

OPTIMIZATION OF PRODUCT LINE CONSIDERING COMPATIBILITY AND RELIABILITY VIA DISCRETE IMPERIALIST COMPETITIVE ALGORITHM

CHUNFENG LIU¹, XIAO YANG¹ AND JUFENG WANG^{2,*}

Abstract. In the era of mass customization, designing optimal products is one of the most critical decision-making for a company to stay competitive. More and more customers like customized products, which will bring challenges to the product line design and the production. If a company adopts customers' favorite levels, this may lead to lower product reliability, or incompatibility among the components that make up the product. Moreover, it is worth outsourcing certain attribute levels to reduce production cost, but customers may dislike these levels because of their delivery delay. If managers consider the compatibility issue, the quality issue, outsource determination, and the delivery due date in the product design and production stages, they will avoid unreasonable product configuration and many unnecessary expenses, thereby bringing benefits to the company. To solve this complicated problem, we establish a nonlinear program that maximizes Per-capita-contribution Margin considering Reliability Penalty. Since the integrated product line design and production problem is NP-hard, we propose an improved Discrete Imperialist Competitive Algorithm (DICA). The proposed DICA is compared with genetic algorithm (GA) and simulated annealing (SA) through extensive numerical experiment, and the results show that DICA displays 6%~17% and 5%~14% improvement over GA and SA in terms of solution quality, respectively.

Mathematics Subject Classification. 90B30, 68W50, 90B50.

Received January 12, 2021. Accepted November 21, 2021.

1. INTRODUCTION

Nowadays, the economic environment in which firms operate is more competitive than ever. The diversification of customers' demand is a significant characteristic of the market, and many manufacturing industries have paid great attention to this phenomenon. The diversified and personalized demand drives enterprises to use a few competitive strategies and computational intelligence for smart manufacturing, such as mass customization (MC) [41, 44], to increase their benefits and customers' satisfaction.

The MC concept has emerged as a major manufacturing strategy. It focuses on changes in demand and technology [12]. Vinodh *et al.* [55] suggested that a focus on customer's requirement is the universal feature of MC and manufacturing agility. In the process of mass customization, the paradigm of market-based product design explores the integration of quantitative customer preference data and product design optimization problems.

Keywords. Mass customization, product line optimization, imperialist competitive algorithm, compatibility, product reliability.

¹ School of Management, Hangzhou Dianzi University, Hangzhou 310018, P.R. China.

² Department of Mathematics, China Jiliang University, Hangzhou 310018, P.R. China.

*Corresponding author: wang_jufeng@163.com

Tsafarakis *et al.* [51] pointed out that product design is a complex and not well formalized discipline, which combines the fields of marketing and engineering.

To meet the needs of different customers, a company must continuously introduce new products or redesign existing ones. However, such procedures are risky and expensive to implement. To minimize the associated risks, managers assess the potential success of a new product concept at an early design stage. For example, Abedi [1] designed a model that helps managers make decisions early in the product life. These optimal product (line) design problems have constituted a wide area of research in quantitative marketing [51]. A product line is defined as a set of product variants (products for short) offered by a single company with similar structures and functions but with different configurations. The product line needs to be optimized under several specific goals, such as market share and profit. Each product is represented as a set of attributes with several specific levels. A wardrobe for example, consists of the attributes of board, color, and hardware accessory, which take the levels of solid wood particle board or ecological board, pure color or combination color, and metal, alloy or stainless steel, respectively. Customers can choose the levels of attributes according to their preferences.

The large number of attributes and levels increases the number of product configurations. It complicates the selection of an appropriate product design configuration. For example, a company plans to introduce three kinds of new wardrobe models. If each wardrobe consists of seven attributes, each taking six different levels, the number of possible wardrobe profiles is 6^7 (*i.e.*, 279 936). For designing a line of three kinds of wardrobes, the number of candidates is $279\,936^3$ (*i.e.*, 2.2×10^{16}). At the same time, there are compatibility issues among the levels of attributes. For example, when designing a customized wardrobe, the opening method of the cabinet door should be compatible with its shape. Obviously, sliding door is not compatible with curved surface but flat one. After the product is designed according to customers' needs, it enters the production stage. The production process must meet the requirements of product reliability and delivery time.

The inter-relationship between product design and production process also leads to high complexity. The design of product line should balance customer preferences and production decision. A selection for a preferred level may lead to lower product reliability, or incompatibility among the components that make up the product. On the other hand, managers are willing to outsource certain attribute level in order to reduce production cost. However, the outsourcing vendor may be far away from the company and cannot deliver the products in time; or the outsourcing vendor may have too many other urgent orders and cannot process the products in time; or although the company agrees on the delivery date with the outsourcing vendor, the delivery needs cannot be met by the customers due to relatively long waiting time. Consequently, customers may not prefer this level due to its delivery delay.

This paper constructs a nonlinear model around two intertwined issues of product design and production process. The objective is to maximize Per-capita-contribution Margin (PM) considering Reliability Penalty (PMRP), which implies the profit achieved from per customer. The imperialist competitive algorithm (ICA) has been applied to optimization problems such as Flexible Job shop Scheduling Problem (FJSP) [16], Traveling Salesman Problem (TSP) [7], and two-sided assembly lines [65]. However, the classical ICA is generally used for continuous problems, because the movement distance of a colony toward its imperialist is a random variable with uniform distribution in movement strategy. Discretizing the classical ICA is a necessity for solving discrete problems. We propose a discrete ICA (DICA) to assist managers in designing attributes' levels and making outsourcing strategy.

The remainder of this paper is organized as follows. Section 2 introduces the related literature of product line design optimization. Section 3 describes the integrated model of product line design optimization. Sections 4 and 5 introduce DICA, genetic algorithm (GA) and simulated annealing (SA) methods for the proposed model. In Section 6, DICA is compared with GA and SA through extensive numerical experiments. Section 7 concludes this paper with further study direction.

2. LITERATURE REVIEW

In this section, we present related literature review of studies about mass customization, product line design problems, and intelligent optimization algorithms.

2.1. Mass customization

Since the late 1980s, the analysis and implementation of mass customization (MC) systems have received considerable attention from researchers and practitioners [12]. Mass customization provides customized products or services in high volume at reasonable low cost through flexible processes. As demand for an individual product and its variant increases, mass personalized customization becomes imperative [38–40, 56, 64].

Many researchers analyzed MC from different aspects such as organizational reform, influent factors, customized capabilities, and product design method. Rungtusanatham and Salvador [44] studied organizational changes related to the transition from mass production to mass customization. Fogliatto *et al.* [12] believed that the main factors that determined the success of MC were customer demand, markets, value chain, technology, as well as customizable offer and knowledge. They identified several promising areas for future research, including rapid manufacturing, MC value, quality control, warranty, and empirical validation. Simon *et al.* [48] pointed out that MC required three basic capabilities: Distinguishing the product attributes that vary among customers; reusing or reorganizing existing production line resources; and helping customers recognize or create solutions to their own demands. As one of the main strategies to satisfy customers and ensure the company's survival in today's market, MC cannot be ignored [8]. In the process of mass customization, market-based product design connects the marketing and engineering domains, bringing profits to the company while meeting the needs of different customers.

2.2. Product line design problems

Product line design is an important part of mass customization. Its definition has been expanded with the development of related theories. Draganska and Jain [10] believed that product line was a product collection with the same core content and closely related products. The products in a product line have similar functions and can meet similar but slightly different customer needs. In order to meet the personalized orders, companies must continuously introduce new products or redesign existing ones. However, product configuration conflicts cause enormous problems, such as ineffective product design and unnecessary expenses, in the product or service design [4].

Many studies have proven that optimizing the product line at the product design stage can benefit both enterprises and customers. For example, Modrak and Soltysova [39] believed that the incompatibility of product components when customers customized products cause customer disappointment. A promising solution is to eliminate or at least reduce mutually incompatible components in the early stages of the design process. Luo [36] studied the product design of consumer durables, and pointed out the need to consider not only the relationship between consumer preferences and the engineering feasibility of each product, but also the revenue and cost interactions across products in the product line. He optimized the problem using GA. Albritton *et al.* [2] modeled customers' dynamic preferences in the development and robust design of a single product. Saridakis *et al.* [46] developed a particle swarm optimization (PSO) algorithm to design car lines. The algorithm optimizes the degree of differentiation *vs.* commonality among models in the line, and elevates customer satisfaction. Li *et al.* [26] optimize the price and quality of new products when there are existing products in the product line. They applied convex optimization and analyzed the properties of the optimal solution. Existing products often prompt a company to provide new products with higher quality and price.

Some scholars further extended product line design problems combined with customization, cellular manufacturing, risk, and social factors. For example, Tookanlou and Wong [49] extended the standard product line design problem by considering two interdependent aspects, namely customization level and product delivery lead time. In another study [50], they studied the product line design from market channels with vertical and horizontal consumer heterogeneity, and used mathematical analysis and game theory to analyze the influence of

channel structure on the optimal quality and customization level. Liu *et al.* [33] constructed a model to determine product variants through product line design and resource allocation in cellular manufacturing. Aydin and Badurdeen [3] proposed a new sustainable production line design method, which not only considered conflicting goals: total lifecycle profit (economic), as well as total energy and water usage (environmental), but also multiple life cycle closed-loop logistics with different recycling strategies. Liu *et al.* [34] discussed the balance between configuration design and production line under uncertain production requirements. They applied a heuristic algorithm based on greedy randomised adaptive search procedure combined with variable neighbourhood search for operation assignment. Gupta and Krishnan [18] found that suppliers' participation in the early stages of product design could reduce product development risk, detect potential problems early, improve product quality, eliminate product rework, reduce the internal complexity of product development by providing possible outsourcing channels, improve communication and information exchange, reduce delays, and get many other benefits. Jiao and Zhu [20] mainly studied the problems of product pricing and maintenance in the product line. They provided insights into the influence of sales price, warranty period and preventive maintenance interval. Luo *et al.* [37] proposed a multi-objective optimization model based on product platform. This model can optimize product family configuration and supplier pre-selection from multiple perspectives in the early stage of product design. The NSGA-II algorithm was developed to achieve Pareto non-dominated solutions of the model. Based on predecessors' attention to standardization work to simply product line, Yin *et al.* [61] put forward a model to quantitatively evaluate the degree of standardization, which further enriched the content of product line optimization research.

