Journal of Manufacturing Systems 48 (2018) 122–133

Contents lists available at ScienceDirect

Journal of Manufacturing Systems

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

An augmented Lagrangian coordination method for optimal allocation of cloud manufacturing services

Geng Zhang [a,b], Yingfeng Zhang [a,*], Xun Xu [b], Ray Y. Zhong [b]

[a] Key Laboratory of Contemporary Design and Integrated Manufacturing Technology, Ministry of Education, Northwestern Polytechnical University, Xi'an, 710072, China

[b] Department of Mechanical Engineering, The University of Auckland, Auckland, 1142, New Zealand

ARTICLE INFO

Article history:
Received 1 September 2017
Received in revised form 6 November 2017
Accepted 25 November 2017
Available online 12 December 2017

Keywords:
Cloud manufacturing
Manufacturing service allocation
Augmented Lagrangian coordination

ABSTRACT

With the rapid development of information and manufacturing technologies, cloud manufacturing (CMfg) was proposed and attracted wide attention. In CMfg, manufacturing service allocation (MSA) plays an important role in facilitating high-quality service management. MSA aims to optimize the allocation of services for manufacturing tasks. All-in-one (AIO) methods are widely used to obtain an optimal MSA result. However, the current AIO methods usually use one decision model so that it is difficult to maintain the autonomous decision rights of service providers. As a distributed optimization mechanism, augmented Lagrangian coordination (ALC) can offer an open-structure collaboration and allow participants to keep autonomous decision rights. In this paper, the MSA problem is partitioned into an ALC model based on the decision scope of service providers and solved in a loose coupling and distributed manner. A case study demonstrates the specific steps of ALC for solving the MSA problem. The results show the effectiveness and efficiency of ALC method in solving the MSA problem, as well as its promising feature in maintaining decision autonomy of a service provider.

© 2017 The Society of Manufacturing Engineers. Published by Elsevier Ltd. All rights reserved.

1. Introduction

Due to the growth of globalization competition and product personalization, manufacturing enterprises are changing their businesses modes [1]. To fulfill the diversiform producing demands, many advanced manufacturing modes have been proposed [2]. As a new service-oriented manufacturing paradigm, cloud manufacturing (CMfg) bears the traces of some of the existing manufacturing modes (e.g. agile manufacturing [3], virtual manufacturing [4] and manufacturing grid [5]). It integrates advanced information technologies, such as cloud computing [6,7], Internet of Things (IoT) [8–10], and service-oriented techniques [11,12].

In CMfg, manufacturing resources with different capabilities are encapsulated into manufacturing services [13,14]. Manufacturing service allocation (MSA) can be considered as the process of service composition and selection for specific manufacturing demands or tasks [15]. MSA plays an important role in implementing the full-scale sharing and on-demand use of various cloud manufacturing services. MSA has some characteristics such as large scale, high heterogeneity, dynamic interconnection and group collaboration [16]. Effective and efficient MSA can avoid both over-provisioning and idle resources for achieving sustainable productions.

Since CMfg was proposed, abundant research work has been carried out in this area, including manufacturing resource perception and connection [17], cloud service modeling and description [18], service searching and matching [19,20], service management platform [21], etc. Various all-in-one (AIO) optimization methods have also been proposed for MSA problems. The AIO optimization method can be defined as an adoption of a centralized strategy to implement the whole optimization process. All the related decision variables are considered in one optimization model. In terms of the MSA problems, they can be categorized according to two aspects. One aspect is the MSA algorithms. Examples include cuckoo search-based artificial bee colony (ABC) and differential evolution-based ABC for service composition and optimal selection [22,23], Gale–Shapley algorithm for service sharing [24], mechanism design approach for service allocation [25], resource-service chain composition algorithm [26], game theory-based service scheduling [27]. The other aspect is about different manufacturing objectives and constraints for MSA. The quality of service indexes (e.g. cost, time, and reliability) are often used to distinguish the optimal MSA results [28,29]. In order to promote sustainable production, energy

* Corresponding author.
E-mail address: zhangyf@nwpu.edu.cn (Y. Zhang).

https://doi.org/10.1016/j.jmsy.2017.11.008
0278-6125/© 2017 The Society of Manufacturing Engineers. Published by Elsevier Ltd. All rights reserved.

consumption is also considered as an objective [30,31]. Constraints are proposed for meeting specific allocation requirements, such as the correlation between services [32], geo-perspective transportation [33], and workload-based multi-task scheduling in CMfg [34].

The applications of AIO methods have made significant progress in improving MSA. However, some challenges still hinder the implementation of effective and efficient manufacturing services management.

1) AIO methods mostly adopt centralized decision strategy without considering the autonomous decision rights of service providers. Actually, service providers with different capabilities may have their own provision modes to maintain autonomous decision rights to keep their flexibilities during the MSA process.
2) In order to get the optimal MSA results, AIO methods may consider a large number of service parameters and manufacturing task requirements in one decision model. The complexity of the MSA problem will be extremely high when a large number of manufacturing tasks or a manufacturing task with multiple sub-stages is submitted to CMfg platform. This could result in significant cost from computation perspective.
3) AIO methods lack the capability of service re-allocation. When production exception occurs, local re-allocation of cloud manufacturing service is needed.

Given the above challenges, a distributed method may solve the MSA problem.

Augmented Lagrangian coordination (ALC) method is a newly emerged distributed approach for multidisciplinary design optimization [35,36]. It is based on a combination of augmented Lagrangian relaxation and block-coordinate descent from mathematical programming. The basic working logic of ALC method is first to partition the whole system or problem into several independent decision elements according to certain decomposition rules and then coordinate these elements to get a global optimal solution. ALC has the features of providing disciplinary design autonomy, offering high degree flexibility in setting up coordination structure, maintaining mathematical rigor, and being efficient in getting global optimal designs. Due to the promising features of ALC in solve engineering problems (e.g. cluster supply chain configuration [37], large complex system design [38]), it is extended for the MSA problem in this paper.

