Computers & Industrial Engineering 117 (2018) 60–70

Contents lists available at ScienceDirect

Computers & Industrial Engineering

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

A delayed product differentiation model for cloud manufacturing

Chunyang Yu[a], Shandong Mou[c], Yangjian Ji[b,*], Xun Xu[d], Xinjian Gu[b]

[a] Department of Industrial Design, Zhejiang University City College, Hangzhou 310015, China
[b] Key Laboratory of Advanced Manufacturing Technology of Zhejiang Province, College of Mechanical Engineering, Zhejiang University, Hangzhou 310027, China
[c] Graduate School of Management, the Business School, University of Auckland, Auckland 1142, New Zealand
[d] Department of Mechanical Engineering, University of Auckland, Auckland 1142, New Zealand

ARTICLE INFO

Keywords:
Cloud manufacturing
Delayed product differentiation
Inventory management
Postponement

ABSTRACT

Cloud manufacturing (CMfg) enables in-depth customization but raises demand uncertainties. Delayed product differentiation (DPD) is one attempt to solve such problem. However, current DPD approaches are mostly focused on the production of a dominant company, which tend to hold the supply chains with fixed network structures. Nevertheless, CMfg requires more flexible structures (agile supply chain) to facilitate multiple manufacturers to access external resources with various manners. Therefore, linking DPD to CMfg becomes an important research topic, and the paper proposes an optimization model, namely DPDCM, for such purpose. The model is established on the basis of integrating the order-release, generic inventory and sourcing decisions, and is formulated as an integer programming problem, to meet diverse requirements of companies for carrying out DPD in CMfg environment. Case studies on bicycle and industrial-transformer manufacturing have been applied and analyzed, in which genetic algorithm is adopted to obtain near-optimal solutions. It validates the effectiveness, flexibility and universality of DPDCM.

1. Introduction

Cloud manufacturing (CMfg) has recently become popular in manufacturing industry. By virtualizing physical resources as consumable services over the Internet, CMfg creates an integrated, distributed, and service-oriented manufacturing paradigm (Xu, 2012), in which the users (service demanders) – either the original equipment manufacturers (OEMs) or suppliers – can connect to the desirable services via cloud-based applications (see Fig. 1). It enables agile supply chain, of which the network structure is unfixed, in opposite to the conventional supply chain (Wu, Greer, Rosen, & Schaefer, 2013; Jassbi et al., 2016). Under such flexible environment, the manufacturing firms (service demanders) become more accessible to external resources and capabilities, and capable of delivering more in-depth customization solutions (Yu & Xu, 2015). However, the demand uncertainty is highly increased at the same time, which hurts the manufacturing efficiency. Hence, to provide higher varieties of products with efficiency, delayed product differentiation (DPD) is one attempt for CMfg.

DPD means to delay the final customization of a product (the point of differentiation) as much as possible, usually until the arrivals of customer orders, to reduce the manufacturing lead time (Lee & Billington, 1994). It normally divides the entire manufacturing procedure into two stages – generic and customization stages (Li & Tang, 1997; Skipworth & Harrison, 2004). Taking a microwave radio product for example, the microwave units and filters (generic goods) are made at the generic stage, while the radio frequencies (customized goods) are specified at the customization stage (Olhager, 2010). Researchers have proved the effectiveness and practicability of DPD for traditional manufacturing operations (e.g., He, Kusiak, & Tseng, 1998; Song & Kusiak, 2010; Su, Chang, Ferguson, & Ho, 2010; Trentin, Salvador, Forza, & Rungtusanatham, 2011), but so far few has studied the application of DPD to CMfg.

This paper focuses on the formulation of delayed product differentiation for cloud manufacturing (DPDCM). According to the main characteristics of CMfg compared to the traditional manufacturing patterns, DPDCM, on one hand, must be universal enough to fit various types of manufacturing firms (either OEMs or suppliers). On the other hand, it should be flexible enough to satisfy the agile supply chain. Therefore, DPDCM is formulated on the basis of the following key decision strategies.

1. *Order-release decision*: The size of order-release influences the economic lot sizing of the customization stage. It is one of important factors for optimal production. Most current DPD models tend to force the orders to be released one after another at customization stage (e.g., Gupta & Benjaafar, 2004; Su, Chang, & Ferguson, 2005),

* Corresponding author.
E-mail address: mejyj@zju.edu.cn (Y. Ji).

https://doi.org/10.1016/j.cie.2018.01.019
Received 6 January 2017; Received in revised form 7 November 2017; Accepted 22 January 2018

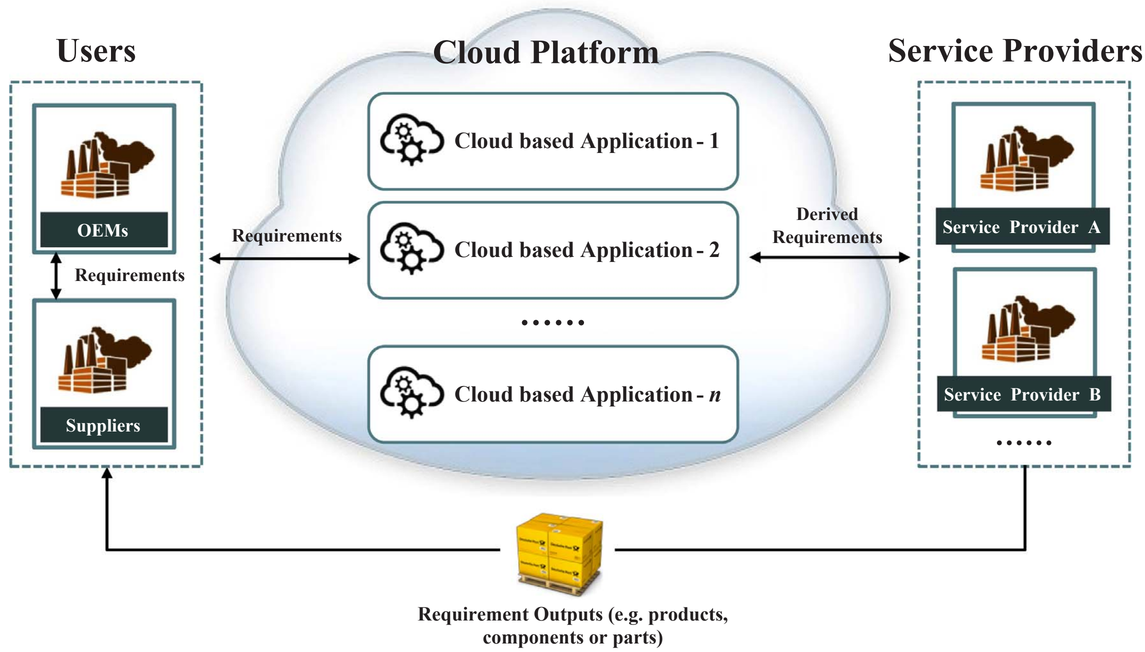

Fig. 1. Supply chain of cloud manufacturing (Wu et al., 2013).

which are fit for the pattern of single-piece production (SP), but fail to satisfy the manufacturing firms inclining to adopt batch production (BP) pattern (e.g., Yu, Ji, Qi, Gu, & Tao, 2015). Hence, to be applicable to multiple manufacturing firms regardless of their applied production patterns, DPDCM should integrate both SP and BP patterns at customization stage and offer the optimal order-release size.

2. *Generic inventory decision:* Manufacturing firms, even the same manufacturer at different manufacturing periods, may confront orders with various due date constraints (e.g., urgent or normal orders), restricting the manufacturing lead times. To universalize the DPDCM to satisfy any due date constraint, using a constantly deterministic generic inventory filled with work-in-process (WIP) items as most traditional models do (e.g., Caux, David, & Pierreval, 2006) becomes inadequate. For urgent orders, it might be better to eligibly stock some additional WIP items than to simply raise the inventory level. For example, storing some already colored parts (e.g., green, red and blue parts) in generic inventory, instead of the generic un-colored parts that needs further colorin, can save the coloring time at customization stage to meet the urgence. Hence, the feasibility of the adjustable generic inventory in DPDCM is studied.
3. *Sourcing decision:* Thanks to the agility enabled by CMfg, manufacturing firms are able to seek or change supply chain partners with a much more convenient manner than ever before. DPDCM tends to increase the flexibility of sourcing decision, helping the manufacturers to optimally determine what to make or not (purchasing from the suppliers) according to their real time production conditions. The validity of such strategy carried out in DPDCM is analyzed.

In conclusion, DPDCM integrates these three decision strategies to ensure the universality and flexibility of DPD applied to CMfg environment. A genetic algorithm (GA) is implemented to seek for near-optimal solutions. The rest of the paper is organized as follows. Section 2 reviews the related literature. DPDCM is explained in Sections 3 and 4. Case studies of bicycle and industrial-transformer manufacturing are given in Section 5, followed by the relevant discussion in Section 6. Section 7 concludes the paper and suggests future work.