2.3. Optimization algorithms for product line design

Many researchers improved some classic intelligent algorithms to make them more suitable for the research problems. Belloni *et al.* [5] compared nine methods of product line optimization. These methods are mainly divided into three categories: (1) Methods that operate in attribute space (coordinate ascent, genetic algorithm, and simulated algorithm); (2) Methods that operate in product space (greedy heuristic, divide-and-conquer heuristic, and product-swapping heuristic); (3) Methods that partially evaluate formed products (dynamic programming heuristic, beam search heuristic, and nested partitions heuristic). Dou *et al.* [9] took advantage of iterative genetic algorithm (IGA), which makes the existing product and design process evolve bi-directionally in the customized process to seek the optimal design. Tsafarakis *et al.* [52] applied a new hybrid PSO approach to design an optimal industrial product line. They found that the incorporation of a mutation operator to the standard PSO algorithm significantly improved its performance. The hybrid PSO approach outperformed GA in a simulated study with artificial datasets pertaining to industrial cranes. Tsafarakis *et al.* [53] applied Fuzzy Self-Tuning Differential Evolution to a product line optimization problem, and compared it to GA.

2.4. Research gap

Product line design of personalized customized products usually depends on the marketing and engineering fields. Although existing research has proposed complex technologies to deal with the unique problems of each field, most of the interdependent problems across marketing and engineering have not been solved. In addition, existing research focused on balancing product cost and income, or on optimizing product diversity and universality of component assembly molds. However, there is a lack of research on the process of product design and production, compatibility issues among product components, product reliability, delivery time of finished products, and outsourcing decision-making. Although some optimization software, such as LINGO and CPLEX [17, 25], can solve these complicated models, they are only suitable for small-sized problems due to their long runtime. Consequently, this paper constructs a product line design model with the consideration of compatibility and reliability, and develops DICA to solve it even for large-sized instances.

3. PROBLEM STATEMENT AND FORMULATION

3.1. Background of the problem

Consider a company that produces customized cell-phones, computers and pads. Each product has different attributes and each attribute has several levels, which is shown in Table 1. Although managers hope to design products according to customers' preferences to increase market share, some configurations give rise to problems of compatibility and reliability. There is a compatibility case in Table 1. If a customer chooses the level of customized camera in the media attribute for a cell phone, the camera needs a high-resolution screen to match it.

Reliability refers to the quality standard that the product must reach before it can be put into use in the market. The reliability of product depends on the quality of its components. Some customers prefer new products that are associated with immature processing technology or components, and some choose low-cost material for cost saving. Such cases result in poor quality; thus, can't be produced. For example, the integrated graphics card of Table 1 is cheap, but it occupies the system's memory and affects the reliability performance. It may fail when dealing with complex images and videos due to low reliability. In contrast, the discrete graphics card is expensive and has its own memory source and power source, so it provides higher reliability performance than the integrated graphics card. Therefore, the choice of graphics card level should be based on a certain reliability standard. Low-reliability of products may easily cause product failure, which gives rise to high product rework rate and production cost, and thus reduces PM. Consequently, PMRP penalizes low-reliability products based on PM. The lower the reliability is, the smaller PMRP is. In addition, the company needs to consider the production time during designing and production process, because customers wish their products to be delivered on time. The production time depends on the decision of attributes' levels, and whether the associated components are outsourced. In order to maximize PMRP, company managers must make reasonable decisions including design, production planning, etc.

3.2. Problem hypotheses and notations

The problem is formulated based on the following hypotheses:

- (1) Customers: the number of customers is known in advance. Each customer has his/her own favorite product configuration. That is to say, each customer has different preference for certain level of a product's attribute, which is denoted by the variable part-worth.
- (2) Products: the number of products in a product line is known in advance. Each product has the same number of attributes, and each attribute has the same number of levels. There are some competitive counterparts in the market for each product variant in the product line.
- (3) Cost: the basic production cost and the cost of each component are known. If the operation for certain level of attribute in the production needs to be outsourced, the outsourcing cost will occur.
- (4) Reliability: the reliability of a product is calculated from the reliability of the components that make up the product. The more the product is unreliable, the larger the penalty cost is.
- (5) Time: the basic production time of product configuration is known, and the additional production time of customized components can be estimated. The total production time cannot be larger than the delivery due time expected by customers.

The input parameters and decision variables used in the model are listed as follows.

TABLE 1. The attributes and corresponding levels of each product.

Products	Attributes	Levels
Cell-phones	Capacity	64 GB
		128 GB
		256 GB
	Media	Camera
MP3 Camera+MP3		
Screen	Basic screen	
	Color screen	
	High solution screen	
Computers	Memory	8 GB
		16 GB
		32 GB
	Storage	512 GB SSD
		1 TB SSD 2 TB SSD
Graphics card	Integrated graphics card	
	Discrete graphics card	
	Core graphics card	
Pads	Capacity	128 GB
		256 GB
		1 TB
	Camera	8 MP Wide camera
		12 MP Wide camera
12 MP+10 MP Ultra Wide cameras		
Color	Silver	
	Space gray	
	Rose gold	

Input parameters

- I Number of customers, i denotes the index of customer ($i = 1, 2, \dots, I$).
 J Number of products of a product line in a company, j denotes the index of product ($j = 1, 2, \dots, J$).
 K Number of product attributes, k denotes the index of attribute ($k = 1, 2, \dots, K$).
 L Number of levels for each product attribute, l denotes the index of level ($l = 1, 2, \dots, L$).
 N Number of products in the market, n denotes the index of product ($n = 1, 2, \dots, N$).
 C^b Basic production cost.
 C_{jkl} Customized cost of component associated with level l of attribute k of product j .
 C'_{jkl} Outsourcing cost of component associated with level l of attribute k of product j .
 m The ratio of profit to cost.
 ω_{ikl} The part-worth that customer i perceives for level l of attribute k .
 β Penalty coefficient for reliability.
 $R_{jklk'l'}$ 1 if level l of attribute k is compatible with level l' of attribute k' for product j , and 0 otherwise ($k \neq k'$).
 b_{jkl} Reliability of component associated with level l of attribute k of product j .
 B_j Threshold for reliability of product j .
 d_j Basic production time of product j .
 d'_{jkl} Internal customized time of component associated with level l of attribute k of product j .

- s'_{jkl} Outsourcing time of component associated with level l of attribute k of product j .
 D_j Customer's expected delivery due time for product j .

Decision variables

- X_{jkl} 1 if customers choose level l of attribute k of product j , and 0 otherwise.
 Y_{jkl} 1 if outsourcing is needed for the component associated with level l of attribute k of product j , and 0 otherwise.

Intermediate variables

According to the input parameters and decision variables, the intermediate variables are defined as follows:

- P_j The cost of product j , $P_j = C^b + \sum_{k=1}^K \sum_{l=1}^L (C_{jkl} \cdot X_{jkl} + C'_{jkl} \cdot Y_{jkl})$.
 U_{ij} The utility that customer i perceives for product j . It is the sum of the part-worths. $U_{ij} = \sum_{k=1}^K \sum_{l=1}^L \omega_{ikl} \cdot X_{jkl}$.
 p_{ij} The probability that customer i will choose product j using Bradley Terry Luce (BTL) model [6, 35], $p_{ij} = \frac{U_{ij}}{\sum_{n=1}^N U_{in}}$.
 Ω_j Reliability of product j , $\Omega_j = \prod_{k=1}^K \left(\sum_{l=1}^L b_{jkl} \cdot X_{jkl} \right)$.
 SOP_j The share of preference for product j , $SOP_j = \frac{\sum_{i=1}^I p_{ij}}{I} = \frac{\sum_{i=1}^I \frac{U_{ij}}{\sum_{n=1}^N U_{in}}}{I} = \frac{\sum_{i=1}^I \frac{\sum_{k=1}^K \sum_{l=1}^L \omega_{ikl} \cdot X_{jkl}}{\sum_{n=1}^N \sum_{k=1}^K \sum_{l=1}^L \omega_{ikl} \cdot X_{nkl}}}{I}$.

3.3. Mathematical model

The problem can be formulated as a non-linear integer programming model as follows:

$$\begin{aligned} \text{Max PMRP} &= \sum_{j=1}^J P_j \cdot m \cdot SOP_j - \sum_{j=1}^J (1 - \Omega_j) \cdot \beta \\ &= \sum_{j=1}^J \left(C^b + \sum_{k=1}^K \sum_{l=1}^L (C_{jkl} \cdot X_{jkl} + C'_{jkl} \cdot Y_{jkl}) \right) \cdot m \\ &\quad \cdot \frac{\sum_{i=1}^I \frac{\sum_{k=1}^K \sum_{l=1}^L \omega_{ikl} \cdot X_{jkl}}{\sum_{n=1}^N \sum_{k=1}^K \sum_{l=1}^L \omega_{ikl} \cdot X_{nkl}}}{I} \\ &\quad - \sum_{j=1}^J \left[1 - \prod_{k=1}^K \left(\sum_{l=1}^L b_{jkl} \cdot X_{jkl} \right) \right] \cdot \beta \end{aligned} \quad (3.1)$$

$$\text{s.t.} \quad \sum_{l=1}^L X_{jkl} = 1, \quad \forall j, k \quad (3.2)$$

$$R_{jklk'l'} - X_{jkl} \cdot X_{jk'l'} \geq 0, \quad \forall j, k, l, k'l' \cdot (k \neq k') \quad (3.3)$$

$$\prod_{k=1}^K \left(\sum_{l=1}^L b_{jkl} \cdot X_{jkl} \right) \geq B_j, \quad \forall j \quad (3.4)$$

$$d_j + \text{Max}_{k \in \{1, \dots, K\}} \left(\sum_{l=1}^L [X_{jkl} \cdot (1 - Y_{jkl}) \cdot d'_{jkl} + X_{jkl} \cdot Y_{jkl} \cdot s'_{jkl}] \right) \leq D_j, \quad \forall j \quad (3.5)$$

$$X_{jkl} \in \{0, 1\}, \quad \forall j, k, l \quad (3.6)$$

$$Y_{jkl} \in \{0, 1\}, \quad \forall j, k, l. \quad (3.7)$$

TABLE 2. A customer's score of each level of different attributes for product 1 and its competitive products.

