and *S2*, respectively, as shown in Figure 8(b). The reason is that the single-sourcing model prefers the supplier offering the

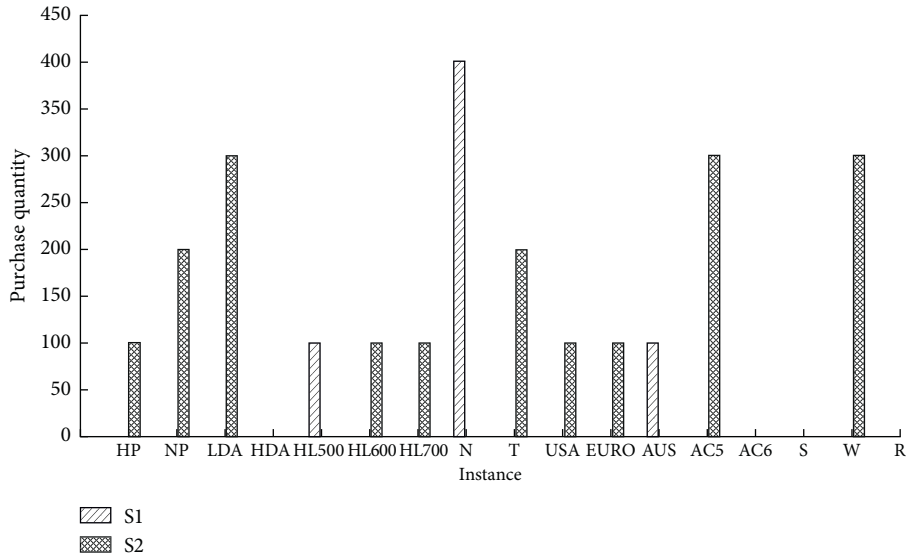
Minimize : Equation (1)

s.t.

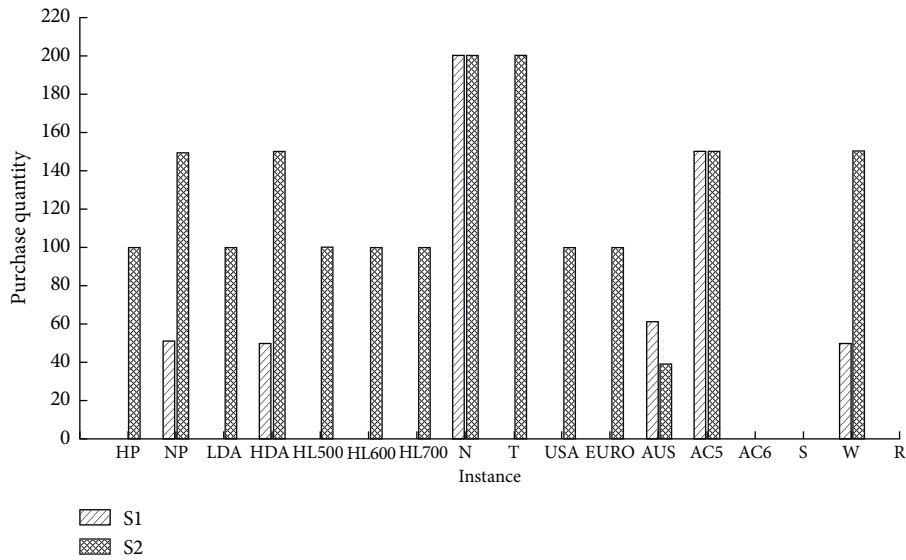
Equations (2)–(7)

Equations (9)–(11)

ALGORITHM 1: [PC-CAP-S]



(a)



(b)

FIGURE 8: Configuration results under both purchasing strategies: (a) purchase amount based on single-sourcing; (b) purchase amount based on multisourcing.

instances with low costs such as LDA, whereas under multisourcing strategy, a part of the purchase amount may be provided by the suppliers with a relatively high cost, such as HDA, due to the restriction of purchase amount from each supplier.