2. Literature review

2.1. Cloud manufacturing

Manufacturing is enabled by information and computer technologies (ICT). CMfg is a relatively new manufacturing concept driven by cloud computing (Xu, 2012). Compared to traditional ICT-enabled manufacturing technologies, such as computer integrated manufacturing (CIM) and distributed manufacturing (DM), the distinguished characteristic of CMfg is service-oriented, which virtualizes the manufacturing resources as consumable services (Adamson, Wang, & Holm, 2013; Li et al., 2010; Wu et al., 2013; Xu, 2012). According to this feature, Xu (2012) defines CMfg as "*a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable manufacturing resources* (e.g. *manufacturing software tools, manufacturing equipment, and manufacturing capabilities) that can be rapidly provisioned and released with minimal management effort or service provider interaction*".

By far most researchers focus on the formulation of CMfg architecture, platform, model, and framework, attempting to achieve a functional system (e.g. Wang & Xu, 2013; Lu, Xu, & Xu, 2014). Plenty of service-oriented systems with various layer structures are proposed for different requirements and circumstances, such as the 3-layer (e.g. Liu & Jiang, 2012), 4-layer (e.g. Xu, 2012), 5-layer (e.g. Li, Hu, Wang, & Zhu, 2011) and 6-layer (e.g. Xiang & Hu, 2012) systems.

Moreover, some studies are more centered on the service paradigm of CMfg to lead to advanced service-oriented manufacturing, such as the research of service encapsulation and combination (e.g. Ding, Yu, & Sun, 2012; Zhang, Zhang, Liu, & Hu, 2015; Chen et al., 2016) and service planning and scheduling (e.g. Laili et al., 2013; Yu, Zhang, Xu, Ji, & Yu, 2015) for CMfg. For details, please refer to the recent reviews, including Adamson et al. (2013) and Wu et al. (2013).

To sum up, most articles mainly pay attention to the high-level operations of CMfg (i.e. service-oriented architecture and management), while few concerns the low-level operations, such as the production level, representing the execution and realization of the manufacturing services. This paper aims to fill the gap, by proposing DPDCM.

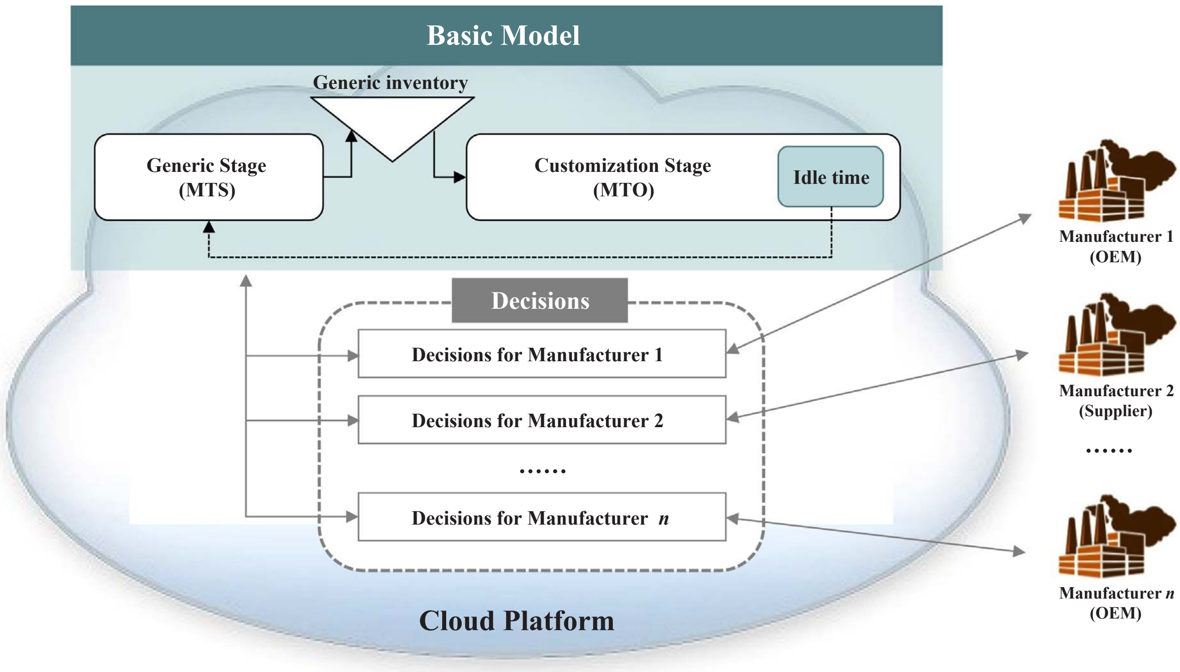

Fig. 2. Delayed product differentiation for cloud manufacturing.

2.2. Delayed product differentiation

DPD defers the customized forms and functional utilities of a product (Van Hoek, 2001). It has been successfully implemented by multiple well-known companies, such as Dell and Hewlett-Packard (Feitzinger & Lee, 1997; Magretta, 1998). The fundamental idea of DPD is to cope with the high demand uncertainty, impacted by escalating product variety, while ensuring low operational costs and short manufacturing lead time. The key is to find the optimal differentiation point – the timing to specialize the generic inventory into the customized goods (Forza, Salvador, & Trentin, 2008). Many researchers engage on the provision of mathematics models of DPD in order to achieve an optimal production. Some attempt to formulate it as queuing models (e.g. Jewkes & Alfa, 2009; Zhou, Zhang, & Zhou, 2013), while some adopt dynamic programming models for such purpose (e.g. Hsu & Wang, 2004; Chen, Kang, & Lee, 2006). This research stream is usually centered on the brief problem modeling for quick solutions, thus the proposed models are not comprehensive enough to meet the complexity of CMfg environment.

To make further efforts and consider the complexity of the problem, the following three factors are frequently paid attention to (Dekok & Graves, 2003), including:

- *Market factor* is subsequent to demand fluctuations and due date restrictions. For example, Tang (2011) studies the inventory management in a capacitated DPD problem, considering the influence of demand forecast on base-stock inventory models.
- *Product factor* in essence relates to the product design, resulting from the modularity and commonality design. In this aspect, He et al. (1998) suggest three product design rules for the optimal planning of DPD (e.g. minimum differentiation and manufacturing costs). Moreover, Song and Kusiak (2010) present a framework to decide the optimal design of WIP, with two competing objectives, i.e., the mean value of assembly operations and expected pre-assembly cost.
- *Process factor* contains the impacts of operation sequence, stock selection, supply chain structure, etc. For instance, to find the optimal configurations and inventory levels of WIP of vanilla boxes, Swaminathan and Tayur (1998) model the problem as a two-stage integer programming with recourse.

Based on these factors, a close relationship can be drawn between the product and process factors, e.g., the change of product design may influence the relevant manufacturing process. Lee and Tang (1997) formalize a simple model to evaluate three DPD redesign approaches, namely standardization, modular design (product factor), and process restructuring (process factor). Algeddawy and Elmaraghy (2010) propose an assembly line layout design model, analyzing product commonality (product factor) and assembly sequences (process factor). Furthermore, applying median-joining phylogenetic networks, Hanafy and Elmaraghy (2015) recommend another DPD model to develop product platforms (product factors) and assembly line layout of modular product families (process factors).

These proposed DPD models are mainly designed for the dominant companies in a supply chain of which the network structure is fixed. Hence, they cannot satisfy the CMfg environment that requires more flexible and agile policies for multiple-company management. More universal and flexible models are demanded for CMfg. This paper works on one of such models – DPDCM. Compared to conventional approaches, DPDCM considers multiple factors impacted on the optimality of DPD, as it integrates the order-release, generic inventory and sourcing decisions in a model to offer comprehensive result. It implies the consideration of market, product and process factors. In specific, to response to the dynamicity of market demand, DPDCM can meet both requirements of SP and BP. The product factor is represented by the convenient determination of WIP, i.e. designing into a semi-finished one or a bunch of varied finished ones, although the engineering design factors are excluded. Sourcing decision (process factor) takes advantage of the agile supply chain of CMfg to ensure the flexibility of the solution. According to these features, DPDCM would be more universal and flexible than traditional methods.