	Attribute 1			Attribute 2		Attribute 3			
	Level 1	Level 2	Level 3	Level 1	Level 2	Level 1	Level 2	Level 3	Level 4
Product 1	3	5	2	4	3	4	4	5	3
Competitive product 2	2	3	4	3	1	5	3	4	2
Competitive product 3	3	4	5	2	5	4	1	5	3
Competitive product 4	3	4	3	4	5	3	4	3	4

Objective function (3.1) is to maximize Per-capita-contribution Margin considering Reliability Penalty (PMRP), which consists of two terms defined as follows:

The first term is Per-capita-contribution Margin (PM), which implies that the profitability of a product is approximately evaluated using contribution margin per customer in the market.

- Cost P_j is incurred when product j is produced. The cost includes the basic production cost and the cost of producing customized components. The latter contains internal customized cost for those components produced inside the company and outsourcing cost for those components produced outside the company.
- SOP_j is the average of choice probabilities of all customers for product j . Table 2 gives an example of customers' preference scores for each level of various product attributes. The scores of product 1 and its competitive products are randomly selected from the integer interval $[1, 5]$. The scores can be transformed into part-worths of the attribute levels. The utility value of product 1 is usually expressed as the sum of the part-worths (with superscript “*”) of the preferred attribute levels. The utility values of competitive products are similarly calculated based on the preferred levels of product 1 (see *e.g.*, Tab. 3). Through the use of BTL choice model, the utilities that each customer assigns to each product and its competitive ones can be converted to choice probability p_{ij} . SOP_j is then derived from the aggregation of choice probabilities across the whole customer population.
- PM is obtained by multiplying the production cost P_j , the ratio of profit to cost m , and the preference share SOP_j .

The second term is the penalty cost which occurs when the product does not reach ideal reliability. Ω_j is obtained by accumulating the reliability of all components of product j . Penalty coefficient β is set according to the actual situation.

Constraint (3.2) ensures that only one level of each attribute of each product is selected. Constraint (3.3) ensures that the level selection for different attributes of each product can not violate the compatibility between each other when designing customized products. Constraint (3.4) guarantees that the accumulative value of all components' reliability can not be less than a given reliability threshold of the product. Constraint (3.5) ensures that the product can be delivered to the customer no later than his/her expected delivery time. Constraints (3.6) and (3.7) indicate that the decision variables are binary.

Previous studies have proved that the product line design problem belongs to the class of NP-hard combinatorial optimization problems [23, 24, 54], and thus no algorithm can identify the global optimum of the problem in tractable time [43]. The proposed model has two simultaneous phases. The first phase is mainly to solve the problem of product line design. The second phase is to make decision of outsourcing. Consequently, the proposed model is reasonably NP-hard since it integrates two above phases along with compatibility among product components, product reliability, and product delivery time. This excludes the chance of applying exhaustive exact algorithms such as backtracking, dynamic programming, or branch-and-bound technique, because these algorithms would be too time consuming even for a moderate scale problem.

TABLE 3. The part-worth that a customer assign to each level of different attributes for product 1 and its competitive products (normalized score).

	Attribute 1			Attribute 2		Attribute 3				Utility
	Level 1	Level 2	Level 3	Level 1	Level 2	Level 1	Level 2	Level 3	Level 4	
Product 1	0.300	0.500*	0.200	0.571*	0.429	0.250	0.250	0.312*	0.188	1.383
Competitive product 2	0.222	0.333*	0.445	0.750*	0.250	0.357	0.214	0.286*	0.143	1.369
Competitive product 3	0.272	0.364*	0.714	0.286*	0.714	0.308	0.077	0.385*	0.230	1.035
Competitive product 4	0.300	0.400*	0.300	0.444*	0.556	0.214	0.286	0.214*	0.286	1.058

4. DISCRETE IMPERIALIST COMPETITIVE ALGORITHM

Imperialist Competitive Algorithm (ICA) is classified into population-based methods, like genetic algorithm. ICA is a useful method in optimization because of its reasonable runtime and effectiveness [32, 45]. Moreover, ICA is a new solution approach, which has not been applied to product line design optimization problems. In ICA, a “virtual” country represents an encoded solution to the problem. The better countries are chosen to become the imperialists. The other countries are divided among the imperialists as colonies. The whole set of an imperialist and its colonies is called an empire. After seizing some colonies, the imperialists try to penetrate into their colonies by making them similar to themselves. Then the imperialists compete with each other to seize more colonies and gain more power. This process makes some empires more powerful and some weaker. The weaker empires eventually collapse.

In order to better adapt to the real situation, some effective changes are made in the classical ICA as follows. As mentioned above, each imperialist assimilates its colonies by making them similar to itself, which is called movement strategy. In the proposed DICA, colony movement is redefined to be a mutation or a mutation followed by at most S swimming. Movement strategy also applies the idea of simulated annealing (SA) to accept an inferior solution from a mutation or swimming. Although movement strategy helps the colonies to become better, it lessens the diversity of the population and makes the algorithm to trap into local optima easily. Thus, in order to prevent this situation and make the colonies to have better movement goal, each imperialist obtains a better solution from a mutation if possible.

Instead of using the possession probability method in classical ICA, we employ more intuitive and effective “roulette wheel” sampling to select the prior empire and let it to seize the weakest colony in the weakest empire in competition strategy.

In real world, the imperialists usually make some strategies to develop their power and domination, but the classical ICA does not provide a corresponding method. Obviously it is a shortcoming of the classical ICA [21]. Hence, to make up for that, a strategy called the imperialists’ development plan is applied in the proposed DICA. This helps the countries in the population to learn from each other to get better power.

4.1. The proposed DICA procedure

Some notations to be used in DICA are summarized as follows. The pseudo-code of DICA is shown in Algorithm 1, which is explained in Sections 4.2–4.5.

$Popsiz$	Population size.
N^{imp}	Number of imperialists.
N^{col}	Number of colonies.

C_i^{imp}	PMRP of imperialist i . ($i = 1, 2, \dots, N^{\text{imp}}$).
C_j^{col}	PMRP of colony j . ($j = 1, 2, \dots, N^{\text{col}}$).
C_k^{cou}	PMRP of country k . ($k = 1, 2, \dots, \text{Popsize}$).
$\overline{\text{NC}}_i$	Normalized power of imperialist i .
I_i^{col}	Number of colonies controlled by imperialist i in the initial population.
S_i^{col}	Set of colonies of empire i . ($i = 1, 2, \dots, N^{\text{imp}}$).
s	Counter for swimming steps.
S	Maximum number of swimming steps.
T	Temperature.
ρ	Cooling rate.
TC_i	Total power of empire i . ($i = 1, 2, \dots, N^{\text{imp}}$).
θ	Weight coefficient for mean colony cost.

Algorithm 1. DICA algorithm.

- 1: **Input:** Popsize = 50, $N^{\text{imp}} = 10$, $N^{\text{col}} = 40$, $S = 10$, $\theta = 0.1$;
 - 2: **Input:** Initial temperature $T = \frac{-|\text{OFV}(X_j) - \text{OFV}(X_i)|}{\ln(0.5)}$, where $\text{OFV}(X_i)$ and $\text{OFV}(X_j)$ are the objective function values of random solutions X_i and X_j , respectively [22]; Cooling rate $\rho = 0.8$;
 - 3: **Initial population:** Randomly generate Popsize = 50 feasible countries;
 - 4: Calculate initial power of each country k : $C_k^{\text{cou}} = \text{PMRP}$ of country k ;
 - 5: Select N^{imp} most powerful countries as the imperialists, and the remaining N^{col} countries as the colonies;
 - 6: Calculate initial normalized power of each imperialist i : $\overline{\text{NC}}_i = \frac{C_i^{\text{imp}}}{\sum_{e=1}^{N^{\text{imp}}} C_e^{\text{imp}}}$;
 - 7: Calculate number of colonies controlled by each imperialist i : $I_i^{\text{col}} = \text{round}\{\overline{\text{NC}}_i \cdot N^{\text{col}}\}$;
 - 8: For each imperialist i , randomly choose I_i^{col} unallocated colonies to it;
 - 9: **while** ($N^{\text{imp}} > 1$) **do**
 - 10: **for** ($i = 1 \rightarrow N^{\text{imp}}$) **do**
 - 11: Implement movement strategy for the countries in empire i (Algorithm 2);
 - 12: **end for**
 - 13: Implement competition strategy among all empires (Algorithm 3);
 - 14: Implement development strategy for all countries in the population (Algorithm 4);
 - 15: $T := T \times \rho$
 - 16: **end while**
-

4.2. Encoding scheme, initialization, and empires' formation

(1) Encoding scheme.

Each country we designed consists of two parts, namely the configuration decision of the products and the outsourcing decision of associated configuration components.