The primary aim of this research is to investigate how ALC method can be extended to deal with the MSA problem while retaining autonomous decision making and distributed computing advantages. In order to maintain simplicity without losing generality, this research designates the MSA of a manufacturing task with multiple sub-stages as a research problem referring some research questions. The main contributions of this paper can be described as follows. First, the mechanism of MSA in cloud manufacturing is investigated, including the working logic of MSA and different types of service allocation. Second, a distributed optimization strategy to the MSA process is introduced. An ALC model is constructed and formulated for the MSA problem. Third, the verification of effectiveness and sensitivity of ALC in solving MSA problem is carried out.

The rest of this paper is organized as follows. Section 2 introduces the manufacturing service allocation in cloud manufacturing. The general principles of ALC method are explained in Section 3. Section 4 describes the detailed ALC steps for an MSA problem. Results analysis is given in Section 5. Section 6 draws the conclusions and future work.

2. Manufacturing service allocation in cloud manufacturing

In CMfg, MSA is responsible for allocating right services to match the manufacturing requirements from various tasks. The working logic of MSA and different MSA types in CMfg are described as follows.

2.1. Working logic of MSA

There are three kinds of users in CMfg. They are cloud operator, service provider, and customer. Cloud operator is responsible for setting up operation mechanisms and rules (e.g., technical process and business strategy) to run the CMfg platform and ensure the communication between service provider and customer. Service providers offer manufacturing resources for specific manufacturing demands. Customer submits its manufacturing task to CMfg platform and gets the on-demand services.

Fig. 1 shows the working logic of MSA which can be described as follows. Manufacturing resources (e.g. machine) from service providers are encapsulated into services and registered at the CMfg platform. A service pool will be formed by these registered services. Meanwhile, the customer submits manufacturing task that can be composed of multiple sub-tasks to the CMfg platform. According to task information (e.g. processes sequence, delivery time, and cost) and service information, the optimization method will be invoked to implement the MSA. In this paper, the optimization method means ALC method. During the optimization process, if a service provider has the decision autonomy, an optimization model of the MSA problem will be constructed according to service provider's decision right. Then the MSA can be performed based on the constructed optimization model. After the MSA, the customer will get a series of on-demand services, and the service provider will get one sub-task to complete. Under the support of information technologies (e.g. IoT), real-time processing information can be shared among CMfg platform, customer and service provider. If an exception happens to one service, re-allocation of cloud manufacturing services is needed to find other available services to complete matching sub-task.

2.2. Two different types of MSA problems

As shown in Fig. 2, two different types of MSA problems are discussed given their autonomous decision rights. For better illustration, some notations are as follows. S represents the CMfg platform, $S\text{-}T_i$ represents i^{th} sub-task, O indicates service options, O_i^j represents j^{th} candidate services for i^{th} sub-task. System or services in grey dashed circle are the owners with independent decision rights. Dashed block indicates the decision scope of an owner with independent decision rights.

Fig. 2(a) illustrates the MSA problems without decision autonomy. In these problems, the CMfg platform has the supreme decision right at all the manufacturing stages. S has the decision right for allocating services to all the three sub-tasks (i.e. $S\text{-}T_1$, $S\text{-}T_2$, and $S\text{-}T_3$). Each candidate service provider (e.g. service provider of O_2^1) has no decision autonomy. Fig. 2(b) illustrates the MSA problems with decision autonomy. In these problems, service providers of a sub-task can maintain their autonomous decision rights and form different service chains with upstream sub-tasks. This is because that these service providers often have several alliance members to complete some specific tasks. As shown in Fig. 2(b), two service providers (i.e. service provider of O_2^1 and O_2^2) have their own decision autonomy rights to compete $S\text{-}T_2$ with two different service chains of upstream sub-task $S\text{-}T_1$. For example, O_2^1 has its alliance members O_1^1 and O_1^2, and O_2^1 is responsible for selecting one service to complete sub-task $S\text{-}T_1$.

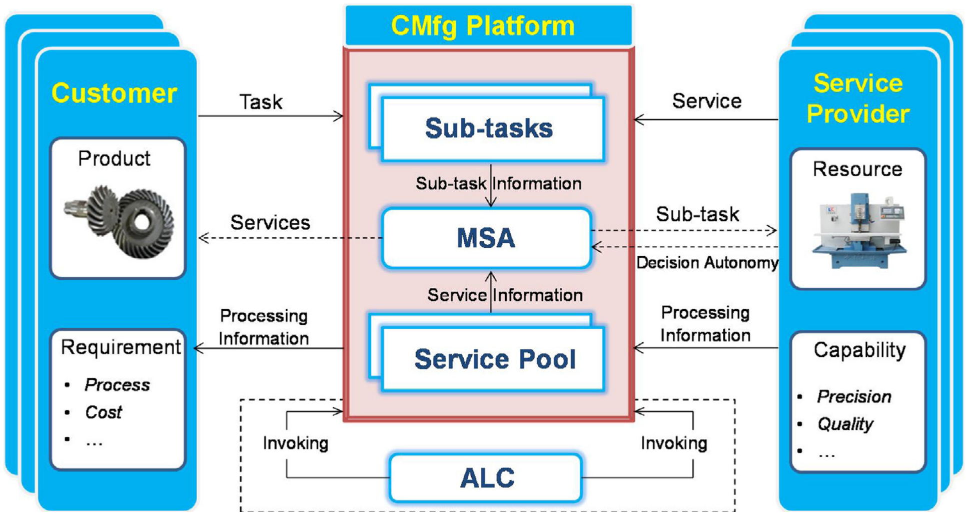

Fig. 1. Working logic of manufacturing service allocation.