3. Decision strategies

As shown in Fig. 2, DPDCM is built upon a two-stage production system, including generic and customization stages. The generic stage is carried out on a make-to-stock (MTS) basis, generating a portion of generic inventory (the rest portion can be sourced from suppliers). The customization stage is on a make-to-order (MTO) basis, specializing the generic inventory into the finished goods (customized goods). The

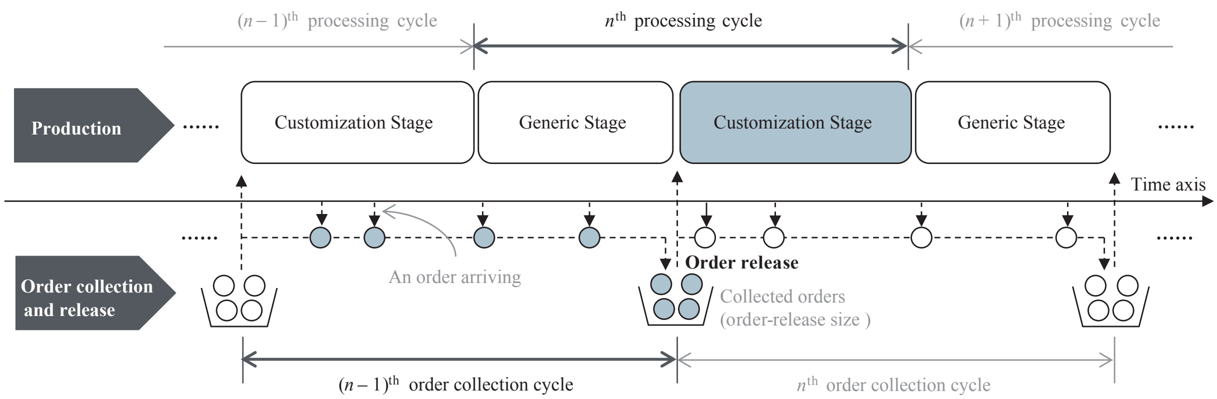

Fig. 3. Order collection cycle vs. processing cycle.

fundamental feature of DPDCM is to utilize the manufacturing idle, appearing at customization stage due to the demand fluctuations, as the trigger to arouse the generic stage to prevent aimless production. Based on it, the order-release, generic inventory and sourcing decisions are formulated to meet various production conditions of CMfg manufacturers. In addition, to restrict the research scope, we only consider the processing procedure rather than assembling.

3.1. Order-release decision

Order-release decision is to determine the optimal order-release size according to the manufacturers' production conditions. Once the order-release size is determined, the production pattern can be confirmed as well. The SP's order-release size is always equal to one, while the BP's order-release size is larger than one. Obviously, the determination of the order-release size reflects the integration of both SP and BP patterns.

Order-release size is commonly computed by the multiple of demand rate and order collection cycle time. An order collection cycle is defined as the period between two adjacent order-release activities (as Fig. 3 shows), of which the length is equal to the processing cycle to guarantee the succession of production (refer to Fig. 3). Therefore, the order-release size is represented as $Q = \lambda T$ (Qi, 2013). Q is the order-release size, equal to the order quantity in a processing cycle. λ represents the constant customer annual demand rate, and T is the processing cycle time.

3.2. Generic inventory decision

The aim of generic inventory decision is to find out the optimal composition of WIP inventory. To do so, the WIP items are classified into two groups, i.e., *common* and *variant*, according to the product development standards – product family (Simpson, Siddique, & Jiao, 2006). A *common* is the same item shared by some or all products of a family. A *variant* offers an identical function among some or all products of a family, but the final forms, e.g. size and color, may be slightly different from each other. Hence, the determination of a *variant* is usually much more complex than a *common*, and correspondingly, the *variant* tends to have more impact on the control of inventory costs and manufacturing lead time than a *common*. To this end, the generic inventory decision is mainly dependent on the selection of *variant*.

In details, as the varied items (VIs) of a *variant* (finished items) are usually similar with each other, they can be all derived from a single semi-finished *variant* (SFV). The generic inventory decision is, therefore, mainly dependent on the determination of the storage scheme for *variants*, i.e. storing a SFV or its overall VIs. For instance, Fig. 4 presents a result of the generic inventory decision as well as the taxonomy of WIP. As it shows, the finished goods is assembled by six WIP items, consisting of four *variants* and two *commons*. Among the *variants* stocked in generic inventory, V_1 and V_4 are in the form of VIs, while the rest are SFVs.

It is obvious that stocking SFV is good for controlling inventory costs, while storing VIs can reduce the manufacturing lead time. Due to the trade-off between them, DPDCM should determine the optimal composition of SFV and VIs according to the manufacturer's production condition (e.g., urgent or ordinary period). β_n (0–1 binary variable) and U_n are used to formulate the decision as $U_n(1-\beta_n)$. β_n reflects the stocking items associated with the $n^{-\text{th}}$ *variant*, i.e., SFV or all its VIs. If the SFV is chosen to store, $\beta_n = 1$, otherwise $\beta_n = 0$. U_n is the total quantity of VIs of the $n^{-\text{th}}$ *variant*.

3.3. Sourcing decision

Because of the agile supply chain and almost unlimited manufacturing capabilities of CMfg, sourcing has become convenient for the manufacturing firms to replenish the generic inventory, and more items are tended to source from suppliers rather than self-production. It can ensure the competitiveness, as the manufacturer can pay more attention to the high value-added activities at the customization stage. Under this circumstance, the generic stage is mainly carried out for the use of manufacturing idle, and regarded as the buffer to adjust the trade-off between inventory cost and manufacturing lead time, thus it won not last too long. For this reason, the SFV, rather than the *common* and VIs, is the only WIP required to produce at the generic stage, as SFV is critical that influences the production of customization stage. In addition, as the manufacturing assembly is not considered, while the consumption of *common* and VIs is usually happened in assembly, it won not affect the inventory costs in DPDCM. Therefore, the sourcing decision only concerns the arrangement of SFVs.

Based on this, the sourcing strategy needs to decide the optimal proportion of purchased and produced SFVs, as well as the timing to place a sourcing order. Q^0 is used to represent the order quantity of the sourced SFV in a processing cycle, while $Q-Q^0$ implies its production quantity at generic stage. The sourcing timing depends on the way of generic inventory management, and a lot size-reorder point (LSRP) system is adopted, as the system is commonly used in commercial environment (Nahmias, 2001). LSRP is a continuous review system, of which the on-hand inventory level is always accessible to decision-maker. According to this mechanism, the sourcing order is placed when the generic inventory level reaches the reorder point R.

4. Model formulation

To acquire an integrated optimal solution for order-release, generic inventory and sourcing decisions, DPDCM is formulated based on the trade-off between processing cycle time and generic inventory costs. The following notations are used throughout the paper.

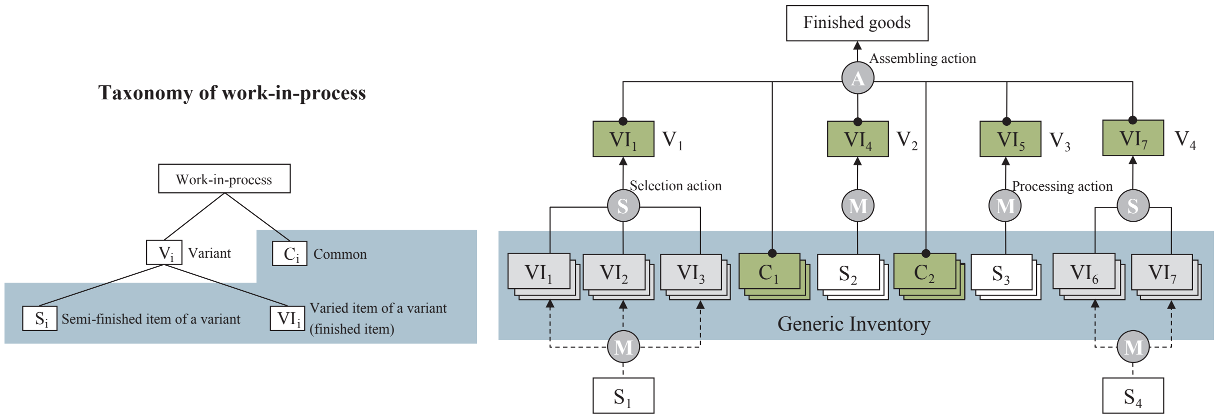

Fig. 4. An example of the result of generic inventory decision.

T	processing cycle time
λ	constant customer annual demand rate
t^s	length of a generic stage in a processing cycle
t^m	length of a customization stage in a processing cycle
Q	order-release size
Q^0	order quantity of sourced SFV in a processing cycle
R	SFV inventory reorder point
τ	SFV inventory reorder lead time
x^τ	order quantity during period τ
T^u	expected annual manufacturing idle time
φ	working-hour coefficient
$E(Q^s)$	expected quantity of common in an order
N	quantity of variant relating to an order
a_n^s	processing time of the $n^{-\text{th}}$ SFV, $n = 1,2,\ldots,N$
b_n^s	setup time of the $n^{-\text{th}}$ SFV
a_n^m	average processing time from the $n^{-\text{th}}$ SFV to the corresponding VIs
b_n^m	average setup time from the $n^{-\text{th}}$ SFV to the corresponding VIs
β_n	binary variable, implying whether the $n^{-\text{th}}$ SFV or its VIs is stored in generic inventory
U_n	quantity of VIs can be derived from $n^{-\text{th}}$ SFV, $U_n \geqslant 1$
$E(P_n)$	expected demand of the $n^{-\text{th}}$ SFV in an order
$\overline{F}$	average SFV inventory level per processing cycle
h	holding cost of per unit per year
s	expected on-hand SFV inventory level before an order arrives
p	stock-out cost of per unit of unsatisfied demand
c	ordering cost of per unit of generic inventory
k	setup cost of inventory per processing cycle
C^{hm}	inventory holding cost of the SFV per year
C^{hs}	inventory holding cost of the common and VIs per year
C^h	total inventory holding cost per year
C^p	inventory penalty cost per year
C^o	inventory ordering cost per year
C^s	inventory setup cost per year
TC	total generic inventory cost per year
$T^{\max}$	maximum processing cycle time