The first ingredient, named matrix $P-L_0$, is the specific product configuration, and shown in equation (4.1). The components of this $J \times K \times L$ matrix are variables X_{jkl} which are either 0 or 1. For example, in Figure 1 the term $X_{314} = 1$ means that attribute 1 of product 3 has a level of 4. While completing the matrix, constraints (3.2)–(3.6) should be satisfied.

$$P-L_0 = \begin{bmatrix} X_{1K1} & \cdots & X_{JK1} & & \\ \vdots & \ddots & & \ddots & \\ X_{1KL} & X_{111} & \cdots & X_{J11} & \\ & \ddots & \vdots & \vdots & \\ & & X_{11L} & \cdots & X_{J1L} \end{bmatrix}. \quad (4.1)$$

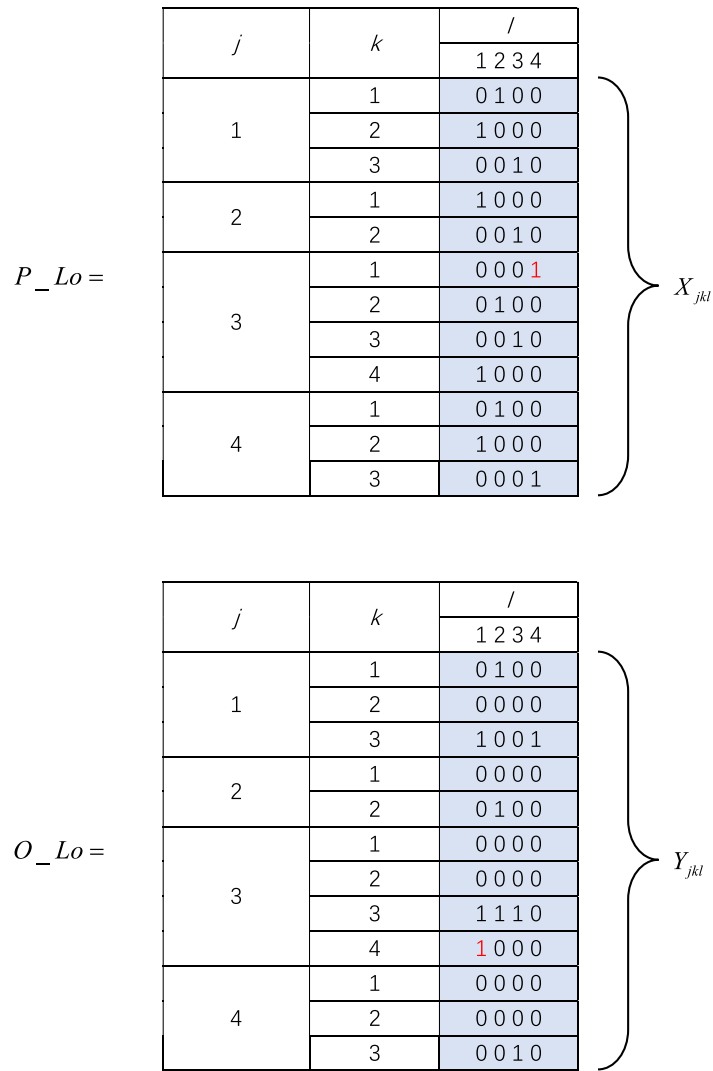


FIGURE 1. An instance of solution representation.

The second ingredient, named O_{L_0} , is related to whether the corresponding configuration component of the product needs to be outsourced, and shown in equation (4.2). The components of this $J \times K \times L$ matrix are variables Y_{jkl} which are either 0 or 1. For example, in Figure 1 the term $Y_{341} = 1$ means the component associated with level 1 of attribute 4 of product 3 requires external assistance. While completing the matrix, constraints (3.5) and (3.7) should be satisfied.

$$O_{L_0} = \begin{bmatrix} Y_{1K1} & \cdots & Y_{JK1} \\ \vdots & \ddots & \vdots \\ Y_{1KL} & Y_{111} & \cdots & Y_{J11} \\ \vdots & \vdots & \ddots & \vdots \\ & Y_{11L} & \cdots & Y_{J1L} \end{bmatrix}. \tag{4.2}$$

Combining the two ingredients described above, the solution representation is shown in (4.3). Figure 1 shows a typical instance.

$$\begin{array}{ccccccc}
 X_{1K1} & \cdots & X_{JK1} & & Y_{1K1} & \cdots & Y_{JK1} \\
 \vdots & \ddots & & \ddots & \ddots & & \ddots \\
 X_{1KL} & & X_{111} & \cdots & X_{J11} & & Y_{111} \cdots Y_{J11} \\
 & \ddots & \vdots & & \vdots & & \vdots \\
 & & X_{11L} & \cdots & X_{J1L} & & Y_{11L} \cdots Y_{J1L} \\
 & & & & (P_L0|O_L0). & &
 \end{array} \tag{4.3}$$

(2) Population initialization and empires' formation.

In DICA, a population including Popsiz = 50 countries is generated randomly, and some parameters $(N^{imp}, N^{col}, S, \theta, \rho)$ are initialized. The parameters are set according to some previous related works [7,16,65] and the results of pre-experiments. The initial temperature is specified by using Kia *et al.*'s method [22]. Some powerful countries with better objective function values are selected as the imperialists, and the rest as their colonies. It is assumed that the population initially contains N^{imp} imperialist countries and their N^{col} colonies (see Eq. (4.4)).

$$\text{Popsiz} = N^{imp} + N^{col}. \tag{4.4}$$

To assign the colonies to all imperialists, calculate the initial power (C_i^{imp}) of each imperialist. Then C_i^{imp} is normalized as \overline{NC}_i which is in the interval of (0,1) (see Eq. (4.5)). I_i^{col} colonies are randomly chosen from the population and assigned to each imperialist i using equation (4.6). Each imperialist and its colonies form an empire.

$$\overline{NC}_i = \frac{C_i^{imp}}{\sum_{e=1}^{N^{imp}} C_e^{imp}} \tag{4.5}$$

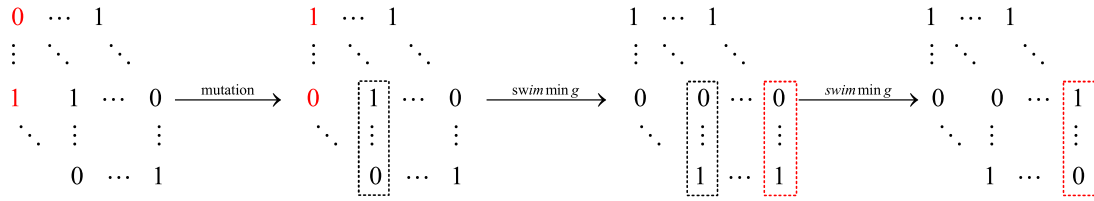
$$I_i^{col} = \text{round}\{\overline{NC}_i \cdot N^{col}\}. \tag{4.6}$$

4.3. The movement strategy (Algorithm 2)

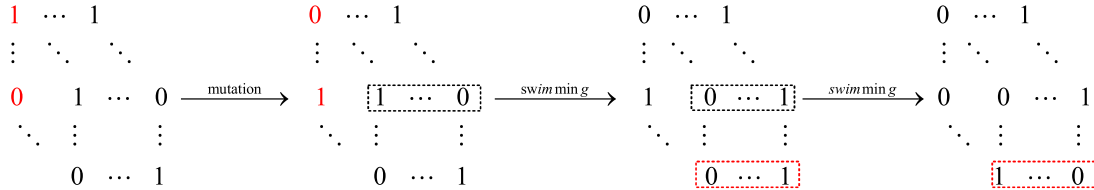
In DICA, colony movement is regarded as a mutation or a mutation followed by at most S swimming. Colony mutation refers to a change in a random dimensional direction toward its imperialist. For example, Figures 2a and 2b shows that a colony runs a mutation in the first/second dimension. Swimming indicates that when PMRP of the above colony is better than its last one, the colony mutates further in a random vector of the same dimension. If the colony continues to increase its PMRP, it continues swimming up to S times. This represents that the colony tends to keep moving if it is headed in the direction of increasingly favorable positions. The main idea of steps 2–19 in Algorithm 2 is that if PMRP of a colony after movement is better than its last one, accept the movement, and otherwise accept the movement by certain probability. The probability of accepting a worse solution is high to begin with, but it gets lower as the temperature goes down. It is worth noting that if all colonies of an imperialist move toward it, many identical colonies are obtained after some iterations. To prevent such assimilation among colonies, after all colonies of an imperialist complete their movement, the imperialist itself will mutate for the next iteration (steps 21–26 in Algorithm 2). If the best colony in an empire reaches a position with larger PMRP than that of its imperialist, exchange the positions of the colony and imperialist. When accomplishing the movement strategy, constraints (3.2)–(3.7) should be satisfied.

4.4. The competition strategy (Algorithm 3)

In Algorithm 3, first calculate the power of each country in the population. In order to model the fact that the total power of an empire is mainly affected by the power of its imperialist and partly affected by that of its



(a)



(b)

FIGURE 2. Movement of the colony toward its imperialist. (a) First matrix $P-L_0$. (b) Second matrix $O-L_0$.

colonies, we define total power as equation (4.7) in step 1 of Algorithm 3. A small value for θ (say 0.1) makes the total power of the empire mainly determined by the imperialist. Based on their total power, all empires start to make a competition. All empires try to take possession of colonies of other empires and control them. The empire with more total power may more likely possess colonies. Instead of using the possession probability method in classical ICA, we employ more intuitive and effective “roulette wheel” sampling to select the prior empire and hand the weakest colony to it in steps 2 and 3 of Algorithm 3. The advantage of roulette operator is that it will not disregard any individual in the pool. Consequently, it maintains the diversity of individuals within solution space [11]. In modeling collapse mechanism different criteria can be defined for considering empire’s powerless. In classical ICA, an empire collapses and is eliminated when it loses all of its colonies. To keep the population size invariant, we model the collapse mechanism by assuming that the imperialist with no colony is colonized by the prior empire (step 7 in Algorithm 3).

$$TC_i = C_i^{imp} + \theta \cdot \sum_{j \in S_i^{col}} C_j^{col}. \tag{4.7}$$

4.5. The development strategy (Algorithm 4)

Each iteration of competition urges each country to learn from others and develop itself. Inspired by the crossover operator in genetic algorithm, DICA adds and models this development strategy by assuming that each country is crossed over with any other country in the population. When finishing the development strategy, constraints (3.2)–(3.7) should be satisfied.