Further, we compare the carbon emissions of configuration results for both strategies under the carbon cap regulation, as shown in Figure 9(a). The total carbon emission amount is 95000 kgCO₂-e for three orders under single-sourcing, while the total carbon emission amount

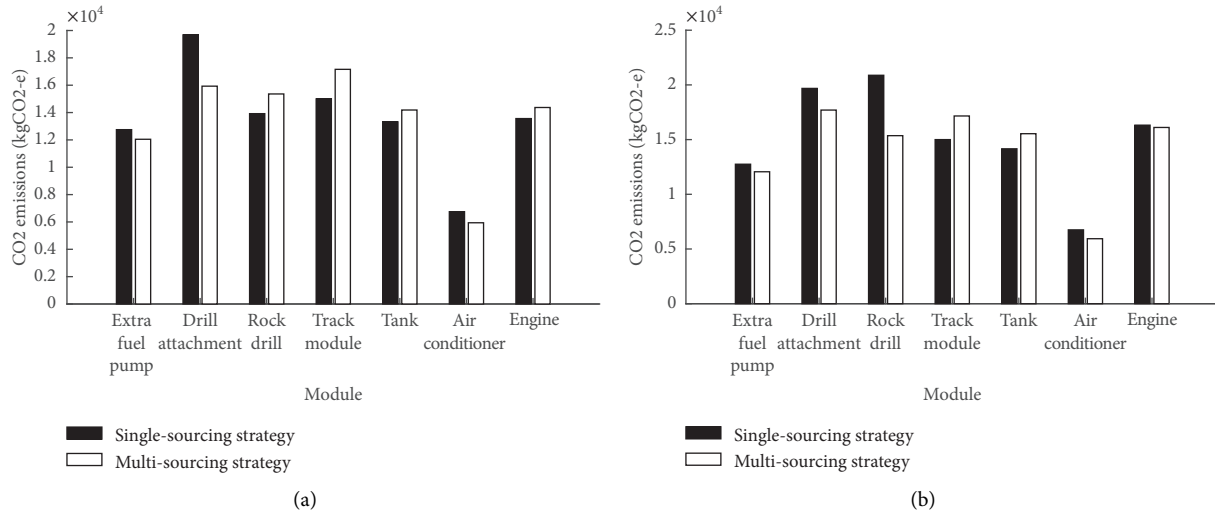


FIGURE 9: Carbon emissions for single-sourcing and multisourcing strategies under carbon regulations: (a) carbon cap; (b) carbon tax.

under multisourcing is 94992 kgCO₂-e. Clearly, the total carbon emissions under the two purchase strategies are almost equal. The reason is that under the carbon cap regulation, a manufacturer will tend to select the models with lower costs as long as the total emissions do not exceed the carbon limitation, considering that the optimization objective is to minimize the total cost.

On the other hand, we also investigate the effect of carbon emissions for both strategies under the carbon tax regulation, as shown in Figure 9(b). When the tax rate is 10\$/tonCO₂-e, the total carbon emission amount under single-sourcing is 105540 kgCO₂-e, while the total carbon emission amount under multi-sourcing is 99870 kgCO₂-e. Obviously, the configuration under multi-sourcing has much less carbon emissions than that under single-sourcing. The reason is that under single-sourcing a manufacturer rather purchases the components from only one supplier with lower cost, but possibly with high carbon emissions. In contrast, under multisourcing, a manufacturer tends to purchase components from several suppliers, possibly with high average cost and shorter average lead-time. Therefore, multisourcing under carbon tax will be beneficial for mass customization enterprises to provide green and low-carbon customized products.