4.1. Processing cycle time

Processing cycle time is the total production length, consisting of a generic and a customization stage (see Fig. 3). The length of a generic stage is represented as φt^s, where

$$t^s = \begin{cases} (Q-Q^0)\sum_{n=1}^{N} E(P_n)\beta_n a_n^s + \sum_{n=1}^{N} E(P_n)\beta_n b_n^s, & Q > Q^0 \\ 0, & Q = Q^0 \end{cases} \tag{1}$$

In Eq. (1), $Q = Q^0$ expresses that all SFVs are purchased from suppliers (hence, $t^s = 0$). $Q > Q^0$ implies a portion of the released orders, of which the SFV inventory needs to be sourced. The setup time is only computed once to reflect the application of batch production. Similarly, the length of a customization stage is φt^m, where

$$t^m = Q\sum_{n=1}^{N} E(P_n)\beta_n a_n^m + \sum_{n=1}^{N} E(P_n)\beta_n b_n^m \tag{2}$$

Furthermore, the interaction between the length of generic stage and manufacturing idle time should be considered, as it influences the processing cycle time. Generally, the appearance of manufacturing idle is due to various factors, such as demand fluctuation, equipment, and process alterations. It is too complicated to quantify and interpret it in a consistent manner (Kirkley, Paul, & Squires, 2002). Hence, for the purpose of simplification, the manufacturing idle time is defined as an expected value (T^u), sampled from the real production environment, and the value of a processing cycle is denoted as QT^u/λ. Thus, the processing cycle time is formulated as

$$\frac{T}{\varphi} = \begin{cases} t^s + t^m, & t^s > QT^u/\lambda \\ QT^u/\lambda + t^m, & t^s \leqslant QT^u/\lambda \end{cases} \tag{3}$$

In Eq. (3), $t^s > QT^u/\lambda$ means no manufacturing idle, i.e., the generic stage making full use of the manufacturing idle. Hence, the processing cycle time is computed by $t^s + t^m$. Otherwise (i.e. $t^s \leqslant QT^u/\lambda$), it should be computed by $QT^u/\lambda + t^m$, representing that the manufacturing idle is still existed. In addition, the constraint of maximum processing cycle time must be taken into account (i.e. $T < T^{\max}$), because no customer is willing to wait much.

4.2. Generic inventory costs

DPDCM considers the following generic inventory costs in a processing cycle: holding cost (h per unit held per year), penalty cost (p per unit of unsatisfied demand), ordering cost (c per unit) and setup cost (k per processing cycle).

4.2.1. Holding cost

Holding cost is the overall cost incurred due to on-hand inventory. It can be formulated according to its proportional relationship with the average inventory level. Meanwhile, as DPDCM applies different sourcing strategies for SFV and other inventory, the holding cost should be computed separately.

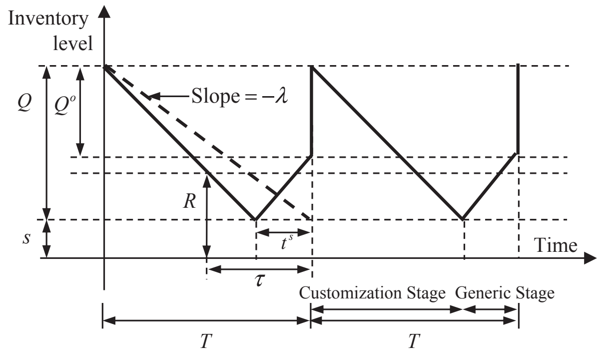

Fig. 5. Expected inventory level of SFV.

Table 1
Setup and processing times of SFVs and VIs of the bicycle (day).

n	Variant name	SFVs		VIs	
		a_n^s	b_n^s	a_n^m	b_n^m
1	Head tube	0.0091	0.0183	0.0183	0.0274
2	Seat tube	0.0183	0.0365	0.0274	0.0456
3	Top tube	0.0274	0.0548	0.0365	0.0639
4	Down tube	0.0365	0.0730	0.0456	0.0821
5	Stay	0.0456	0.0913	0.0548	0.1003
6	Fork	0.0548	0.1095	0.0639	0.1186
7	Crank arm	0.0639	0.1278	0.0730	0.1369
8	Rim	0.0730	0.1460	0.0821	0.1551
9	Chain rings	0.0822	0.1643	0.0912	0.1734
10	Handlebar grip	0.0912	0.1825	0.1003	0.1916

- *Holding cost of SFV.* The inventory level of SFV is expected to vary linearly between s and $Q+s$, where s is the safety stock, $s = R-\lambda\tau$ (Nahmias, 2001, Chapter 5.4), and $t^s \leqslant \tau \leqslant T$. The curve is shown in Fig. 5, increasing at the generic stage (inventory replenishment) and declining at the customization stage (inventory consumption). The average inventory level can be estimated as the average of the curve function,

$$\overline{F} = \frac{(Q-Q^0)t^s}{2T} + \frac{Q(T-t^s)}{2T} + s \tag{4}$$

In Eq. (4), the first and middle terms at the right side of the equal reflect the average values of inventory level at the generic and customization stages respectively, and t^m is replaced with $T-t^s$. Furthermore, substituting T and s with Q/λ and $R-\lambda\tau$ respectively, Eq. (4) is transformed into

$$\overline{F} = \frac{1}{2}\left(Q-\frac{\lambda t^s Q^0}{Q}\right) + R-\lambda\tau \tag{5}$$

Finally, considering the inventory composition (β_n), the holding cost of SFV can be expressed as

$$C^{hm} = \overline{F}h\sum_{n=1}^{N}\beta_n E(P_n) \tag{6}$$

- *Holding cost of common and VIs.* The inventory of *common* and VIs is usually consumed at the assembly procedure. Hence, their inventory levels are constantly equal to Q during the entire processing cycle, and the holding cost can be computed as

Table 3
Setup and processing times of each industrial-transformer component and the number of VIs.

n	Variant name	SFV manufacturing		VIs specialization from SFV		
		a_n^s (day)	b_n^s (day)	a_n^m (day)	b_n^m (day)	U_n (unit)
1	Insulator body	0.6	0.6	0.5	0.5	9
2	Metal body	0.1	0.1	0.3	0.3	9
3	Tank & accessories	0.1	0.1	0.5	0.5	3
4	Lead wire	0.15	0.15	0.1	0.1	5
5	Flashboard	0.05	0.05	0.1	0.1	5
6	Windings	0.3	0.3	0.2	0.2	9
7	Radiator	0	0	0.2	0.2	3
8	Iron core	0.05	0.05	0.4	0.4	9
9	Oiling	0	0	0.15	0.15	1

$$C^{hs} = hQ\left[E(Q^s) + \sum_{n=1}^{N}U_n(1-\beta_n)E(P_n)\right] \tag{7}$$

Therefore, according to the formulation of Eq. (6) and Eq. (7), the overall holding cost is formulated as

$$C^h = C^{hm} + C^{hs} \tag{8}$$

4.2.2. Penalty cost

Penalty cost is caused due to failing to satisfy demands. It is determined by the quantity of stock-outs during a processing cycle, i.e.,

$$n(R) = \int_R^{\infty}(x^\tau-R)f(x^\tau)dx \tag{9}$$

It is an expected value, and the value per unit time can be represented as $n(R)/(T-t^s)$, as in DPDCM the stock-out usually happens at the customization stage. Hence, the penalty cost can be formulated as Eq. (10), in which T is substituted with Q/λ.

$$C^p = \frac{pn(R)}{T-t^s} = \frac{p\lambda n(R)}{Q-\lambda t^s} \tag{10}$$

To compute $n(R)$, the customer demand is assumed to be normally distributed (Nahmias, 2001), of which the probability density function is

$$f(x^\tau) = \frac{1}{\sigma\sqrt{2\pi}}\exp\left[-\frac{1}{2}\left(\frac{x^\tau-\mu}{\sigma}\right)^2\right], \quad \text{where } -\infty < x^\tau < +\infty \tag{11}$$

According to the properties of normal distribution, $n(R)$ can be computed using the standardized loss function $L(z)$ defined as

$$L(z) = \int_z^{\infty}(t-z)\varphi(t)dt, \quad \text{where } \varphi(t) = \frac{1}{\sqrt{2\pi}}e^{-t^2/2} \tag{12}$$

$\varphi(t)$ in Eq. (12) is the standard normal probability density function. Equivalently,

Table 2
Near-optimal solutions of bicycle production.

Subject to	QT^u/λ	t^s	Q	Q^0	R	β_6	β_7	β_8	β_9	β_{10}	φ	TC
$t^s \geqslant QT^u/\lambda$	0.013	0.021	13	0	9	1	1	1	1	1	2	70.5544
$0 < t^s < QT^u/\lambda$	0.015	0.003	15	7	4	1	1	1	1	1	3	86.2515
$t^s = 0$	0.006	0	6	6	4	1	1	1	1	1	3	159.869

Table 4
Near-optimal solutions of industrial-transformer production.