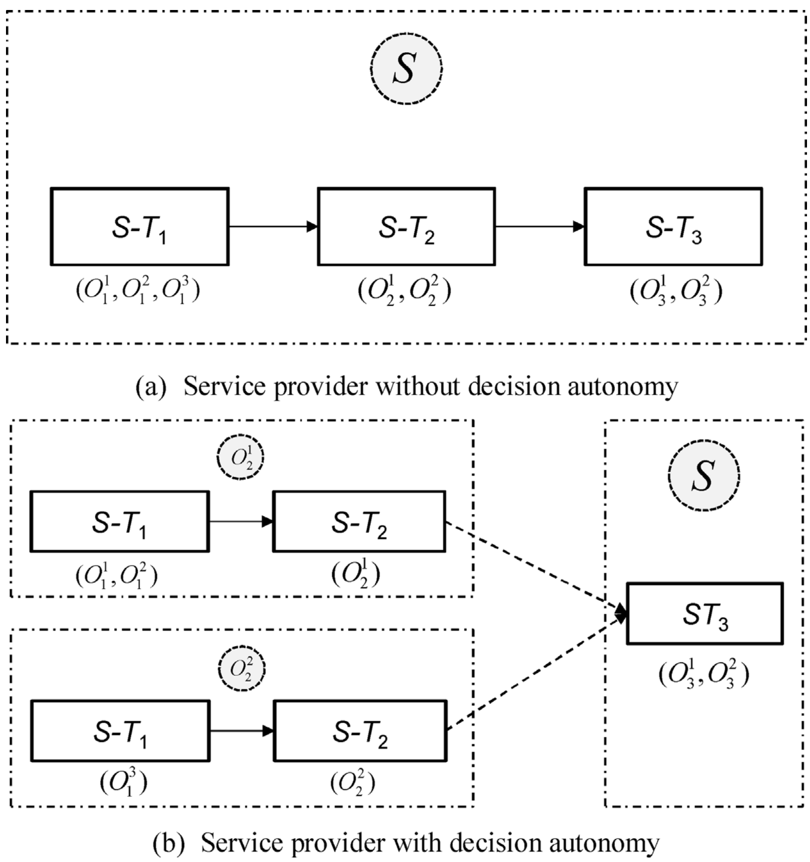

Fig. 2. Two different types of MSA problems.

3. ALC method

This section mainly focuses on illustrating the principle of implementing ALC method. It is composed of following steps: (1) Partition the complex system problem. This step aims to partition the complex system into smaller elements which can be optimized autonomously. Specific rules can be used, such as decision autonomy. (2) Introduce auxiliary variables and consistency constraints to the related elements. Auxiliary variables and consistency constraints are introduced at each element to separate the local constraints. (3) Relax the consistency constraints. The purpose is to figure out fully separable constraint sets. (4) Formulate the partitioned problem. (5) Coordinate the partitioned problem and get the optimal results.

For better understanding, a geometric programming problem [39] is taken as an example to explain the detailed information of each step. The original problem can be defined as follows. The objective function aims to minimize f which is the sum of two local objectives $f_1 = z_1^2$ and $f_2 = z_2^2$. $[z_1, z_2, z_3, z_4, z_5, z_6, z_7]^T$ is the vector of variables.$g_1 = (z_3^{-2} + z_4^2)z_5^{-2} - 1 \leq 0$ and $g_2 = (z_5^2 + z_6^{-2})z_7^{-2} - 1 \leq 0$

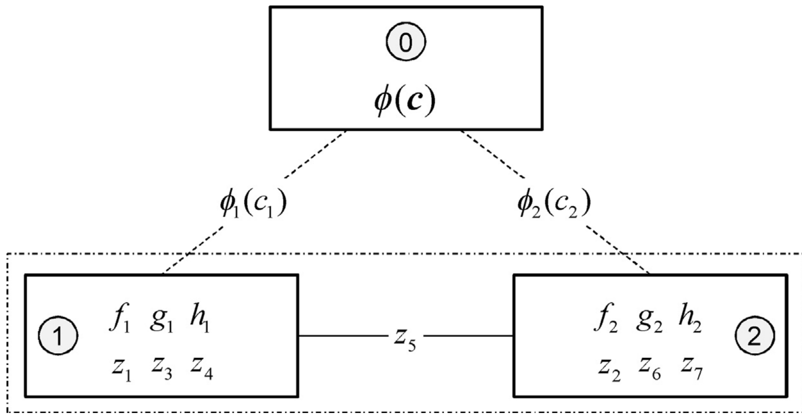

Fig. 3. ALC for geometric programming problem.

are inequality constraints. $h_1 = (z_3^2 + z_4^{-2} + z_5^2)z_1^{-2} - 1 = 0$ and $h_2 = (z_5^2 + z_6^2 + z_7^2)z_2^{-2} - 1 = 0$ are equality constraints.

3.1. Partitioning complex system problem

The partitioning of the original problem is shown in the dashed block of Fig. 3. It consists of two elements: Element 1 and Element 2. Local variables z_1, z_3 and z_4 are allocated to Element 1, along with the local objective f_1 and constraints g_1, h_1. Similarly, Local variables z_2, z_6 and z_7 are allocated to Element 2, along with the objective f_2 and constraints g_2, h_2. Variable z_5 is one linking variable. In ALC method, linking variables are shared by two or more related elements.