After many iterations of movement, competition, development strategies, only the most powerful empire remains.

5. GENETIC ALGORITHM AND SIMULATED ANNEALING

Genetic algorithm (GA) and simulated annealing (SA) have been proven to be effective algorithms, and are often used to act as comparison benchmark [28–31, 57].

Algorithm 2. Movement strategy.

```

1: for (each  $j \in S_i^{\text{col}}$ ) do
2:   Calculate current PMRP  $C_j^{\text{col}}$  of colony  $j$ ;
3:   Try to mutate colony  $j$  in a random dimensional direction toward imperialist  $i$ , and then calculate the temporary
    $C_j^{\text{col}'}$ ;
4:   Let  $s = 0$ ;
5:   while ( $s < S$ ) do
6:     if ( $C_j^{\text{col}'} > C_j^{\text{col}}$ ) then
7:       Accept the movement, and  $C_j^{\text{col}} = C_j^{\text{col}'}$ ;
8:       Try to swim in the previous dimensional direction, and then calculate the temporary  $C_j^{\text{col}'}$ ;
9:        $s := s + 1$ ;
10:    else
11:      Generate a random number  $r$  in the interval  $[0, 1]$ , and set  $\Delta = C_j^{\text{col}} - C_j^{\text{col}'}$ ;
12:      if ( $\exp(-\Delta \div T) > r$ ) then
13:        Accept the movement, and  $C_j^{\text{col}} = C_j^{\text{col}'}$ ;
14:      else
15:        Abandon the movement;
16:      end if
17:      Break;
18:    end if
19:  end while
20: end for
21: Try to mutate imperialist  $i$  in a random dimensional direction, and then calculate the temporary PMRP ( $C_i^{\text{imp}}$ );
22: if ( $C_i^{\text{imp}'} > C_i^{\text{imp}}$ ) then
23:   Accept the mutation, and  $C_i^{\text{imp}} = C_i^{\text{imp}'}$ ;
24: else
25:   Abandon the mutation;
26: end if
27: Identify the colony  $j^*$  with the highest PMRP in  $S_i^{\text{col}}$ ,  $j^* : C_{j^*}^{\text{col}} = \max_{j \in S_i^{\text{col}}} C_j^{\text{col}}$ ;
28: if ( $C_{j^*}^{\text{col}} > C_i^{\text{imp}}$ ) then
29:   Exchange position of colony  $j^*$  and imperialist  $i$ , and update their PMRP;
30: end if

```

Algorithm 3. Competition strategy.

```

1: Calculate total power of each empire  $i$ :  $\text{TC}_i = C_i^{\text{imp}} + \theta \cdot \sum_{j \in S_i^{\text{col}}} C_j^{\text{col}}$ ;
2: The “roulette wheel” sampling is used to select a priori empire  $i^*$  according to  $\text{TC}_i$ ;
3: Hand the weakest colony of the weakest empire to the empire  $i^*$ ;
4: The imperialist with no colony is colonized by the empire  $i^*$ ;
5: Update  $N^{\text{imp}}$  if necessary.

```

Algorithm 4. Development strategy

```

1: Try to crossover each country  $k$  with any other country in the population, generate a new temporary country  $k'$ , and
   calculate its PMRP ( $C_{k'}^{\text{cou}}$ ); The crossover mechanism is described in Algorithm 5 and displayed in Figure 3.
2: if ( $C_{k'}^{\text{cou}} > C_k^{\text{cou}}$ ) then
3:   replace country  $k$  with the new country  $k'$ ;
4: else
5:   Abandon the new country  $k'$ ;
6: end if

```

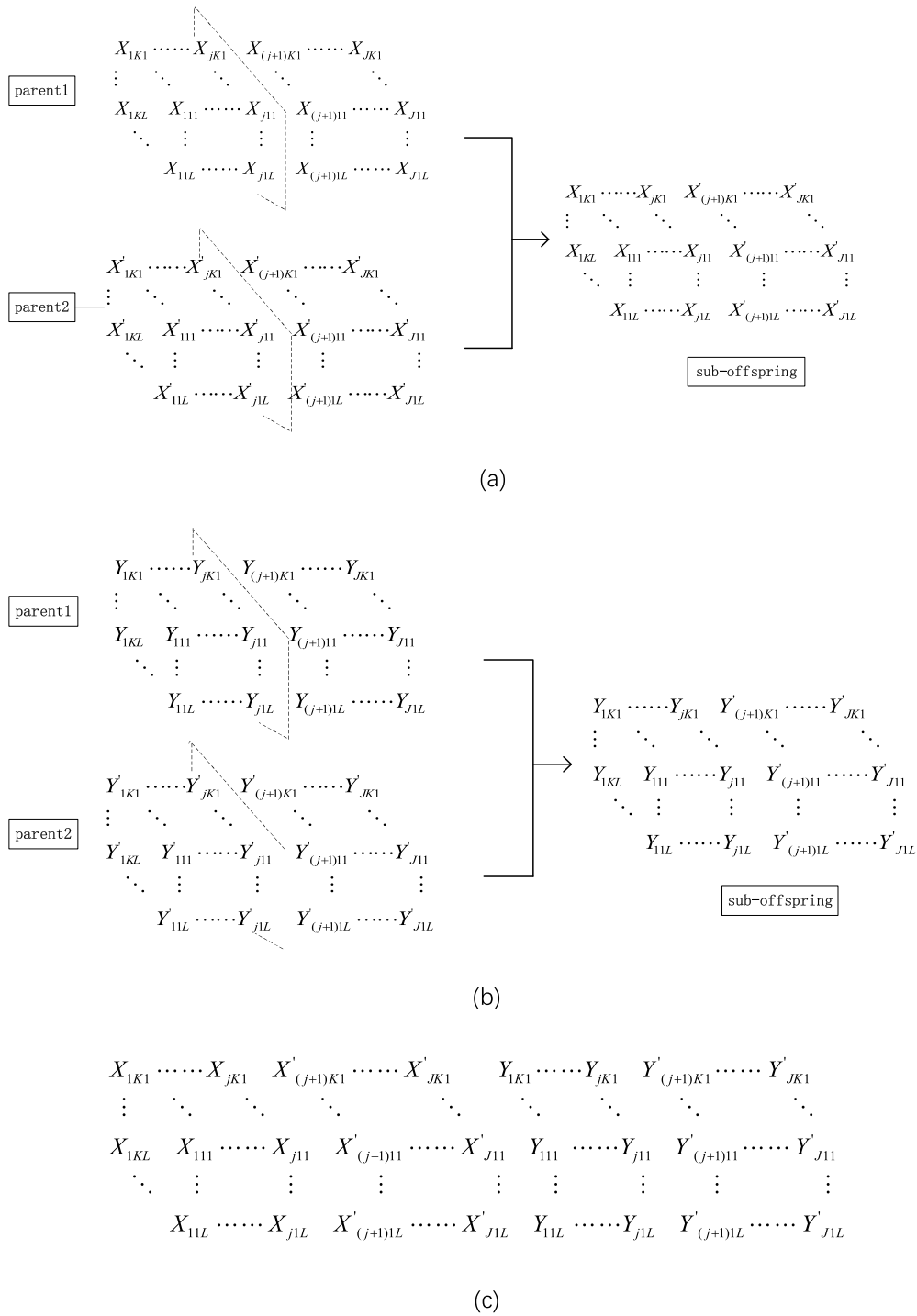


FIGURE 3. Crossover operator. (a) Crossover of P_L_0 . (b) Crossover of O_L_0 . (c) New offspring ($P_L_0|O_L_0$).

Algorithm 5. Crossover operator.

-
- 1: Each of two selected countries is divided into two parts (P_L0 and O_L0);
 - 2: Randomly select a dash plane as crossover point among P_L0 (or O_L0) in each parent country;
 - 3: The left part of the dash plane in P_L0 (or O_L0) of the first country and the right one in P_L0 (or O_L0) of the second country are combined into a new sub-offspring P_L1 (or O_L1). See Figures 3a and 3b;
 - 4: Combining the new sub-offspring P_L1 and O_L1 into a new country. See Figure 3c.
-

5.1. Genetic algorithm

GA was designed and proposed according to the evolution law of organisms in nature. It is a computational model of biological evolution that simulates the natural selection and genetic mechanism of Darwinian biological evolution. It acts as a widely used method for searching the optimal solution by simulating the natural evolution process [15]. The solutions are represented as chromosomes. A set of solutions defines the population of a generation. The population evolves as more fit solutions (chromosomes) substitute less fit solutions. In this way GA gradually nears to optimal solutions [42]. GA for the proposed model is described as follows.

- Chromosome representation: the chromosome representation employs the same solution structure of DICA.
- Fitness function: let F_g^s denote the fitness of s -th chromosome in generation g before selection. F_g^s is equal to the corresponding objective function value (*i.e.*, PMRP) of the s -th chromosome.
- Initial population: randomly generate Popsiz = 50 chromosomes to form the initial population. While generating the initial population, constraints (3.2)–(3.7) should be satisfied.
- Crossover: all chromosomes in the parent generation cross over mutually, the crossover probability is $P_c = 0.9$. The crossover mechanism is similar to Algorithm 5. If the fitness value of the offspring is better than that of the parent, the offspring will be retained in the new generation, and otherwise, it will be discarded. When finishing the crossover operator, constraints (3.2)–(3.7) should be satisfied.
- Mutation: each chromosome will mutate according to a given mutation probability $P_m = 0.2$. The mutation mechanism is to randomly generate new chromosomes. All off-springs from the mutation will be accepted for the new generation. When accomplishing the mutation operator, constraints (3.2)–(3.7) should be satisfied.
- Selection: the method “roulette wheel” sampling is used in the selection. Each chromosome is assigned a slice of the circular roulette wheel, and the size of the slice is proportional to the fitness value of the chromosome. The wheel is spun Popsiz times. On each spin, the chromosome under the wheel’s marker is selected into the pool of parents for the next generation.
- Stopping rule: the calculation is stopped when its runtime reaches DICA’s.