7. Numerical Experiments

To validate the efficiency of the presented two-stage algorithm, numerical experiments are conducted to compare it with the gPSO [37] and classical GA [42]. All experiments are programmed in Matlab2016a running on a computer with Intel(R) Core(TM) i3-6300 processor and 4G memory. The problem instances are generated randomly with case ID I-J-L-K, where I , J , L , and K represent the number of

orders, modules, module instances, and suppliers, respectively. We assume that all modules have the same number of module instances. The purchase costs of modules, carbon emissions in modules production, and carbon emissions in module assembly are taken values from the uniform distributions $U[50, 500]$, $U[20, 70]$, and $U[20, 50]$, respectively. Each problem instance runs 20 times and the best average values of the objective are obtained based on the results of 20 runs. The results are shown in Table 6. The column “two-stage algorithm” represents the presented algorithm in this study. The column “classical GA” refers to the approach by Meena and Sarmah [42]. The next column labeled “gPSO” refers to the algorithm by Ozsoydan and Baykasoglu [37]. The best and average objection values (Avg.) are reported and the best average values for these three algorithms are highlighted. The computation times are given in CPU seconds. It clearly indicates that the two-stage algorithm outperforms classical GA and gPSO in both qualities of solution and computation times, especially in the case of large-scale problem instances. In addition, the GA is superior to gPSO in that the gPSO is only effective in handling discrete variables and is incapable of dealing with continuous variables. By comparison, the presented two-stage algorithm combines the gPSO for handling discrete variables such as module selections and the greedy algorithm for continuous variables like the amount of module purchase. In addition, Figure 10 illustrates the evolution of average fitness values with the number of iterations for the case instance 10-30-10-15. It can be seen that the two-stage algorithm converges much faster than both GA and gPSO. Therefore, the two-stage algorithm is effective in solving larger-scale product configuration problems with supplier selections and environment consideration.

TABLE 6: Computational results.

Case ID	Two-stage algorithm			Classical GA			gPSO		
	Best (e06)	Avg (e06)	CPU time	Best (e06)	Avg (e06)	CPU time	Best (e06)	Avg (e06)	CPU time
I-J-L-K									
5-15-3-3	8.72	8.74	0.54	8.63	8.65	1.12	8.80	8.80	0.78
5-15-5-5	6.69	6.69	1.71	6.68	6.69	2.06	6.83	6.85	1.90
5-15-10-10	5.71	5.71	5.07	5.87	5.90	3.73	5.92	5.94	4.55
10-30-3-5	27.45	27.53	2.73	26.99	27.04	3.32	28.13	28.16	1.84
10-30-5-10	26.13	26.15	7.93	26.37	26.37	5.77	26.84	26.89	4.83
10-30-10-15	27.76	27.78	11.39	28.62	28.64	14.50	28.77	28.88	33.79
20-50-3-10	101.30	101.41	10.59	101.48	101.48	10.20	102.11	102.33	5.85
20-50-5-15	91.58	91.66	25.87	92.61	92.73	55.02	93.36	93.46	25.30
20-50-10-20	85.83	85.91	173.87	87.95	88.04	209.51	88.40	88.54	104.41
30-70-3-15	223.29	223.41	22.46	223.81	224.05	51.40	226.34	226.51	18.38
30-70-5-20	203.02	203.14	112.18	205.10	205.24	128.29	210.37	210.84	58.53
30-70-10-25	196.70	196.78	221.72	199.59	199.91	303.42	200.29	200.37	233.58
50-100-3-20	482.37	476.46	82.99	485.09	485.44	119.59	490.74	491.31	47.76
50-100-5-25	457.46	468.60	185.27	490.48	490.86	235.10	501.15	502.79	139.37
50-100-10-30	484.12	487.45	478.76	500.39	500.55	601.37	521.55	523.76	529.79