3.2. Introduction of auxiliary variables and consistency constraints

In the partitioned geometric programming problem, two auxiliary variables $z_5^{[1]}$ and $z_5^{[2]}$ are respectively introduced to Element 1 and Element 2. They are copies of the linking variable z_5. $z_5^{[1]}$ and $z_5^{[2]}$ are forced equal by non-separable consistency constraints $\mathbf{c} = [c_1, c_2]^T = [0, 0]^T$ (i.e. $c_1 = z_5 - z_5^{[1]}$, $c_2 = z_5 - z_5^{[2]}$). Then, the original problem can be presented as follows.

$$\text{Objective function } \min f = f_1 + f_2 = z_1^2 + z_2^2 \quad \text{Variables } z_1, z_2, z_3, z_4, z_5, z_5^{[1]}, z_5^{[2]} z_6, z_7 \tag{1}$$

$$\text{Subject to } g_1 = (z_3^{-2} + z_4^2)(z_5^{[1]})^{-2} - 1 \le 0 \tag{2}$$

$$g_2 = ((z_5^{[2]})^2 + z_6^{-2})z_7^{-2} - 1 \le 0 \tag{3}$$

$$h_1 = (z_3^2 + z_4^{-2} + (z_5^{[1]})^2)z_1^{-2} - 1 = 0 \tag{4}$$

$$h_2 = ((z_5^{[2]})^2 + z_6^2 + z_7^2)z_2^{-2} - 1 = 0 \tag{5}$$

$$\mathbf{c} = [c_1, c_2]^T = [z_5 - z_5^{[1]}, z_5 - z_5^{[2]}]^T = [0, 0]^T \tag{6}$$

3.3. Relaxation of consistency constraints

The augmented Lagrangian penalty function φ is used to relax consistency constraint c.

$$\phi(\mathbf{c}) = \mathbf{v}^T\mathbf{c} + ||\mathbf{w} \circ \mathbf{c}||_2^2 = \phi_1(c_1) + \phi_2(c_2) = v_1(z_5 - z_5^{[1]}) + ||w_1(z_5 - z_5^{[1]})||_2^2 + v_2(z_5 - z_5^{[2]}) + ||w_2(z_5 - z_5^{[2]})||_2^2 \tag{7}$$

Where $\mathbf{v} = [v_1, v_2]^T$ represents the vector of Lagrange multiplier estimates, $\mathbf{w} = [w_1, w_2]^T$ represents the vector of penalty weights. $\mathbf{v}$ and $\mathbf{w}$ are called penalty parameters. "$\circ$" represent the Hadamand product. The resulting relaxed original problem is given as follows.

Objective function

$$\min z_1^2 + z_2^2 + v_1(z_5 - z_5^{[1]}) + ||w_1(z_5 - z_5^{[1]})||_2^2 + v_2(z_5 - z_5^{[2]}) + ||w_2(z_5 - z_5^{[2]})||_2^2 \quad \text{Variables } z_1, z_2, z_3, z_4, z_5, z_5^{[1]}, z_5^{[2]} z_6, z_7 \tag{8}$$

$$\text{Subject to } g_1 = (z_3^{-2} + z_4^2)(z_5^{[1]})^{-2} - 1 \le 0 \tag{9}$$

$$g_2 = ((z_5^{[2]})^2 + z_6^{-2})z_7^{-2} - 1 \le 0 \tag{10}$$

$$h_1 = (z_3^2 + z_4^{-2} + (z_5^{[1]})^2)z_1^{-2} - 1 = 0 \tag{11}$$

$$h_2 = ((z_5^{[2]})^2 + z_6^2 + z_7^2)z_2^{-2} - 1 = 0 \tag{12}$$

3.4. Formulation of the partitioned problem

After the relaxation, the constraints are fully separable with respect to variables of each element. Variables set $\{z_1, z_3, z_4, z_5^{[1]}\}$ is associated with Element 1 and variables set $\{z_2, z_6, z_7, z_5^{[2]}\}$ is associated with Element 2. As shown in Fig. 3, a master element, Element 0 is introduced. In ALC method, the introduction of the master element allows a parallel solution of the partitioned elements. The formulation of the master element only includes the penalty terms $\phi(\mathbf{c})$. The objective function of each partitioned element (i.e. Element 1 or Element 2) consists of two parts. The first part is the allocated local objective. The second part accounts for the relaxation of consistency constraint. Each element can be formulated as follows.

- Formulation of Element 0

$$\min \phi(\mathbf{c}) = v_1(z_5 - z_5^{[1]}) + ||w_1(z_5 - z_5^{[1]})||_2^2 + v_2(z_5 - z_5^{[2]}) + ||w_2(z_5 - z_5^{[2]})||_2^2 \tag{13}$$

- Formulation of Element 1

$$\text{Objective function } \min z_1^2 + v_1(z_5 - z_5^{[1]}) + ||w_1(z_5 - z_5^{[1]})||_2^2 \tag{14}$$

$$\text{Subject to } g_1 = (z_3^{-2} + z_4^2)(z_5^{[1]})^{-2} - 1 \le 0 \tag{15}$$

$$h_1 = (z_3^2 + z_4^{-2} + (z_5^{[1]})^2)z_1^{-2} - 1 = 0 \tag{16}$$

- Formulation of Element 2

$$\text{Objective function } \min z_2^2 + v_2(z_5 - z_5^{[2]}) + ||w_2(z_5 - z_5^{[2]})||_2^2 \tag{17}$$

$$\text{Subject to } g_2 = ((z_5^{[2]})^2 + z_6^{-2})z_7^{-2} - 1 \le 0 \quad (18)$$

$$h_2 = ((z_5^{[2]})^2 + z_6^2 + z_7^2)z_2^{-2} - 1 = 0 \quad (19)$$

3.5. Coordination for the partitioned problem

A coordination strategy for solving the formulated elements consists of inner loops and outer loops. In the inner loops, each element is solved with fixed penalty parameters, and in the outer loops, penalty parameters are updated. The specific steps to implement the solution strategy can be described as follows: (1) Set initial penalty parameters (i.e. $\boldsymbol{v}$ and $\boldsymbol{w}$). (2) Adopt block coordinate descent (BCD) algorithm to solve each element for fixed penalty parameters. (3) Check the convergence and terminate when convergence conditions are satisfied, otherwise go to next step. (4) Update penalty parameters with the method of multipliers and return to step (2).