5.2. Simulated annealing (Algorithm 6)

SA is a stochastic optimization algorithm based on Monte-Carlo iterative solution strategy. Its main idea is based on the similarity between the annealing process of solid matter in physics and general combinatorial optimization problems [62, 63]. SA repeats the neighbor generation process to improve the objective function value. When exploring solution space, SA will accept an inferior solution probabilistically according to the metropolis criterion, jump out of the local optima, and obtain the global optimal solution [19]. It has been widely used in various optimization problems due to its strong local searching ability, easy operation and fast solving speed [27]. SA for the proposed model is described in Algorithm 6.

If an initial solution generated in DICA, GA, and SA is infeasible, it is eliminated, and another one is regenerated until it is feasible. If a solution generated through an operator (*e.g.*, mutation, swimming, development strategy, or crossover) is infeasible, it is eliminated, and the individual abandons this operator.

6. COMPUTATIONAL EXPERIMENTS

In this section, we validate DICA by comparing it with GA and SA. They are performed on a Core-based Lenovo-compatible personal computer with 1.60 GHz clock-pulse and 8 GB RAM. They are coded in C++,

Algorithm 6. Simulated annealing.

```

1: Input: Counter  $n = 0$ , Markov chain length  $\ell = 150$ ;
2: Input: Initial temperature  $T = \frac{-|\text{OFV}(X_j) - \text{OFV}(X_i)|}{\ln(0.5)}$ , where  $\text{OFV}(X_i)$  and  $\text{OFV}(X_j)$  are the objective function values
   of random solutions  $X_i$  and  $X_j$  respectively; Cooling rate  $\rho = 0.8$ ;
3: Initialization: The solution representation employs the same solution structure as that of DICA. Randomly generate
   an initial solution  $X_c$ . Set  $X_{\text{best}} = X_c$  and compare objective function value  $\text{OFV}(X_c)$ .
4: while (1) do
5:   while ( $n < \ell$ ) do
6:     Generate neighborhood solution  $X_k$  by using mutation strategy and random method with the same rate.
     Compute  $\text{OFV}(X_k)$ .
7:     if ( $\text{OFV}(X_k) \geq \text{OFV}(X_c)$ ) then
8:        $X_c = X_k$ 
9:       if ( $\text{OFV}(X_c) \geq \text{OFV}(X_{\text{best}})$ ) then
10:         $X_{\text{best}} = X_c$ 
11:      end if
12:     else
13:       Generate a random number  $r$  in the interval  $[0, 1]$ , and set  $\Delta = \text{OFV}(X_c) - \text{OFV}(X_k)$ 
14:       if ( $\exp(-\Delta/T) > r$ ) then
15:         $X_c = X_k$ 
16:       end if
17:     end if
18:      $n := n + 1$ 
19:     if (runtime reaches a given CPU time) then
20:       Terminate the procedure
21:     end if
22:   end while
23:    $T := T \times \rho$ ,  $n = 0$ 
24: end while

```

complied with the Microsoft Visual C++ 9.0 compiler, and tested under Microsoft Windows 10 operating system.

We observe the difference among solutions obtained by DICA, GA and SA within the same runtime. The CPU in Table 4 is the runtime of DICA without including input and output time, which is recorded by its program. Denote the CPU time of DICA as ψ . GA and SA are terminated when their CPU time reaches ψ . The results of experiment are shown in Table 4.

The performance of the mentioned algorithms is evaluated by the number of customers (I), the number of products (J), the number of product attributes (K), the number of corresponding levels of product attributes (L), and the number of similar competitive products in the market (N). The instances have been randomly generated to verify the proposed algorithm. The parameters used in the instances are also randomly generated. The experiment has five sets. For example, in the first set in Table 4, I is allowed to vary to test its effect, given $J = 4$, $K = 4$, $L = 4$ and $N = 8$. The other four sets in Table 4 test the effects of varying J , K , L and N . The other parameters for the randomly generated instances are given in Table 5. These parameters are given corresponding to the random values between the minimum and the maximum except three constants C^b , m and β . In order to make the optimization model meaningful, we should ensure that certain parts of the objective function are relatively balanced when setting the maximum and minimum values of the parameters. The constants can be set according to the real situation in the company. Given a typical instance with $I = 2000$, $J = 3$, $K = 3$, $L = 4$ and $N = 3$, Figure 4 shows their convergence within the same runtime, and Figure 5 illustrates the optimal solution. Table 4 represents the average of its associated 10 randomly generated instances. In Table 4, \bar{V}_Θ denotes the average objective function value using algorithm Θ , and $\Delta\bar{V}_\oplus^\otimes$ denotes the increasing percentage of \bar{V}_\otimes over \bar{V}_\oplus . A specific example in Table 6 illustrates how to obtain the blue values (\bar{V}_Θ and CPU) in Table 4.

TABLE 4. Performance comparison between DICA, GA and SA for impact parameters.

$J = 4, K = 4$ $L = 4, N = 8$		\bar{V}_{DICA}	\bar{V}_{GA}	\bar{V}_{SA}	$\Delta\bar{V}_{\text{GA}}^{\text{DICA}}$ (%)	$\Delta\bar{V}_{\text{SA}}^{\text{DICA}}$ (%)	CPU (s)
I	1000	4474	3996	4045	12	11	52
	4000	4503	3986	4109	13	10	205
	7000	4514	3944	4005	14	13	375
	10 000	4426	3908	4009	13	10	527
$I = 1000, K = 5$ $L = 5, N = 12$							
J	3	3133	2710	2839	16	10	117
	5	5845	5038	5124	16	14	230
	7	8532	7381	7486	16	14	355
	9	10 929	9621	9900	14	10	480
$I = 4000, J = 5$ $L = 3, N = 8$							
K	3	3518	3274	3269	7	8	145
	4	4553	4123	4180	10	9	195
	5	5543	4749	4881	17	14	253
	6	6478	5604	5748	16	13	790
$I = 7000, J = 4$ $K = 4, N = 6$							
L	2	3941	3619	3712	9	6	156
	6	7869	6881	7097	14	11	394
	10	11 220	10 223	10 422	10	8	640
	14	14 617	13 487	13 681	8	7	881
$I = 10 000, J = 3$ $K = 3, L = 4$							
N	4	5131	4859	4877	6	5	158
	10	2155	1937	2000	11	8	354
	16	1282	1155	1189	11	8	528
	22	891	825	829	8	7	689

DICA is compared with GA and SA to examine its performance. In Table 4, we can observe that $\Delta\bar{V}_{\text{GA}}^{\text{DICA}}$ can reach 6%~17% notwithstanding the different parameters. At the same time, $\Delta\bar{V}_{\text{SA}}^{\text{DICA}}$ can reach 5%~14% in spite of the change of parameters.

The reasons why DICA gets better results than GA and SA are as follows:

- (1) In DICA, colony movement is modified to a mutation or a mutation followed by a given swimming strategy. That is to say, the colony tends to keep moving if it is headed in the direction of increasingly advantageous position. This will result in a deep search in the solution space.
- (2) In order to prevent falling into local optima, DICA accepts colony's worse movement with a probability which drops along with the iterations.
- (3) Each imperialist tries to mutate after colony movement, which increases the diversity of population for the next iteration.
- (4) We have designed a new empire collapse mechanism, *i.e.*, the imperialist without a colony will be ruled by the prior empire. This can maintain the stability of the population size.

TABLE 5. Parameters for randomly generated instances.

Parameter	Value	Min	Max
C^b : Basic production cost	2000		
m : The ratio of profit to cost	0.3		
β : Penalty coefficient for reliability	1000		
C_{jkl}^c : Customized cost for level l of attribute k of product j		100	3000
C_{jkl}^o : Outsourcing cost for level l of attribute k of product j		500	3000
ω_{ikl} : The part-worth that customer i assigns to level l of attribute k		0.1	0.5
b_{jkl} : Reliability of components for level l of attribute k of product j		0.97	0.99
B_j : Threshold for reliability of product j		0.91	0.93
d_j : Basic production time of product j		1	4
d_{jkl}^i : Internal customized time for level l of attribute k of product j		1	3
s_{jkl}^o : Outsourcing time for level l of attribute k of product j		1	3
D_j : Customer's expected delivery due time for product j		15	20

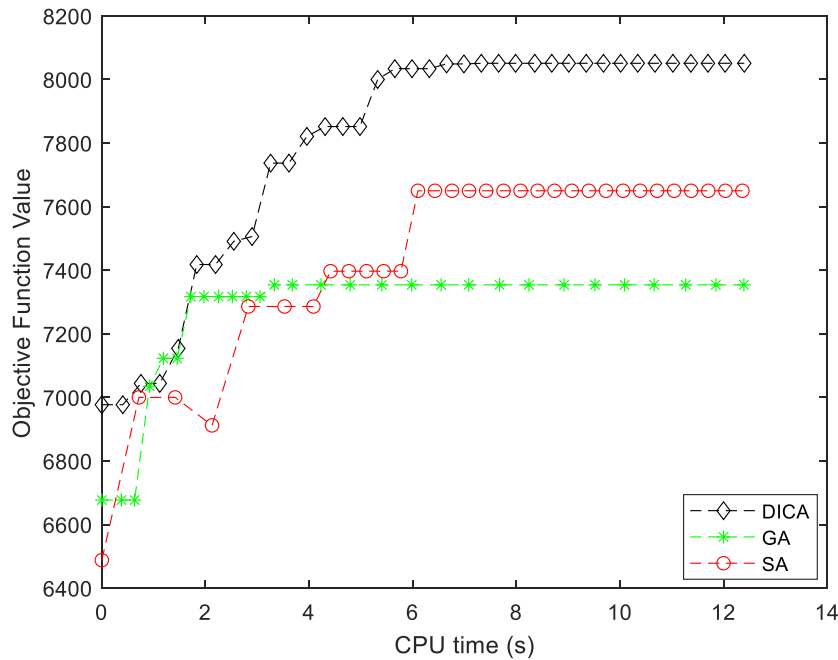


FIGURE 4. Typical convergence of DICA, GA and SA within the same runtime.