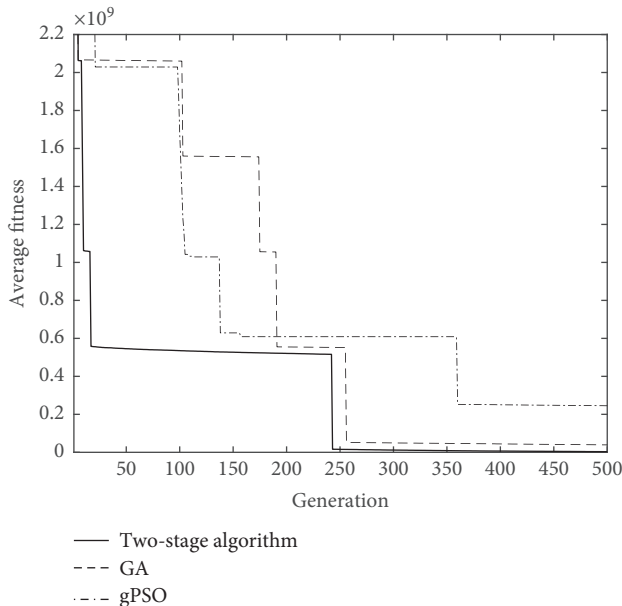


FIGURE 10: Converge behavior of a problem instance.

8. Conclusions

In this article, the product configuration problems with the multisourcing strategy under carbon emissions regulations are investigated. Mathematical models are built for both carbon cap and carbon tax, and a two-stage algorithm is presented to solve the models. Sensitivity analysis of the carbon parameters indicates that both a strict carbon emission cap and high carbon tax rate will enable a manufacturer to purchase the components with higher costs and lower carbon emissions. However, the carbon tax is much more flexible for a manufacturer to manage its customization production. Moreover, in comparison with single-sourcing, multisourcing under carbon tax regulation can obtain configured products with lower carbon emissions and ensure providing green and low-carbon customized products for customers. Under carbon cap, nevertheless, both

single-sourcing and multisourcing strategies make no substantial effects on configuration results.

There are a number of research directions that can be considered useful extensions of this research. Although carbon cap and carbon tax regulations are explored in this study, other carbon regulations such as carbon cap-and-trade and carbon offset strategies should also be investigated. Further, taking the uncertainty in purchase costs into account should be an interesting subject. [43].

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

Acknowledgments

This project was funded by Shanghai Philosophy and Social Science Foundation under Grant no. 2018BGL002.

References

- [1] J. Jiao and M. M. Tseng, "A methodology of developing product family architecture for mass customization," *Journal of Intelligent Manufacturing*, vol. 10, no. 1, pp. 3–20, 1999.
- [2] J. Jiao, T. W. Simpson, and Z. Siddique, "Product family design and platform-based product development: A state-of-the-art review," *Journal of Intelligent Manufacturing*, vol. 18, no. 1, pp. 5–29, 2007.
- [3] L. Gauss, D. P. Lacerda, and P. A. Cauchick Miguel, "Module-based product family design: Systematic literature review and meta-synthesis," *Journal of Intelligent Manufacturing*, vol. 32, pp. 265–312, 2020.
- [4] D. Sabin and R. Weigel, "Product configuration frameworks-a survey," *IEEE Intelligent Systems*, vol. 13, no. 4, pp. 42–49, 1998.