Subject to	t^s	Q	Q^0	R	β_1	β_2	β_3	β_4	β_5	β_6	β_7	β_8	β_9	φ	TC
$t^s \geqslant 0.007$	0.0270	14	7	14	1	1	0	1	1	1	0	1	0	1	570.2555
$0 < t^s < 0.007$	0.0055	14	8	14	0	1	0	1	1	0	1	1	0	2	649.4205
$\lambda = 120$	0	14	14	13	1	1	0	0	1	1	0	0	1	2	649.0822

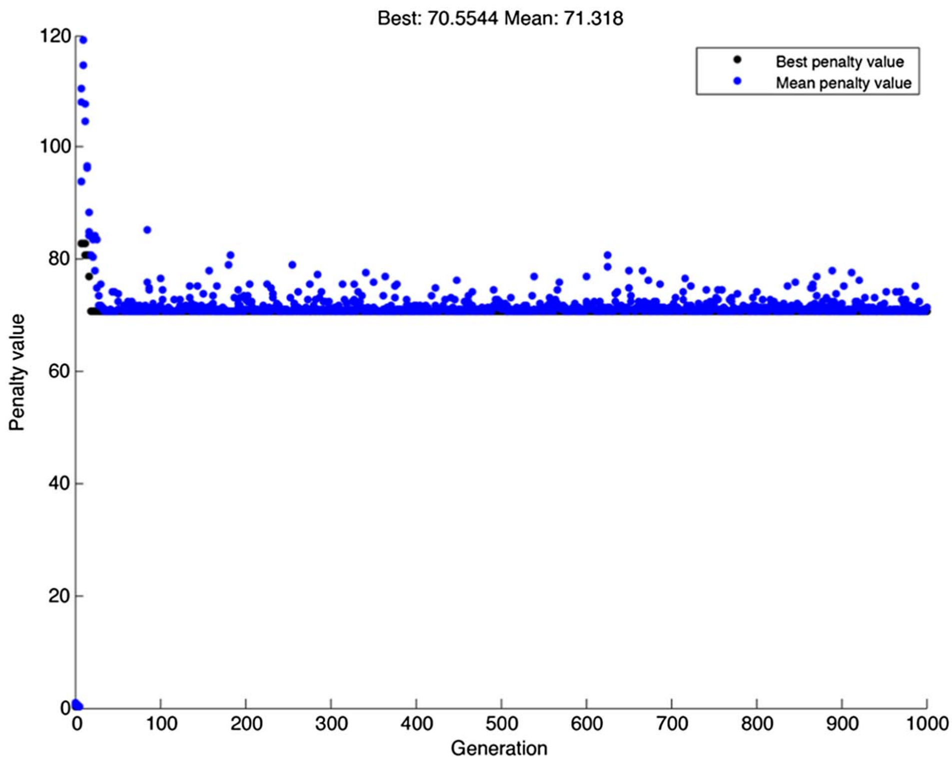

Fig. 6. the convergence of GA for near-optimal bicycle manufacturing.

$$\begin{aligned} L(z) &= \frac{1}{\sqrt{2\pi}}\int_z^{\infty}(t-z)e^{-t^2/2}dt \\ &= \frac{1}{\sqrt{2\pi}}\int_z^{\infty}te^{-t^2/2}dt-\frac{z}{\sqrt{2\pi}}\int_z^{\infty}e^{-t^2/2}dt \\ &= -\frac{1}{\sqrt{2\pi}}e^{-t^2/2}|_z^{\infty}-z[1-F(z)] \\ &= \frac{1}{\sqrt{2\pi}}e^{-z^2/2}-z[1-F(z)] \end{aligned} \tag{13}$$

In Eq. (13), $F(z)$ is the distribution function of $\varphi(t)$. Finally, as the standardized variable z is equal to $(R-\mu)/\sigma$, $n(R)$ can be computed as

$$n(R) = \sigma L(z) = \sigma L\left(\frac{R-\mu}{\sigma}\right) \tag{14}$$

4.2.3. Ordering and setup costs

Ordering cost depends on the amount of units purchased from suppliers. As only the ordering of SFV is considered in DPDCM, it can be formulated as:

$$C^o = \frac{c\lambda Q^0\sum_{n=1}^{N}\beta_n E(P_n)}{Q} \tag{15}$$

In Eq. (15), Q^0/Q represents the proportion of SFV purchased from suppliers.

Setup cost is incurred once in a processing cycle. It is inversely proportional to the processing cycle time, thus can be represented as

$$C^s = k/T = k\lambda/Q \tag{16}$$

4.3. Cost function and constraints

The overall generic inventory costs is the sum of holding, penalty, ordering and setup costs, i.e., $TC = C^h + C^p + C^o + C^s$. The objective of DPDCM is to minimize TC to make the optimal decision for order-release, generic inventory selection and sourcing. It is formulated as

$$\min_{R,Q,\beta_n,\varphi,Q^0}(C^h + C^p + C^o + C^s) \tag{17}$$

subject to

$$Q = \begin{cases} \dfrac{\lambda\varphi[\sum_{n=1}^{N}E(P_n)\beta_n b_n^m + t^s]}{1-\lambda\sum_{n=1}^{N}E(P_n)\beta_n a_n^m}, & t^s > QT^u/\lambda \\ \dfrac{\lambda\varphi\sum_{n=1}^{N}E(P_n)\beta_n b_n^m}{1-T^u-\lambda\sum_{n=1}^{N}E(P_n)\beta_n a_n^m}, & t^s \leqslant QT^u/\lambda \end{cases} \tag{18}$$

$$t^s = \begin{cases} (Q-Q^0)\displaystyle\sum_{n=1}^{N}E(P_n)\beta_n a_n^s + \sum_{n=1}^{N}E(P_n)\beta_n b_n^s, & Q > Q^0 \\ 0, & Q = Q^0 \end{cases} \tag{19}$$

$$0 \leqslant s \leqslant R \tag{20}$$

$$t^s \leqslant \tau \leqslant \frac{Q}{\lambda} \tag{21}$$

$$Q \leqslant \lambda T^{\max} \tag{22}$$

As it shows, DPDCM is formulated as an integer programming problem, and the decision variables are R, Q, β_n, φ, Q^o. Eq. (17) is the objective function. Eq. (18) is the transformation of Eq. (3), where T

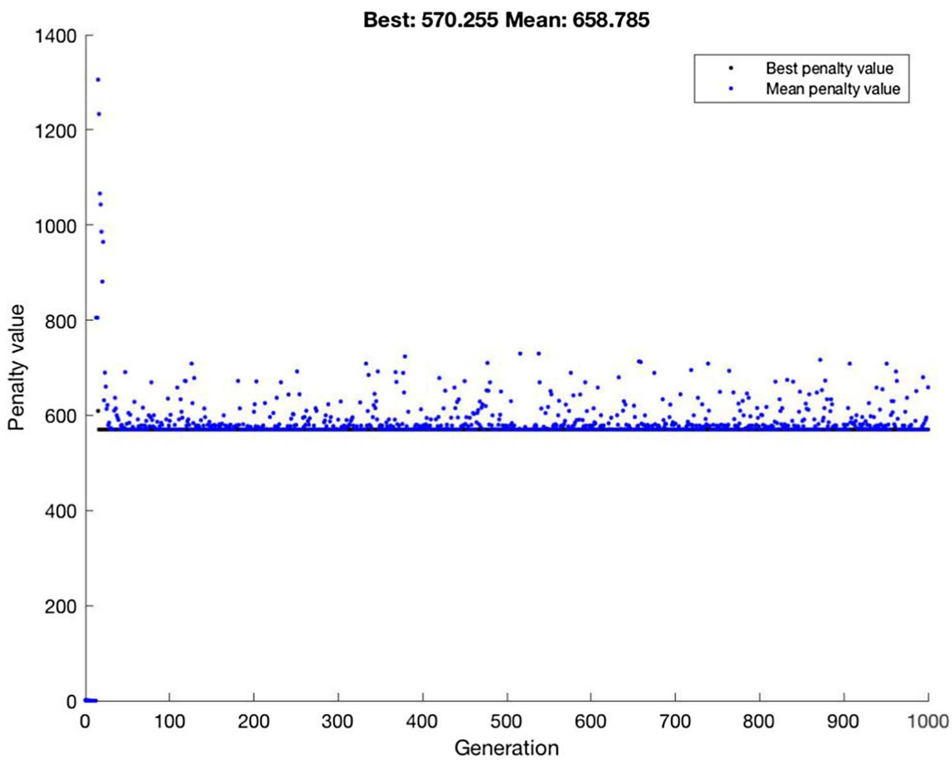

Fig. 7. the convergence of GA for near-optimal industrial-transformer manufacturing.

and t^m are replaced with Q/λ and $Q\sum_{n=1}^{N} E(P_n)\beta_n a_n^m + \sum_{n=1}^{N} E(P_n)\beta_n b_n^m$, respectively. The length of generic stage within a processing cycle is represented by Eq. (19). Eq. (20) requires the safety stock to be smaller than the reorder point. Eq. (21) represents the constraint that the reorder lead time should be longer than the length of generic stage, but shorter than the processing cycle time. The restriction of maximum processing cycle time is defined by Eq. (22).