- Outer loops

In the outer loops, the method of multipliers is adopted to update the estimates of penalty parameters according to following equations.

$$\boldsymbol{v}^{k+1} = \boldsymbol{v}^k + 2\boldsymbol{w}^k \circ \ \boldsymbol{w}^k \circ \mathbf{c}^k \quad (20)$$

$$w_m^{k+1} = \begin{cases} w_m^k |c_m^k| \le \gamma |c_m^{k-1}| \\ \beta w_m^k |c_m^k| > \gamma |c_m^{k-1}| \end{cases} \quad (21)$$

$$||\mathbf{c}^k - \mathbf{c}^{k-1}||_\infty < \varepsilon \quad (22)$$

$$||\mathbf{c}^k||_\infty < \varepsilon \quad (23)$$

Here, k represents the iterations of outer loops, c_m represents m^{th} consistency constraint, w_m is the penalty weight. β and γ are used to speed up convergence. Normally, $\beta > 1$ and $0 < \gamma < 1$. Eqs. (22) and (23) are two convergence conditions. The outer loop is terminated when the conditions are satisfied. Eq. (22) indicates that the change in the maximal consistency constraint value for two consecutive outer loop iterations should be smaller than the user-defined termination tolerance $\varepsilon > 0$. Eq. (23) indicates that the maximal consistency constraint violation should also be smaller than tolerance ε.

- Inner loops

In the inner loops, BCD is used to solve each partitioned element. BCD is known as an alternating optimization method with iterations between solving master element (i.e. Element 0) and the partitioned elements (i.e. Element 1, Element 2) in parallel. The inner loop is terminated when Eq. (24) is satisfied. It indicates that the relative change in the objective function value of the relaxed original problem for two consecutive iterations should be smaller than the user-defined termination tolerance $\varepsilon_{inner} > 0$. Normally, $\varepsilon_{inner} = \varepsilon/100$. ξ represents the inner loop iterations number, F denotes the objective of relaxed problem. For the geometric programming problem, F is represented as Eq. (8).

$$\frac{|F^\xi - F^{\xi-1}|}{1 + |F^\xi|} < \varepsilon_{inner} \quad (24)$$

Alternating direction method of multipliers (ADMOM) is an extreme case of BCD. It is to terminate the inner loop after just a single iteration. ADMOM is expected to convergence faster than basic BCD because less effort is put in costly inner loop iterations. After implementing all the steps, an optimization result of the original problem can be obtained.

Table 1
Notations.

$S - T_i$	Sub-task i
$\boldsymbol{O}_i$	Set of candidate services for sub-task i
$O_i^j \in \boldsymbol{O}_i$	j^{th} candidate service for sub-task i
TC_i	Total cost for completing sub-task i
PC_i	Local processing cost for completing sub-task i
LC_i	Local logistics cost for completing sub-task i
TT_i	Total time for completing sub-task i
PT_i	Local processing time for completing sub-task i
LT_i	Local logistics time for completing sub-task i
ST_i	Start time for processing sub-task i
s_i^j	Select coefficient for service O_i^j
E_i^j	Earliest start time of service O_i^j processing sub-task i
w_c	Weight coefficient for total cost
w_t	Weight coefficient for total time

4. ALC for MSA

MSA problem of a manufacturing task with multiple sub-stages is focused in this paper. This section aims to apply ALC to this problem with a motivating case model. The general steps of implementing ALC method will be followed. Notations are listed in Table 1.

4.1. Motivating model

An automotive engine is chosen as the product to be manufactured. The motivating model referred in this paper is a task of producing some key components for the automotive engine assembly. Note that different types of automotive engines have their own specific producing and assembling processes. Fig. 4 shows the processes and logistics flows of the task which includes six different types of components. According to the processes flow, the components are processed as the sequence of valve, crankcase, connecting rod, oil pan, gear housing, and EGR passage. Specifically, the processing parameter information of the latter component is partly decided by its former component. The detailed processing parameter information of latter component can't be acquired until the finished former component being transported to the assembly station. For example, the detailed parameter information of a crankcase can't be acquired until valves are finished and transported to the assembly station. MSA will be implemented when a customer submits the task to the CMfg platform.

According to the processes flow, the submitted manufacturing task can be decomposed into six sub-tasks, each of which is responsible for producing the corresponding component. There are some registered manufacturing services in the CMfg platform for completing the sub-tasks. Assume that each candidate service can meet the quality requirements from customer. The objective of this MSA problem is to minimize the sum of weighted total cost and time. The mathematical model can be stated as follows.

$$\text{Objective function min} \quad w_c TC_6 + w_t TT_6 \quad (25)$$

$$\begin{aligned} &\text{Variables } TC_i,\ PC_i,\ LC_i,\ TT_i,\ PT_i,\ LT_i,\ ST_i,\ s_i^j \\ &\text{Subject to} \quad TC_i = TC_{i-1} + PC_i + LC_i \end{aligned} \quad (26)$$

$$TT_i = ST_i + PT_i + LT_i \quad (27)$$

$$PC_i = \sum_{O_i^j \in \boldsymbol{O}_i} s_i^j \cdot pc_i^j \quad (28)$$

$$LC_i = \sum_{O_i^j \in \boldsymbol{O}_i} s_i^j \cdot lc_i^j \quad (29)$$

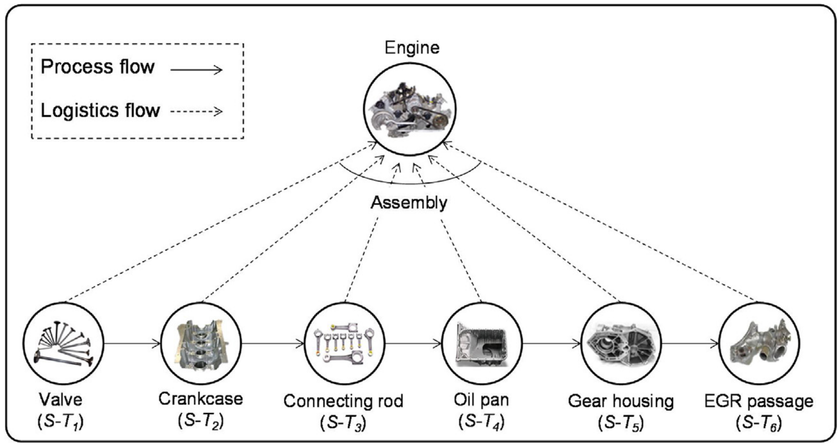

Fig. 4. Motivating model.