- [5] L. L. Zhang, "Product configuration: A review of the state-of-the-art and future research," *International Journal of Production Research*, vol. 52, no. 21, pp. 6381–6398, 2014.
- [6] T. W. Simpson, "Product platform design and customization: Status and promise," *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, vol. 18, no. 1, pp. 3–20, 2004.
- [7] A. Felfernig, G. E. Friedrich, and D. Jannach, "UML as domain specific language for the construction of knowledge-based configuration systems," *International Journal of Software Engineering and Knowledge Engineering*, vol. 10, no. 4, pp. 449–469, 2000.
- [8] T. Soinen, J. Tiihonen, T. Männistö, and R. Sulonen, "Towards a general ontology of configuration," *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, vol. 12, no. 4, pp. 357–372, 1998.
- [9] J. Tiihonen, M. Heiskala, A. Anderson, and T. Soinen, "WeCoTin - a practical logic-based sales configurator," *AI Communications*, vol. 26, no. 1, pp. 99–131, 2013.
- [10] J. McDermott, "R1: A rule-based configurator of computer systems," *Artificial Intelligence*, vol. 19, no. 1, pp. 29–88, 1982.
- [11] H.-E. Tseng, C.-C. Chang, and S.-H. Chang, "Applying case-based reasoning for product configuration in mass customization environments," *Expert Systems with Applications*, vol. 29, no. 4, pp. 913–925, 2005.
- [12] M. Stumptner, G. E. Friedrich, and A. Haselböck, "Generative constraint-based configuration of large technical systems," *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, vol. 12, no. 4, pp. 307–320, 1998.
- [13] M. Aldanondo and E. Vareilles, "Configuration for mass customization: How to extend product configuration towards requirements and process configuration," *Journal of Intelligent Manufacturing*, vol. 19, no. 5, pp. 521–535, 2008.
- [14] J. D. Frutos, E. R. Santos, and D. Borenstein, "Decision support system for product configuration in mass customization environments," *Concurrent Engineering: Research and Applications*, vol. 12, no. 2, pp. 471–478, 2004.
- [15] G. Hong, L. Hu, D. Xue, Y. L. Tu, and Y. L. Xiong, "Identification of the optimal product configuration and parameters based on individual customer requirements on performance and costs in one-of-a-kind production," *International Journal of Production Research*, vol. 46, no. 12, pp. 1–30, 2008.
- [16] C. Zhou, Z. Lin, and C. Liu, "Customer-driven product configuration optimization for assemble-to-order manufacturing enterprises," *International Journal of Advanced Manufacturing Technology*, vol. 38, no. 1-2, pp. 185–194, 2008.
- [17] B. Li, L. Chen, Z. Huang, and Y. Zhong, "Product configuration optimization using a multiobjective genetic algorithm," *International Journal of Advanced Manufacturing Technology*, vol. 30, no. 1-2, pp. 20–29, 2006.
- [18] D. Yang, J. Jiao, Y. Ji, G. Du, P. Helo, and A. Valente, "Joint optimization for coordinated configuration of product families and supply chains by a leader-follower Stackelberg game," *European Journal of Operational Research*, vol. 246, no. 1, pp. 263–280, 2015.
- [19] Y.-M. Wei, Z.-F. Mi, and Z. Huang, "Climate policy modeling: An online SCI-E and SSCI based literature review," *Omega*, vol. 57, no. SI, pp. 70–84, 2015.
- [20] C. F. Lee, S. J. Lin, and C. Lewis, "Analysis of the impacts of combining carbon taxation and emission trading on different industry sectors," *Energy Policy*, vol. 36, no. 2, pp. 722–729, 2008.
- [21] D. Konur, "Carbon constrained integrated inventory control and truckload transportation with heterogeneous freight trucks," *International Journal of Production Economics*, vol. 153, pp. 268–279, 2014.
- [22] S. Wang, F. Tao, Y. Shi, and H. Wen, "Optimization of vehicle routing problem with time windows for cold chain logistics based on carbon tax," *Sustainability*, vol. 9, no. 5, p. 694, 2017.
- [23] T. Wang, X. Wang, and Q. Meng, "Joint berth allocation and quay crane assignment under different carbon taxation policies," *Transportation Research Part B: Methodological*, vol. 117, no. A, pp. 18–36, 2018.
- [24] X. F. Zhang, S. Y. Zhang, Z. Y. Hu, G. Yu, C. H. Pei, and R. N. Sa, "Identification of connection units with high GHG emissions for low-carbon product structure design," *Journal of Cleaner Production*, vol. 27, pp. 118–125, 2012.
- [25] M. Kwak and H. Kim, "Green profit maximization through integrated pricing and production planning for a line of new and remanufactured products," *Journal of Cleaner Production*, vol. 142, no. 4, pp. 3454–3470, 2017.
- [26] D. Tang, Q. Wang, and I. Ullah, "Optimisation of product configuration in consideration of customer satisfaction and low carbon," *International Journal of Production Research*, vol. 55, no. 12, pp. 3349–3373, 2017.
- [27] D. Yang, J. Li, B. Wang, and Y.-j. Jia, "Module-based product configuration decisions considering both economical and carbon emission-related environmental factors," *Sustainability*, vol. 12, no. 3, p. 1145, 2020.
- [28] C.-H. Wang, "Incorporating customer satisfaction into the decision-making process of product configuration: A fuzzy Kano perspective," *International Journal of Production Research*, vol. 51, no. 22, pp. 6651–6662, 2013.
- [29] P. Pitiot, L. G. Monge, M. Aldanondo, E. Vareilles, and P. Gaborit, "Optimisation of the concurrent product and process configuration: An approach to reduce computation time with an experimental evaluation," *International Journal of Production Research*, vol. 58, no. 2, pp. 631–647, 2020.
- [30] Y. Jiao and Y. Yang, "A product configuration approach based on online data," *Journal of Intelligent Manufacturing*, vol. 30, no. 6, pp. 2473–2487, 2019.
- [31] J.-S. Song and K.-M. Lee, "Development of a low-carbon product design system based on embedded GHG emissions," *Resources, Conservation and Recycling*, vol. 54, no. 9, pp. 547–556, 2010.
- [32] Q. Yi, C. Li, Y. Tang, and X. Chen, "Multi-objective parameter optimization of CNC machining for low carbon manufacturing," *Journal of Cleaner Production*, vol. 95, pp. 256–264, 2015.
- [33] M. Kwak and H. M. Kim, "Assessing product family design from an end-of-life perspective," *Engineering Optimization*, vol. 43, no. 3, pp. 233–255, 2011.
- [34] P. L. Meena, S. P. Sarmah, and A. Sarkar, "Sourcing decisions under risks of catastrophic event disruptions," *Transportation Research Part E: Logistics and Transportation Review*, vol. 47, no. 6, pp. 1058–1074, 2011.
- [35] N. Costantino and R. Pellegrino, "Choosing between single and multiple sourcing based on supplier default risk: A real options approach," *Journal of Purchasing and Supply Management*, vol. 16, no. 1, pp. 27–40, 2010.
- [36] T. Sawik, "Optimization of cost and service level in the presence of supply chain disruption risks: single vs. multiple sourcing," *Computers & Operations Research*, vol. 51, pp. 11–20, 2014.