(5) We incorporate the development strategy into DICA, so that the countries can learn from each other.

Based on the above analysis, we find the internal logicity and principle that DICA can show better ability of exploitation and exploration than GA and SA.

7. CONCLUSIONS

This paper proposes a product line optimization model that considers design constraint and reliability, and develops an effective Discrete Imperialist Competitive Algorithm (DICA). This optimization model integrates many practical factors of multi-product design and production decision-making. The main considerations include

$P_{Lo} =$

j	k	l			
		1	2	3	4
1	1	1	0	0	0
	2	0	0	1	0
	3	1	0	0	0
2	1	0	0	0	1
	2	0	0	0	1
	3	0	0	1	0
3	1	0	0	0	1
	2	0	1	0	0
	3	0	1	0	0

X_{jkl}

$O_{Lo} =$

j	k	l			
		1	2	3	4
1	1	1	1	1	0
	2	1	1	1	1
	3	0	1	0	1
2	1	1	0	0	1
	2	1	1	1	0
	3	0	1	0	0
3	1	1	0	1	1
	2	1	1	1	1
	3	1	0	1	0

Y_{jkl}

FIGURE 5. The optimal solution of the instance with $I = 2000, J = 3, K = 3$ and $L = 4$.

TABLE 6. Results of 10 randomly generated instances.

	1	2	3	4	5	6	7	8	9	10	Average value
DICA	4452	4765	4425	4528	4297	4423	4442	4600	4651	4156	4474
GA	3931	3985	4098	4282	4054	3955	4170	3956	3990	3539	3996
SA	3987	4084	4170	4184	4005	4097	4023	3939	4135	3823	4045
CPU	47	52	51	51	51	55	52	54	53	53	52

the customized cost inside and outside the company, the costumers' bias towards various attributes' levels, the quality of customized components and finished products, the compatibility among different attributes' levels, and the delivery due date of finished products. The objective is to maximize the Per-capita-contribution Margin considering Reliability Penalty (PMRP).

Considerable modifications made to the classical ICA in DICA include:

- (1) The classic ICA is mainly used to solve continuous optimization problems, which is not suitable for our discrete problem. So we design a new movement strategy including mutation and swimming for the discrete colonies in the population. In order to prevent the reduction of population diversity due to colony movement toward its imperialist, we let the imperialist mutate after the colony movement in each iteration.

- (2) In the process of empire competition, we use “roulette wheel” sampling instead of possession probability to select the prior empire that will control the other colony.
- (3) In the collapse mechanism, the weakest empire with no colony will be reassigned instead of being destroyed directly as in ICA.
- (4) We supplement a development strategy so that all countries can learn from each other actively.

An interesting direction for future research is to optimize multiple objectives at the same time. Since consumer preferences may be affected by past experience and previous choices, future research can consider dynamic customer preference levels that change over time to make product design more robust. In addition, the introduction of new products will generate a competitive response in the market. It has an impact on the sales of the company’s existing products, or gives rise to the imitation of other companies in developing the same type of new products. Game theory is worth to be tried in this kind of product line design problems with market competition and evolution. Some advanced heuristic methods, such as bees algorithm and evolutionary algorithm [13–15, 59, 63], are worthy of comparison with the proposed DICA to further verify the effectiveness of DICA. In the future, the proposed nonlinear programming model will be linearized, and solved by CPLEX to obtain global optimal solutions for small-size or medium-size problems. We can check the performance of certain heuristic method according to the gap between the near optimal solution by the heuristic and the global optimal solution by CPLEX. The proposed DICA can also be extensively applied to the distributed, collaborative and intelligent social manufacturing mode [47, 58, 60], such that customers can customize the products with superior compatibility and reliability in the processes of innovation and design.

Acknowledgements. This research was supported by the Humanities and Social Sciences Planned Foundation of the PRC Ministry of Education under Grant 21YJA630065, and the Zhejiang Provincial Natural Science Foundation of China under Grants LY19G020015 and LY19A010007. The authors are grateful for the financial supports.

REFERENCES

- [1] V.S. Abedi, Compartmental diffusion modeling: describing customer heterogeneity & communication network to support decisions for new product introductions. *Phys. A Stat. Mech. App.* **536** (2019) 1–18.
- [2] M.D. Albritton and P.R. McMullen, Optimal product design using a colony of virtual ants. *Eur. J. Oper. Res.* **176** (2007) 498–520.
- [3] R. Aydin and F. Badurdeen, Sustainable product line design considering a multi-lifecycle approach. *Resour. Conserv. Recycl.* **149** (2019) 727–737.
- [4] A. Aydogdu, A. Akin and M.P. Saka, Design optimization of real world steel space frames using artificial bee colony algorithm with levy flight distribution. *Adv. Eng. Softw.* **92** (2016) 1–14.
- [5] A. Belloni, R. Freund, M. Selove and D. Simester, Optimizing product line designs: Efficient methods and comparisons. *Manage. Sci.* **54** (2008) 1544–1552.
- [6] R.A. Bradley and M.E. Terry, Rank analysis of incomplete block designs: the method of paired comparisons. *Biometrika* **39** (1952) 324–345.
- [7] M. Chen, S. Chen and P. Chang, Imperial competitive algorithm with policy learning for the traveling salesman problem. *Soft Comput.* **21** (2015) 1–13.
- [8] J. Daaboul, C.D. Cunha, A. Bernard and F. Laroche, Design for mass customization: product variety vs. process variety. *CIRP Ann. – Manuf. Technol.* **60** (2011) 169–174.
- [9] R. Dou, Y. Zhang and G. Nan, Customer-oriented product collaborative customization based on design iteration for tablet personal computer configuration. *Comput. Ind. Eng.* **99** (2016) 474–486.
- [10] M. Draganska and D.C. Jain, Product-line length as a competitive tool. *J. Econ. Manage. Strategy* **14** (2005) 1–28.
- [11] H. Faris, A.M. Al-Zoubi, A.A. Heidari, I. Aljarah, M. Mafarja, M.A. Hassonah and H. Fujita, An intelligent system for spam detection and identification of the most relevant features based on evolutionary Random Weight Networks. *Inf. Fusion* **48** (2019) 67–83.
- [12] F.S. Fogliatto, G.J.C. da Silveira and D. Borenstein, The mass customization decade: an updated review of the literature. *Int. J. Prod. Econ.* **138** (2012) 14–25.
- [13] Y. Fu, M. Zhou, X. Guo and L. Qi, Scheduling dual-objective stochastic hybrid flow shop with deteriorating jobs via bi-population evolutionary algorithm. *IEEE Trans. Syst. Man Cybern. Syst.* **50** (2020) 5037–5048.
- [14] K. Gao, Z. Cao, L. Zhang, Z. Chen, Y. Han and Q. Pan, A review on swarm intelligence and evolutionary algorithms for solving flexible job shop scheduling problems. *IEEE-CAA J. Autom. Sinica* **6** (2019) 904–916.
- [15] M. Ghahramani, Y. Qiao and M. Zhou, AI-based modeling and data-driven evaluation for smart manufacturing processes. *IEEE-CAA J. Autom. Sinica* **7** (2020) 1026–1037.