$$PT_i = \sum_{O_i^j \in \mathbf{O}_i} s_i^j \cdot pt_i^j \tag{30}$$

$$LT_i = \sum_{O_i^j \in \mathbf{O}_i} s_i^j \cdot lt_i^j \tag{31}$$

$$s_i^j = \begin{cases} 1 \text{ if } O_i^j \text{ is selected} \\ 0 \text{ otherwise} \end{cases} \quad \text{and} \sum_{O_i^j \in \mathbf{O}_i} s_i^j = 1 \tag{32}$$

$$TT_{i-1} \le ST_i \tag{33}$$

$$E_i^j \le ST_i \quad \text{if } O_i^j \text{ is selected} \tag{34}$$

$$i = 1, 2., \ldots, 6 \tag{35}$$

Two parts are included in the objective function, i.e. weighted total manufacturing cost and time. Total manufacturing cost for completing each task contains the total manufacturing cost of its upstream sub-task, local processing cost, and local logistics cost. Total manufacturing time for completing sub-task can be represented by the start time of this task, local processing time and logistics time. Eqs. (26) and (27) are used to get the total manufacturing cost and time for completing each sub-task. Eqs. (28)–(31) are employed to calculate the local processing cost, logistics cost, processing time and logistics time. Eq. (32) ensures that only one manufacturing service is selected to complete subtask *i*. Constraints (33) and (34) indicate that each sub-task can't be processed until its upstream sub-tasks are completed and the selected service is available.

4.2. Decomposed structure of the motivating model

The first step of implementing ALC method is to partition the problem into several individual elements according to some specific rules. As autonomous decision right is important for keeping service provider's flexibility and desired level of autonomy over services, it is considered as the rule to partition the MSA problem.

The lower part of Fig. 5 shows the available services in different sub-tasks. Two alternative manufacturing services O_4^1 and O_4^2 are available in sub-task $S\text{-}T_4$ for selection. Especially, O_4^1, O_3^1, O_3^2, and O_3^3 are from the same service alliance. O_4^2, O_3^4, and O_3^5 are from the same service alliance. Both the service providers of O_4^1 and O_4^2 have independent decision rights as autonomous decision makers. They allocate services to the sub-task $S\text{-}T_3$ respectively. Meanwhile, they compete against each other and only one of them will be selected to execute $S\text{-}T_4$. In other sub-tasks, the service provider doesn't have the autonomous decision rights and only the CMfg platform has the decision rights to allocate services to them. As shown in the higher part of Fig. 5, the motivating model is partitioned into four elements according to the different decision rights, i.e. upstream element, downstream element, and two middle elements. Though $S\text{-}T_1$, $S\text{-}T_2$, $S\text{-}T_5$, and $S\text{-}T_6$ get services to be allocated by the CMfg platform, they are partitioned into two individual elements to reduce the computational complexity. The upstream element is responsible for allocating services to the most two upstream sub-tasks $S\text{-}T_1$ and $S\text{-}T_2$. The downstream element is employed to allocate services to the most two downstream sub-tasks $S\text{-}T_5$ and $S\text{-}T_6$. Two middle elements are represented by O_4^1 and O_4^2 respectively. They are used to allocate services to the middle sub-tasks (i.e. $S\text{-}T_3$ and $S\text{-}T_4$) while maintaining the autonomous decision rights of the service providers of O_4^1 and O_4^2.

4.3. Auxiliary variables and consistency constraints

The coupled relationships among the partitioned elements are depicted in Fig. 6 where upstream element and middle elements are coupled by the linking variables TC_2 and TT_2, while the downstream element and middle elements are coupled by the linking variables TC_4 and TT_4. In addition, a master element is introduced to the ALC model. It enables the partitioned elements to be solved in a parallel way while using the BCD method.

The second step of implementing ALC is to introduce auxiliary variables to separate the local constraints in each element. As auxiliary variables of TC_2 and TT_2, TC_2^U and TT_2^U are introduced at the upstream element, while TC_2^M and TT_2^M are introduced at middle elements. Similarly, as the auxiliary variables of TC_4 and TT_4, TC_4^D and TT_4^D are introduced at the downstream element, while TC_4^M and TT_4^M are introduced at middle elements. All the auxiliary variables and original variables are equal by the non-separable consistency constraints $\mathbf{c} = \left[\mathbf{c}_U^T, \mathbf{c}_D^T, \mathbf{c}_M^T\right]^T$. $\mathbf{c}_U$, $\mathbf{c}_D$ and $\mathbf{c}_M$ are the consistency constraints in the upstream element, downstream element and middle elements respectively.

$$\mathbf{c}_U = [TC_2 - TC_2^U, TT_2 - TT_2^U]^T = [0, 0]^T \tag{36}$$

$$\mathbf{c}_D = [TC_4 - TC_4^D, TT_4 - TT_4^D]^T = [0, 0]^T \tag{37}$$

$$\mathbf{c}_M = [TC_2 - TC_2^M, TT_2 - TT_2^M, TC_4 - TC_4^M, TT_4 - TT_4^M]^T = [0, 0, 0, 0]^T \tag{38}$$