- [37] F. B. Ozsoydan and A. Baykasoglu, "A swarm intelligence-based algorithm for the set-union knapsack problem," *Future Generation Computer Systems*, vol. 93, pp. 560–569, 2019.
- [38] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *Proceedings of the International Conference on Neural Networks*, pp. 1942–1948, IEEE, Perth, December 1995.
- [39] J. Kennedy and R. C. Eberhart, "A discrete binary version of the particle swarm algorithm," in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, pp. 4104–4109, Orlando, 1997.
- [40] L. X. Cui, "Joint optimization of production planning and supplier selection incorporating customer flexibility: an improved genetic approach," *Journal of Intelligent Manufacturing*, vol. 27, no. 5, pp. 1017–1035, 2014.
- [41] P. C. Chu and J. E. Beasley, "A genetic algorithm for the multidimensional knapsack problem," *Journal of Heuristics*, vol. 4, no. 1, pp. 63–86, 1998.
- [42] P. L. Meena and S. P. Sarmah, "Multiple sourcing under supplier failure risk and quantity discount: a genetic algorithm approach," *Transportation Research Part E: Logistics and Transportation Review*, vol. 50, pp. 84–97, 2013.
- [43] P. Henderson and M. Kernahan, "Modelling and solving engineering product configuration problems by constraint satisfaction," *International Journal of Production Research*, vol. 43, no. 20, pp. 4455–4469, 2005.