- [16] B. Ghasemishabankareh, N. Shahsavari-Pour, M.A. Basiri and X. Li, A hybrid imperialist competitive algorithm for the flexible job shop problem. In: *Artificial Life and Computational Intelligence*, edited by T. Ray, R. Sarker and X. Li. Springer International Publishing, Cham (2016) 221–233.
- [17] M. Goswami, Y. Daultani and M.K. Tiwaris, An integrated framework for product line design for modular products: product attribute and functionality-driven perspective. *Int. J. Prod. Res.* **55** (2017) 3862–3885.
- [18] S. Gupta and V. Krishnan, Integrated component and supplier selection for a product family. *Prod. Oper. Manage.* **8** (2010) 163–182.
- [19] X. Han, Y. Dong, L. Yue, Q. Xu, G. Xie and X. Xu, State-transition simulated annealing algorithm for constrained and unconstrained multi-objective optimization problems. *Appl. Intell.* **10** (2020) 1–13.
- [20] C. Jiao and X. Zhu, Optimal design of sales and maintenance under the renewable warranty. *RAIRO-Oper. Res.* **52** (2018) 529–542.
- [21] S. Karimi, Z. Ardalan, B. Naderi and M. Mohammadi, Scheduling flexible job-shops with transportation times: mathematical models and a discrete imperialist competitive algorithm. *Appl. Math. Model.* **41** (2017) 667–682.
- [22] R. Kia, A. Baboli, N. Javadian, R. Tavakkoli-Moghaddam, M. Kazemi and J. Khorrami, Solving a group layout design model of a dynamic cellular manufacturing system with alternative process routings, lot splitting and flexible reconfiguration by simulated annealing. *Comput. Oper. Res.* **39** (2012) 2642–2658.
- [23] R. Kohli and R. Krishnamurti, Optimal product design using conjoint analysis: computational complexity and algorithms. *Eur. J. Oper. Res.* **40** (1989) 186–195.
- [24] R. Kohli and R. Sukumar, Heuristics for product line design using conjoint analysis. *Manage. Sci.* **36** (1990) 1464–1478.
- [25] S. Kumar and A.K. Chatterjee, A profit maximising product line optimization model under monopolistic competition. *Int. J. Prod. Res.* **53** (2015) 1584–1595.
- [26] H. Li, S. Webster and G. Yu, Product design under multinomial logit choices: optimization of quality and prices in an evolving product line. *Manuf. Serv. Oper. Manage.* **22** (2020) 1–15.
- [27] Y. Li, C. Wang, L. Gao, Y. Song and X. Li, An improved simulated annealing algorithm based on residual network for permutation flow shop scheduling. *Complex Intell. Syst.* **7** (2021) 1173–1183.
- [28] C. Liu and J. Wang, Cell formation and task scheduling considering multi-functional resource and part movement using hybrid simulated annealing. *Int. J. Comput. Intel. Syst.* **9** (2016) 765–777.
- [29] C. Liu, J. Wang and J.Y.-T. Leung, Worker assignment and production planning with learning and forgetting in manufacturing cells by hybrid bacteria foraging algorithm. *Comput. Ind. Eng.* **96** (2016) 162–179.
- [30] C. Liu, J. Wang, J.Y.-T. Leung and K. Li, Solving cell formation and task scheduling in cellular manufacturing system by discrete bacteria foraging algorithm. *Int. J. Prod. Res.* **54** (2016) 923–944.
- [31] C. Liu, J. Wang and J.Y.-T. Leung, Integrated bacteria foraging algorithm for cellular manufacturing in supply chain considering facility transfer and production planning. *Appl. Soft Comput.* **62** (2018) 602–618.
- [32] C. Liu, J. Wang and M. Zhou, Reconfiguration of virtual cellular manufacturing systems via improved imperialist competitive approach. *IEEE Trans. Autom. Sci. Eng.* **16** (2019) 1301–1314.
- [33] C. Liu, Y. Liu and J. Wang, A revised imperialist competition algorithm for cellular manufacturing optimization based on product line design. *J. Ind. Manage. Optim.* (2021) DOI: [10.3934/jimo.2021175](https://doi.org/10.3934/jimo.2021175).
- [34] X. Liu, J. Chen and A. Li, Optimisation of line configuration and balancing for reconfigurable transfer lines considering demand uncertainty. *Int. J. Prod. Res.* **59** (2021) 444–466.
- [35] D.R. Luce, Individual choice behavior: a theoretical analysis. *J. Am. Stat. Assoc.* **67** (1959) 1–15.
- [36] L. Luo, Product line design for consumer durables: an integrated marketing and engineering approach. *J. Marketing Res.* **48** (2011) 128–139.
- [37] X. Luo, W. Li, C.K. Kwong and Y. Cao, Optimisation of product family design with consideration of supply risk and discount. *Res. Eng. Design* **27** (2016) 37–54.
- [38] D.T. Matt, E. Rauch and P. Dallasega, Trends towards distributed manufacturing systems and modern forms for their design. *Proc. CIRP* **33** (2015) 185–190.
- [39] V. Modrak and Z. Soltysova, Management of product configuration conflicts to increase the sustainability of mass customization. *Sustainability* **12** (2020) 1–14.
- [40] V. Modrak, D. Marton and S. Bednar, The influence of mass customization strategy on configuration complexity of assembly systems. *Proc. CIRP* **33** (2015) 538–543.
- [41] P.W. Murray, B. Agard and M.A. Barajas, Forecast of individual customer’s demand from a large and noisy dataset. *Comput. Ind. Eng.* **118** (2018) 33–43.
- [42] S.S. Pal and S. Kar, Forecasting stock market price by using fuzzified choquet integral based fuzzy measures with genetic algorithm for parameter optimization. *RAIRO-Oper. Res.* **54** (2020) 597–614.
- [43] C.H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*. Prentice-Hall, Inc., New Jersey, USA (1982).
- [44] M.J. Rungtusanatham and F. Salvador, From mass production to mass customization: Hindrance factors, structural inertia and transition hazard. *Prod. Oper. Manage.* **17** (2010) 385–396.
- [45] F. Sarayloo and R. Tavakkoli-Moghaddam, Imperialistic competitive algorithm for solving a dynamic cell formation problem with production planning. In: *Advanced Intelligent Computing Theories and Applications*, edited by D.S. Huang, Z. Zhao, V. Bevilacqua and J.C. Figueroa. *International Conference on Advanced Intelligent Computing Theories & Applications: Intelligent Computing*. Springer, Berlin, Heidelberg (2010) 266–276.

- [46] C. Saridakis, S. Tsafarakis, P. Delias, G. Baltas and N. Matsatsinis, Optimizing differentiation and commonality levels among models in car line-ups: An empirical application of a nature-inspired heuristic mechanism. *Expert Syst. App.* **42** (2014) 2323–2335.
- [47] X. Shang, F. Wang, G. Xiong, T.R. Nyberg, Y. Yuan, S. Liu, C. Guo and S. Bao, Social manufacturing for high-end apparel customization. *IEEE-CAA J. Autom. Sinica* **5** (2018) 489–500.
- [48] J. Simon, M. Trojanova, J. Zbihlejš and J. Sarosi, Mass customization model in food industry using industry 4.0 standard with fuzzy-based multi-criteria decision making methodology. *Adv. Mech. Eng.* **10** (2018) 113–145.
- [49] P.B. Tookanlou and H. Wong, Determining the optimal customization levels, lead times, and inventory positioning in vertical product differentiation. *Int. J. Prod. Econ.* **221** (2020) 1–20.
- [50] P.B. Tookanlou and H. Wong, Product line design with vertical and horizontal consumer heterogeneity: the effect of distribution channel structure on the optimal quality and customization levels. *Eur. J. Market.* **55** (2020) 95–131.
- [51] S. Tsafarakis, Y. Marinakis and N. Matsatsinis, Particle swarm optimization for optimal product line design. *Int. J. Res. Market.* **28** (2011) 13–22.
- [52] S. Tsafarakis, C. Saridakis, G. Baltas and N. Matsatsinis, Hybrid particle swarm optimization with mutation for optimizing industrial product lines: an application to a mixed solution space considering both discrete and continuous design variables. *Ind. Marketing Manage.* **42** (2013) 496–506.
- [53] S. Tsafarakis, K. Zervoudakis, A. Andronikidis and E. Altsitsiadis, Fuzzy self-tuning differential evolution for optimal product line design. *Eur. J. Oper. Res.* **287** (2020) 1–9.
- [54] S. Tsafarakis, K. Zervoudakis, A. Andronikidis and E. Altsitsiadis, Fuzzy self-tuning differential evolution for optimal product line design. *Eur. J. Oper. Res.* **287** (2020) 1161–1169.
- [55] S. Vinodh, G. Sundararaj, S.R. Devadasan, D. Kuttalingam and D. Rajanayagam, Amalgamation of mass customisation and agile manufacturing concepts: the theory and implementation study in an electronics switches manufacturing company. *Int. J. Prod. Res.* **48** (2010) 2141–2164.
- [56] Y. Wang, H. Ma, J. Yang and K. Wang, Industry 4.0: A way from mass customization to mass personalization production. *Adv. Manuf.* **5** (2017) 311–320.
- [57] J. Wang, C. Liu and K. Li, A hybrid simulated annealing for scheduling in dual resource cellular manufacturing system considering worker movement. *Automatika* **60** (2019) 172–180.
- [58] Y. Wang, X. Shang, R. Qin, G. Xiong and T.R. Nyberg, Social manufacturing: a paradigm shift for smart prosumers in the era of societies 5.0. *IEEE Trans. Comput. Soc. Syst.* **6** (2019) 822–829.
- [59] G. Wei, Q. Wu and M. Zhou, A hybrid probabilistic multiobjective evolutionary algorithm for commercial recommendation systems. *IEEE Trans. Comput. Soc. Syst.* **8** (2021) 589–598.
- [60] G. Xiong, F. Wang, T.R. Nyberg, X. Shang, M. Zhou, Z. Shen, S. Li and C. Guo, From mind to products: towards social manufacturing and service. *IEEE-CAA J. Autom. Sinica* **5** (2018) 47–57.
- [61] L. Yin, M. He, W. Xie, F. Yuan, D. Chen and Y. Sun, A quantitative model of universalization, serialization and modularization on equipment systems. *Phys. A Stat. Mech. App.* **508** (2018) 359–366.
- [62] H. Yuan, J. Bi and M. Zhou, Multiqueue scheduling of heterogeneous tasks with bounded response time in hybrid green IaaS clouds. *IEEE Trans. Ind. Inf.* **15** (2019) 5404–5412.
- [63] H. Yuan, M. Zhou, Q. Liu and A. Abusorrah, Fine-grained resource provisioning and task scheduling for heterogeneous applications in distributed green clouds. *IEEE-CAA J. Autom. Sinica* **7** (2020) 1380–1393.
- [64] X. Zhang, X. Ming, Z. Liu, Y. Qu and D. Yin, State-of-the-art review of customer to business (C2B) model. *Comput. Ind. Eng.* **132** (2019) 207–222.
- [65] Y. Zhang, X. Hu and C. Wu, Improved imperialist competitive algorithms for rebalancing multi-objective two-sided assembly lines with space and resource constraints. *Int. J. Prod. Res.* **58** (2020) 3589–3617.

Subscribe to Open (S2O)

A fair and sustainable open access model



This journal is currently published in open access under a Subscribe-to-Open model (S2O). S2O is a transformative model that aims to move subscription journals to open access. Open access is the free, immediate, online availability of research articles combined with the rights to use these articles fully in the digital environment. We are thankful to our subscribers and sponsors for making it possible to publish this journal in open access, free of charge for authors.

Please help to maintain this journal in open access!

Check that your library subscribes to the journal, or make a personal donation to the S2O programme, by contacting subscribers@edpsciences.org

More information, including a list of sponsors and a financial transparency report, available at: <https://www.edpsciences.org/en/maths-s2o-programme>