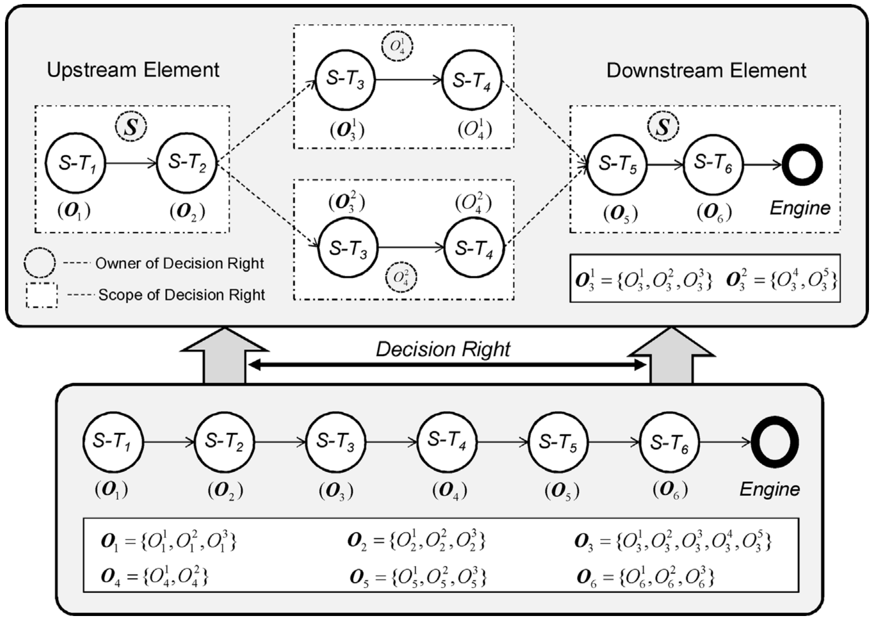

Fig. 5. Decomposed structure of the motivating model.

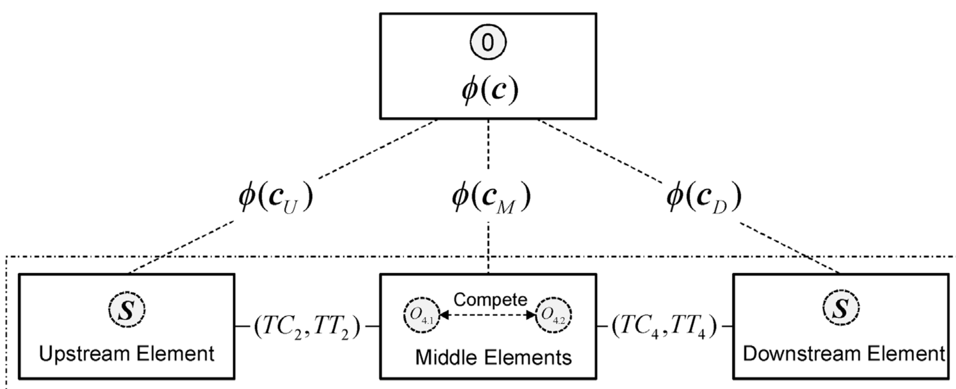

Fig. 6. ALC model of the MSA problem.

The third step of implementing ALC method is to relax the consistency constraints by augmented Lagrangian penalty function ϕ. The penalty function associated with each element can be defined as follows.

$$\phi(\mathbf{c}_U) = \mathbf{v}_U^T\mathbf{c}_U + ||\mathbf{w}_U \circ \mathbf{c}_U||_2^2 \tag{39}$$

$$\phi(\mathbf{c}_D) = \mathbf{v}_D^T\mathbf{c}_D + ||\mathbf{w}_D \circ \mathbf{c}_D||_2^2 \tag{40}$$

$$\phi(\mathbf{c}_M) = \mathbf{v}_M^T\mathbf{c}_M + ||\mathbf{w}_M \circ \mathbf{c}_M||_2^2 \tag{41}$$

Where $\mathbf{v}_U = [v_{TC_2}^U, v_{TT_2}^U]^T$, $\mathbf{v_M} = [v_{TC_2}^M, v_{TT_2}^M, v_{TC_4}^M, v_{TT_4}^M]^T$, and $\mathbf{v}_D = [v_{TC_4}^D, v_{TT_4}^D]^T$ are the vectors of Lagrange multiplier estimates for consistency constraints, while $\mathbf{w}_D = [w_{TC_4}^U, w_{TT_4}^U]^T$, $\mathbf{w}_U = [w_{TC_2}^U, w_{TT_2}^U]^T$, $\mathbf{w}_M = [w_{TC_2}^M, w_{TT_2}^M, w_{TC_4}^M, w_{TT_4}^M]^T$ are the vectors of penalty weights. After the relaxation of the consistency constraints, the local constraints of the partitioned elements are separable with respect to their variables.

4.4. Formulations of different elements

Table 2 lists the associated variables in each element which can be formulated as follows. The constraints are as the same as in MSA problem described in previous sections.

Table 2
Variables in each element.