5. Case study

Two cases are applied to validate DPDCM. One is a bicycle manufacturer offering customized products for high-end market, in which the bicycles are specified according to personal heights, weights and preferences (e.g. favorite colors and printings). The bicycle is mainly made up of seventeen components (WIP): seven *commons* and ten *variants*. Among the *variants*, five (n from 1 to 5) have infinite variations, such as the component that permits the customers to draw their own printings on it. For this kind of *variant*, only the SFVs are stored in the generic inventory. By contrary, the variations of the rest *variants* (n from 6 to 10) are finite, and each of them is required to have three VIs ($U_6 = U_7 = ...=U_{10} = 3$). These VIs can be stocked instead of their SFVs. Therefore, only the elements of $\{\beta_n|6 \leqslant n \leqslant 10\}$ are the decision valuables, while those of $\{\beta_n|1 \leqslant n \leqslant 5\}$ are constantly equal to one. Table 1 shows the average setup and processing times of SFVs and VIs.

The other conditions of the manufacturer are listed as follows. The average annual demand rate is 100 bicycles ($\lambda = 100$), and the expected demand of each SFV in an order is equal to one, i.e., $E(P_1) = E(P_2) = ...=E(P_{10}) = 1$. According to historical data statistics, the average annual manufacturing idle time $T^u = 0.1$ year, and the processing cycle time should be less than 0.16 year ($T^{\max} = 0.16$). The delivery lead time from suppliers is 0.04 year ($\tau = 0.04$). There are averagely coming 4 bicycles ($\mu = 4$) during this period, and the standard deviation is 2 ($\sigma = 2$). The setup cost $K = \$5$ per processing cycle. The holding cost is \$0.02 per unit per year ($h = \$0.02$). The ordering cost $c = \$0.04$ per unit. The stock-out cost $p = \$0.6$ per unit of unsatisfied demand. The working-hour can set as 8, 16 or 24 h that correspond to $\varphi = 1$, 2 and 3 respectively.

A genetic algorithm (GA) is used to solve the problem, as it can conveniently encode the integer variables of DPDCM. In addition, as the purpose of case studies is to verify the universality and flexibility of DPDCM, a near-optimal solution provided by a GA is sufficient. Therefore, the *gaoptimset* function in *MATLAB Global Optimization Toolbox* – one GA application – is adopted. The relevant parameters are set as default, i.e., *opts = gaoptimset ('StallGenLimit', inf, 'TolFun', 1e-10, 'PopulationSize', 30, 'TolCon', 1e−10, 'Generations', 1000, 'PlotFcns', @ gaplotbestf)*, defining the population size as 30, and the number of generations as 1000. The crossover and mutation options are set with the default value, as the *gaoptimset* function do not recommend to use crossover and mutation functions with integer problem (referring to the *MATLAB documentation*). The GA algorithm is run by the function that *[x, fval, exitflag] = ga (@hfpfitness, 14, [ ], [ ], [ ], [ ], lb, ub, @hfpconst, IntCon, opts)*, in which "*@hfpfitness*" and "*@hfpconst*" are the objective and constraint functions of the problem respectively, "*[ ]*" set the default option, "*14*" means the number of the decision variables, "*lb*" and "*ub*" define the lower and upper limits of the decision variables respectively, "*IntCon*" implies the integer decision variables, and "*opts*" is the parameters representing the GA characteristics.

Table 2 shows the near-optimal solution where $TC = 70.5544$. The sensible thing to do is to set the order-release size as 13. It validates the feasibility of applying batch production in DPDCM. Furthermore, the near-optimal solution is acquired under the restriction of $t^s \geqslant QT^u/\lambda$, which proves the validity of utilizing the manufacturing idle to trigger the generic stage. In addition, no VIs is required to hold in generic inventory, as $\beta_6 = \beta_7 = ...=\beta_{10} = 1$. It demonstrates that DPDCM satisfies the common feature of DPD, i.e., delaying the commitment of differentiation as much as possible.

Apart from the bicycle manufacturing, to study the robustness of DPDCM, we extend the case study to another complex industrial product, namely industrial-transformer, requiring much more manufacturing time. The corresponding setup and processing times are shown in Table 3, and the rest conditions are as follows: $\lambda = 120$, $E(P_1) = E(P_2) = ...=E(P_9) = 1$, $T^u = 0.06$, $T^{\max} = 0.12$, $\tau = 0.04$, $\mu = 5$, $\sigma = 3$, $K = \$60$, $h = \$0.04$, $c = \$0.03$ and $p = \$5$.

The results are presented in Table 4. The near-optimal solution is

Table 5
Near-optimal solution of the bicycle case when $c = \$0.004$.

QT^u/λ	t^s	Q	Q^0	R	β_6	β_7	β_8	β_9	β_{10}	φ	TC
0.015	0.0165	15	5	9	1	1	1	1	1	2	41.9098

Table 6
Near-optimal solution of the bicycle case in which all a_n^m and b_n^m increased by 10 times.

QT^u/λ	t^s	Q	Q^0	R	β_6	β_7	β_8	β_9	β_{10}	φ	TC
0.009	0.066	12	0	7	0	0	0	0	1	1	124.493

Table 7
Near-optimal solutions with the vary of demand rates of the bicycle case.

λ	μ	σ	QT^u/λ	t^s	Q	Q^0	R	β_6	β_7	β_8	β_9	β_{10}	TC
10	0	1	0.0100	0.0041	1	0	0	1	1	1	1	1	57.5808
20	1	1	0.0050	0.0041	1	0	1	1	1	1	1	1	110.667
25	1	1	0.0040	0.0041	1	0	1	1	1	1	1	1	115.599
30	1	1	0.0033	0.0055	2	0	2	1	1	1	1	1	85.8387
40	2	1	0.0050	0.0055	2	0	2	1	1	1	1	1	116.547
50	2	1	0.0060	0.0083	4	0	4	1	1	1	1	1	77.7919
100	4	2	0.0130	0.0193	13	0	9	1	1	1	1	1	70.5544
150	6	2	0.0127	0.0289	19	0	11	1	1	1	1	1	87.0705
200	8	3	0.0125	0.0371	25	0	16	1	1	1	1	1	103.659
250	10	4	0.0132	0.0385	33	7	20	1	1	1	1	1	139.050
300	12	4	0.0160	0.0399	48	21	22	1	1	1	1	1	180.957

acquired, subject to $t^s \geqslant QT^u/\lambda$ ($t^s \geqslant 0.007$) as well, which is in accordance with the bicycle case. It again proves the validity of the DPDCM, in terms of triggering the generic stage by making use of the manufacturing idle. Fig. 6 and Fig. 7 show the convergences of GA for both cases, in which the penalty value represents the value of TC.

6. Discussion

The solutions shown in Tables 2 and 4 report a part of attributes of DPDCM. It needs further analysis on the universality and flexibility of DPDCM that satisfy the CMfg requirements. For this purpose, the two cases are compared for further discussion. In addition, some of the initial conditions are changed to simulate different scenarios of CMfg to study the impact of various factors on optimal solutions.

6.1. Impact of production rate and ordering cost

The results shown in Tables 2, 4 and 5 reflect that the changes in production rate and ordering cost influence the sourcing and generic inventory decisions.

- *Sourcing decision.* It can be seen that the production rate of a single product indeed influences the result of sourcing decision. Longer manufacturing time (e.g. production of an industrial-transformer) seems to contribute to the hybrid production and sourcing policy ($Q > Q^0 > 0$), while the opposite (e.g. production of a bicycle) results in complete self-production ($Q^0 = 0$). Besides, another factor leading to the decision of complete self-production of the bicycle case is that its ordering cost is relatively high (\$0.04 per unit), compared to its fast production. Reducing the ordering cost to \$0.004 per unit, the result will be changed. As shown in Table 5, only the SFVs of ten bicycle orders are suggested to produce at the generic stage, and the rest (five orders) are required to source from suppliers. It reveals the application of hybrid production and sourcing policy as well. Therefore, the sourcing decision is impacted by the relationship between production rate and ordering cost, and the results in Tables 2, 4 and 5 validate the applicability of integrating SFV's manufacturing and sourcing in DPDCM.
- *Generic inventory decision.* The result of the industrial-transformer case proves the validity of the alternative between SFV and VIs, in which β_3, β_7 and β_9 are equal to zero (Table 4). In addition, increasing the setup and processing times of each bicycle *variant* by ten times (decreasing the production rate), the result validates the situation as well. As listed in Table 6, because of the decreased production rate, the constraint of the processing cycle time becomes much fiercer than the initial condition. The processing cycle time have to be reduced, thus four out of five SFVs of the bicycle production have to be replaced with their alternative VIs.

To sum up, the near-optimal solutions listed in Tables 5 and 6, as well as those shown in Tables 2 and 4, demonstrate the universality and flexibility of DPDCM with respect to providing various suitable decisions for generic inventory management in diverse conditions.

6.2. Impact of demand rate

Demand rate is one of corefactors influencing the decision results of DPDCM. Taking the bicycle manufacturing as example, set the possible annual demand rate λ as 10, 20, 25, 30, 40, 50, 100, 150, 200, 250 and 300. Table 7 presents the near-optimal solutions, showing the impacts on order-release and sourcing decisions.

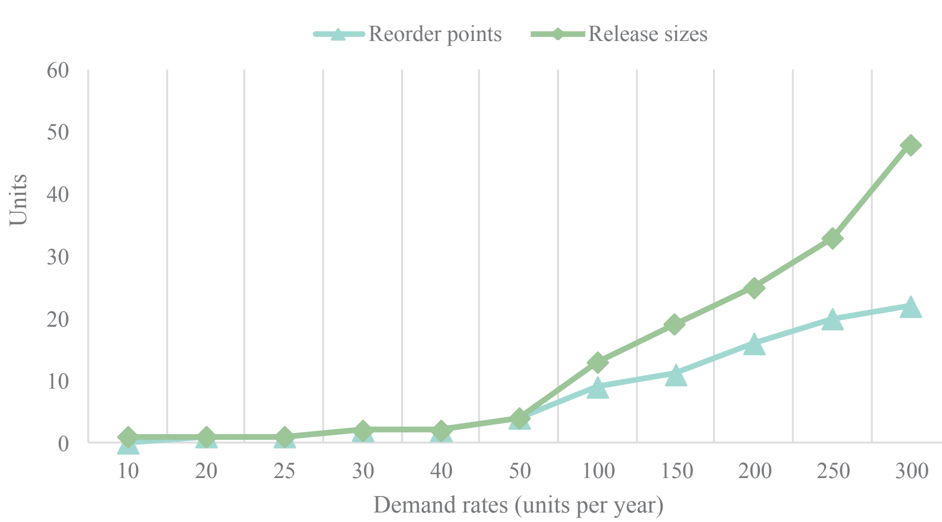

Fig. 8. Release sizes and reorder points with different demand rates.

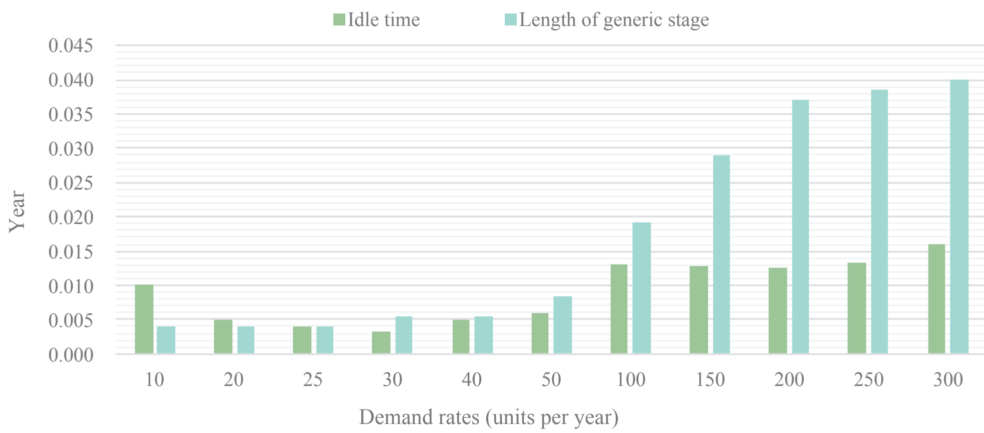

Fig. 9. Relationship between manufacturing idle time and length of generic stag.

Table 8
Near-optimal solutions with the vary of demand rates of the industrial-transformer case.

λ	μ	σ	QT^u/λ	t^s	Q	Q^0	R	β_1	β_2	β_3	β_4	β_5	β_6	β_7	β_8	β_9	TC
10	0	1	0.0060	0.0059	1	0	1	1	1	0	0	1	1	1	1	1	615.169
20	1	1	0.0060	0.0057	2	1	2	1	1	1	1	1	1	1	1	0	625.670
30	2	1	0.008	0.0070	4	3	2	1	1	1	1	0	1	1	1	1	479.770
60	3	2	0.007	0.0135	7	4	7	1	1	0	1	1	1	0	1	0	547.464
90	4	2	0.0067	0.0219	10	5	10	1	1	1	1	1	1	0	1	1	569.991
120	5	3	0.0070	0.0270	14	7	14	1	1	0	1	1	1	0	1	0	570.255
150	6	4	0.0072	0.0149	18	14	18	1	1	0	0	1	1	1	1	0	593.885
180	7	4	0.0070	0.0196	21	11	20	0	1	0	1	1	1	1	1	0	646.998
210	8	5	0.0071	0.0214	25	14	25	0	1	0	1	1	1	0	1	0	683.611
240	9	6	0.0070	0.0131	28	21	28	0	1	0	1	0	1	0	1	1	771.498
270	10	6	0.0071	0.0096	32	23	30	0	1	0	1	1	0	1	1	0	817.963
300	11	7	0.0072	0.0098	36	31	35	0	1	0	1	0	1	0	1	0	830.161

- *Order-release decision.* As shown in Table 7, DPDCM applies BP pattern when $\lambda \geqslant 30$, otherwise SP pattern. It proves that DPDCM is capable of switching between BP and SP patterns. Moreover, Q and R both rise with the growth of λ (refer to Fig. 8), which satisfies the common sense that more demands more products. Furthermore, Fig. 9 shows the relationship between the manufacturing idle time (QT^u/λ) and the length of generic stage (t^s). The generic stage is able to make full use of the manufacturing idle in most situations ($t^s \geqslant QT^u/\lambda$) except for the results at $\lambda = 10$ and $\lambda = 20$. Nevertheless, the outlier results are also reasonable, as it implies that the manufacturing idle cannot be eliminated without enough demand. Overall, the results validate the universality of DPDCM and the fact of utilizing the manufacturing idle to trigger the generic stage.
- *Sourcing decision.* Table 7 shows that when $\lambda \leqslant 200$ all SFV inventory are planned to produce at the generic stage ($Q^0 = 0$), whereas a hybrid strategy is employed when $\lambda \geqslant 250$ (as $0 < Q^0 < Q$). It demonstrates the feasibility of sourcing SFV inventory for DPDCM, especially in the case of facing high demand rates.

The same pattern of order-release and sourcing decisions can be seen from the industrial-transformer case. Please refer to the data shown in Table 8. Moreover, the frequency of stocking the VIs instead of SFVs tends to be increased with the growth of annual demand rate, which meets the requirements of competing for fast response. In addition, the components with less VIs (e.g. oiling, radiator and tank & accessories with $U_n = 1$, 3 and 3 respectively) have the priority to hold in inventory, as they contribute to lower inventory cost compared to those that have more VIs. It meets the requirement of inventory control.

All the facts shown by Tables 7 and 8 illustrate the validity of the formulations of order-release, generic inventory and sourcing decisions.

6.3. Impact of maximum processing cycle time

The relationship among $T^{\max}$, Q and TC are analyzed as well. Taking the bicycle manufacturing as example, $T^{\max}$ is set to vary from 0.04 to 0.1 with an interval of 0.01, as it cannot be shorter than τ. Fig. 10 shows the comparison between Q and TC. With the increase of $T^{\max}$, the order-release size grows, while the inventory cost decreases. It clarifies the inverse-proportional relationship between Q and TC, proving that BP is economic, and reflecting the trade-off between generic inventory cost and processing cycle time, as Q is proportional to T.

7. Conclusion

The paper formulates an optimization model – DPDCM – to make DPD decision for CMfg, which integrates order-release, generic inventory, and sourcing decisions. The case studies prove that DPDCM is effectively flexible and universal. It not only has the common features of DPD, but also can cater for various manufacturing requirements of CMfg.

(i) The order-release decision, integrating both SP and BP patterns, can optimize order-release size regardless of the adopted production patterns. As the cloud system is usually centrally controlled and has large connection networks, using an integrated policy is more efficient than employing two separate ones, reducing the complexity of a cloud system.
(ii) The manufacturing idle at customization stage triggers the production of generic stage. It enables the manufacturing firms to pay attention to the value-added customization productions, which can improve the manufacturers' competitiveness.
(iii) With the application of adjustable generic inventory and sourcing decisions, DPDCM can achieve more flexible ways of inventory planning than traditional DPD approaches. It can offer the manufacturers with more suitable decisions according to their unique

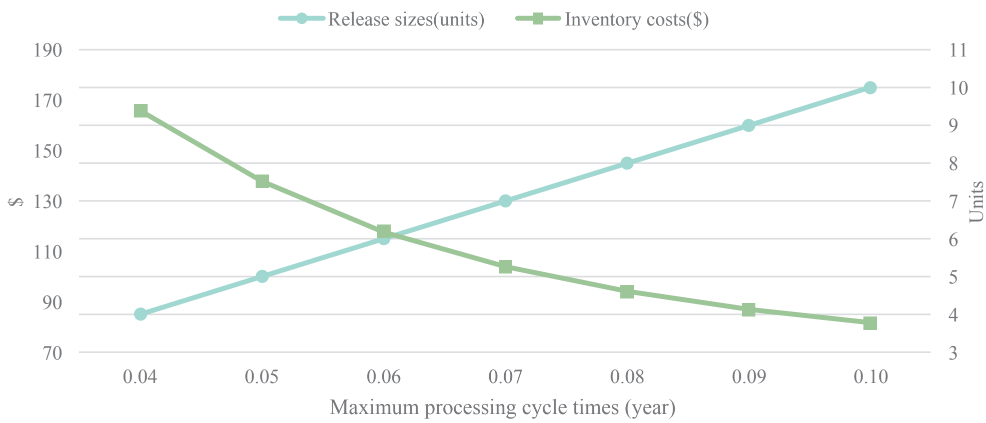

Fig. 10. Relationship between order-release size and inventory cost.

production situations.

(iv) DPDCM can be provided as a cloud service (e.g. software as a service), regarded as a critical way to attract potential manufacturers to join in CMfg, especially the small and medium-sized enterprises.

Nevertheless, there are several limitations. First of all, to simplify the DPDCM model, the expected manufacturing idle period is set as the average of the annual sampling, without considering all complex factors in a real-life environment. In addition, to select the generic inventory (SFV or its corresponding VIs), we only considered the economic factors, such as inventory costs, excluding some technical factors, e.g., engineering design and energy consumption. Current research effort is directed at improving the model.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Program No. 51275456) and the Zhejiang Natural Science Foundation of China (Program No. LY18E050014). The first and second authors acknowledge the financial support received from the China Scholarship Council (CSC) Scholarship.

References

Adamson, G., Wang, L., & Holm, M. (2013). The state of the art of cloud manufacturing and future trends. *ASME 2013 international manufacturing science and engineering conference collocated with the 41st North American manufacturing research conference*ASME pp. V002T02A004.

Algeddawy, T., & Elmaraghy, H. (2010). Assembly systems layout design model for delayed products differentiation. *International Journal of Production Research, 48*(18), 5281–5305.

Caux, C., David, F., & Pierreval, H. (2006). Implementation of delayed differentiation in batch process industries – A standardization problem. *International Journal of Production Research, 44*(16), 3243–3255.

Chen, F., Dou, R., Li, M., & Wu, H. (2015). A flexible QoS-aware web service composition method by multi-objective optimization in cloud manufacturing. *Computers & Industrial Engineering, 99*, 423–431.

Chen, S. H., Kang, H. Y., & Lee, H. T. (2006). Dynamic programming approach for analyzing delayed product differentiation. *International Journal of Advanced Manufacturing Technology, 28*(5), 445–449.

Dekok, A. G., & Graves, S. C. (Vol. Eds.), (2003). *Supply chain management: Design, coordination and operation. vol. 11*. Amsterdam: Elsevier.

Ding, B., Yu, X. Y., & Sun, L. J. (2012). A cloud-based collaborative manufacturing resource sharing services. *Information Technology Journal, 11*(9), 1258–1264.

Feitzinger, E., & Lee, H. L. (1997). Mass customization at Hewlett Packard: The power of postponement. *Harvard Business Review, 75*(1), 116–121.

Forza, C., Salvador, F., & Trentin, A. (2008). Form postponement effects on operational performance: A typological theory. *International Journal of Operations and Production Management, 28*(11), 1067–1094.

Gupta, D., & Benjaafar, S. (2004). Make-to-order, make-to-stock, or delay product differentiation? A common framework for modeling and analysis. *IIE Transactions, 36*(6), 529–546.

Hanafy, M., & Elmaraghy, H. (2015). Developing assembly line layout for delayed product differentiation using phylogenetic networks. *International Journal of Production Research, 53*(9), 2633–2651.

He, D., Kusiak, A., & Tseng, T. (1998). Delayed product differentiation: A design and manufacturing perspective. *Computer-Aided Design, 30*(2), 105–113.

Hsu, H., & Wang, W. (2004). Dynamic programming for delayed product differentiation. *European Journal of Operational Research, 156*(1), 183–193.

Jassbi, J., Orio, G. D., Barata, D., & Barata, J. (2014). The impact of cloud manufacturing on supply chain agility. *IEEE international conference on industrial informatics* (pp. 495–500). IEEE.

Jewkes, E. M., & Alfa, A. S. (2009). A queueing model of delayed product differentiation. *European Journal of Operational Research, 199*(3), 734–743.

Kirkley, J., Paul, C. J. M., & Squires, D. (2002). Capacity and capacity utilization in common-pool resource industries. *Environmental and Resource Economics, 22*(1), 71–97.

Laili, Y., Tao, F., Zhang, L., Cheng, Y., Luo, Y., & Sarker, B. R. (2013). A ranking chaos algorithm for dual scheduling of cloud service and computing resource in private cloud. *Computers in Industry, 64*(4), 448–463.

Lee, H. L., & Billington, C. (1994). *Designing products and processes for postponementManagement of design*Netherlands: Springer.

Lee, H. L., & Tang, C. S. (1997). Modelling the costs and benefits of delayed product differentiation. *Management Science, 43*(1), 40–53.

Li, C. Q., Hu, C. Y., Wang, Y. W., & Zhu, P. F. (2011). Research of cloud manufacturing and resource encapsulation technology. *Applied Mechanics and Materials, 58*, 562–566.

Li, B. H., Zhang, L., Wang, S. L., Tao, F., Cao, J. W., Jiao, X. D., et al. (2010). Cloud manufacturing: A new service-oriented networked manufacturing model. *Computer Integrated Manufacturing Systems, 16*(1), 1–7.

Liu, I., & Jiang, H. (2012). Research on key technologies for design services collaboration in cloud manufacturing. *16th international conference on computer supported cooperative work in design (CSCWD)* (pp. 824–829). IEEE.

Lu, Y., Xu, X., & Xu, J. (2014). Development of a hybrid manufacturing cloud. *Journal of Manufacturing Systems, 33*(4), 551–566.

Magretta, J. (1998). The power of virtual integration: An interview with Dell Computer's Michael Dell. *Harvard Business Review, 76*(2), 72–85.

Nahmias, S. (2001). *Production and operations analysis.* Tsinghua University Press.

Olhager, J. (2010). The role of the customer order decoupling point in production and supply chain management. *Computers in Industry, 61*(9), 863–868.

Qi, L. (2013). A continuous-review inventory model with random disruptions at the primary supplier. *European Journal of Operational Research, 225*(1), 59–74.

Simpson, T. W., Siddique, Z., & Jiao, R. J. (2006). *Product platform and product family design: Methods and applications.* Springer Science and Business Media.

Skipworth, H., & Harrison, A. (2004). Implications of form postponement to manufacturing: A case study. *International Journal of Production Research, 42*(10), 223–228.

Song, Z., & Kusiak, A. (2010). Mining Pareto-optimal modules for delayed product differentiation. *European Journal of Operational Research, 201*(2), 123–128.

Su, J. C. P., Chang, Y. L., & Ferguson, M. (2005). Evaluation of postponement structures to accommodate mass customization. *Journal of Operations Management, 23*(3–4), 305–318.

Su, J. C. P., Chang, Y. L., Ferguson, M., & Ho, J. C. (2010). The impact of delayed differentiation in make-to-order environments. *International Journal of Production Research, 48*(19), 5809–5829.

Swaminathan, J. M., & Tayur, S. R. (1998). Managing broader product lines through delayed differentiation using vanilla boxes. *Management Science, 44*(12), 161–172.

Tang, D. (2011). Managing finished-goods inventory under capacitated delayed differentiation. *Omega, 39*(5), 481–492.

Trentin, A., Salvador, F., Forza, C., & Rungtusanatham, M. J. (2011). Operationalising form postponement from a decision-making perspective. *International Journal of Production Research, 49*(7), 1977–1999.

Van Hoek, R. I. (2001). The rediscovery of postponement a literature review and directions for research. *Journal of Operations Management, 19*(2), 161–184.

Wang, X., & Xu, X. (2013). An interoperable solution for cloud manufacturing. *Robotics and Computer-Integrated Manufacturing, 29*(4), 232–247.

Wu, D., Greer, M. J., Rosen, D. W., & Schaefer, D. (2013). Cloud manufacturing: Strategic vision and state-of-the-art. *Journal of Manufacturing Systems, 32*(4), 564–579.

Xiang, F., & Hu, Y. F. (2012). Cloud manufacturing resource access system based on internet of things. *Applied Mechanics and Materials, 121*(126), 2421–2425.

Xu, X. (2012). From cloud computing to cloud manufacturing. *Robotics and Computer-Integrated Manufacturing, 28*(1), 75–86.

Yu, S., & Xu, X. (2015). Development of a product configuration system for cloud manufacturing. *Advances in production management systems: Innovative production management towards sustainable growth.* Springer International Publishing.

Yu, C., Ji, Y., Qi, G., Gu, X., & Tao, L. (2015). Group-based production scheduling for make-to-order production. *Journal of Intelligent Manufacturing, 26*(3), 585–600.

Yu, C., Zhang, W., Xu, X., Ji, Y., & Yu, S. (2015). Data mining based multi-level aggregate service planning for cloud manufacturing. *Journal of Intelligent Manufacturing*, 1–11.

Zhang, Y., Zhang, G., Liu, Y., & Hu, D. (2015). Research on services encapsulation and virtualization access model of machine for cloud manufacturing. *Journal of Intelligent Manufacturing*, 1–15.

Zhou, W., Zhang, R., & Zhou, Y. (2013). A queuing model on supply chain with the form postponement strategy. *Computers & Industrial Engineering, 66*(4), 643–652.