Element	Variables
Master element	TC_2, TT_2, TC_4, TT_4
Upstream element	TC_2^U, TT_2^U, TC_1, TT_1, PC_i, PT_i, LC_i, LT_i, s_i^j (i=1,2)
Middle elements	TC_2^M, TT_2^M, TC_4^M, TT_4^M, TC_3, TT_3, PC_i, PT_i, LC_i, LT_i, s_i^j (i=3,4)
Downstream element	TC_4^D, TT_4^D, TC_i, TT_i, PC_i, PT_i, LC_i, LT_i, s_i^j (i=5,6)

1) Upstream element

Objective function $\min \phi(\mathbf{c}_U) = \mathbf{v}_U^T\mathbf{c}_U + ||\mathbf{w}_U \circ \mathbf{c}_U||_2^2$ (42)

Subject to $TC_2^U = TC_1 + PC_2 + LC_2,\ TC_1 = PC_1 + LC_1$ (43)

$$\begin{aligned} &TT_2^U = ST_2 + PT_2 + LT_2, \\ &TT_1 = ST_1 + PT_1 + LT_1 \end{aligned} \tag{44}$$

$$PC_i = \sum_{O_i^j \in O_i} s_i^j \cdot pc_i^j,\quad LC_i = \sum_{O_i^j \in O_i} s_i^j \cdot lc_i^j \tag{45}$$

$$PT_i = \sum_{O_i^j \in O_i} s_i^j \cdot pt_i^j,\quad LT_i = \sum_{O_i^j \in O_i} s_i^j \cdot lt_i^j \tag{46}$$

$$s_i^j = \begin{cases} 1 \text{ if } O_i^j \text{ is selected} \\ 0 \text{ otherwise} \end{cases} \text{ and } \sum_{O_i^j \in O_i} s_i^j = 1 \tag{47}$$

$$E_i^j \le ST_i \quad \text{if} O_i^j \text{isselected} \tag{48}$$

$$TT_1 \le ST_2,\quad i = 1, 2 \tag{49}$$

Only the penalty term is included in the objective function of the upstream element. It aims to minimize the consistency constraints $\mathbf{c}_U$ with the fixed TC_2, TT_2.

2) Middle Elements

$$\text{Objective function } \min \phi(\mathbf{c}_M) = \mathbf{v}_M^T \mathbf{c}_M + ||\mathbf{w}_M \circ \mathbf{c}_M||_2^2 \tag{50}$$

$$\text{Subject to } TC_4^M = TC_3 + PC_4 + LC_4,\quad TC_3 = TC_2^M + PC_3 + LC_3 \tag{51}$$

$$TT_4^M = ST_4 + PT_4 + LT_4,\quad TT_3 = ST_3 + PT_3 + LT_3 \tag{52}$$

$$PC_i = \sum_{O_i^j \in O_i} s_i^j \cdot pc_i^j,\quad LC_i = \sum_{O_i^j \in O_i} s_i^j \cdot lc_i^j \tag{53}$$

$$PT_i = \sum_{O_i^j \in O_i} s_i^j \cdot pt_i^j,\quad LT_i = \sum_{O_i^j \in O_i} s_i^j \cdot lt_i^j \tag{54}$$

$$s_i^j = \begin{cases} 1 \text{ if } O_i^j \text{ is selected} \\ 0 \text{ otherwise} \end{cases} \quad \sum_{O_i^j \in O_i} s_i^j = 1 \tag{55}$$

$$E_i^j \le ST_i \quad \text{if} O_i^j \text{isselected} \tag{56}$$

$$TT_2^M \le ST_3,\quad TT_3 \le ST_4,\quad i = 3, 4 \tag{57}$$

Similarly, only the penalty term is included in the objective function of middle elements. It aims to minimize the consistency constraints $\mathbf{c}_M$ with the fixed TC_2, TT_2, TC_4, and TT_4.

3) Downstream element

$$\text{Objective function } \min w_c TC_6 + w_t TT_6 + \phi(\mathbf{c}_D) = w_c TC_6 + w_t TT_6 + \mathbf{v}_D^T \mathbf{c}_D + ||\mathbf{w}_D \circ \mathbf{c}_D||_2^2 \tag{58}$$

$$\text{Subject to } TC_6 = TC_5 + PC_6 + LC_6,\quad TC_5 = TC_4^D + PC_5 + LC_5 \tag{59}$$

$$TT_i = ST_i + PT_i + LT_i \tag{60}$$

$$PC_i = \sum_{O_i^j \in O_i} s_i^j \cdot pc_i^j,\quad LC_i = \sum_{O_i^j \in O_i} s_i^j \cdot lc_i^j \tag{61}$$

$$PT_i = \sum_{O_i^j \in O_i} s_i^j \cdot pt_i^j,\quad LT_i = \sum_{O_i^j \in O_i} s_i^j \cdot lt_i^j \tag{62}$$

$$s_i^j = \begin{cases} 1 \text{ if } O_i^j \text{ is selected} \\ 0 \text{ otherwise} \end{cases} \quad \sum_{O_i^j \in O_i} s_i^j = 1 \tag{63}$$

$$E_i^j \le ST_i \quad \text{if } O_i^j \text{ isselected} \tag{64}$$

$$TT_5 \le ST_6,\quad TT_4^D \le ST_5,\quad i = 5, 6 \tag{65}$$

Two parts are included in the objective function of the downstream element. The first part is the local objective which is assigned to the downstream element. It reflects the original objective of this MSA problem. The second part is the penalty term. It aims to minimize the consistency constraints $\mathbf{c}_D$ with the fixed TC_4, TT_4.

4) Master element

$$\text{Objectivefunctionmin } \phi(\mathbf{c}) = \phi(\mathbf{c}_U) + \phi(\mathbf{c}_M) + \phi(\mathbf{c}_D) \tag{66}$$

Only penalty terms are included in the formulation of the master element. It is solved for getting TC_2, TT_2, TC_4, and TT_4, while fixing the auxiliary variables TC_2^U, TT_2^U, TC_2^M, TT_2^M, TC_4^M, TT_4^M, TC_4^D, and TT_4^D.

4.5. Solutions for the MSA problem

In this paper, solutions for the MSA problem can be categorized into two aspects. The first aspect is the coordination strategy which consists of outer loops and inner loops. In the outer loops, the penalty parameters $\boldsymbol{v} = \left[\boldsymbol{v}_U^T, \boldsymbol{v}_D^T, \boldsymbol{v}_M^T\right]^T$ and $\boldsymbol{w} = \left[\boldsymbol{w}_U^T, \boldsymbol{w}_D^T, \boldsymbol{w}_M^T\right]^T$ are updated according to Eqs. (20) and (21). In the inner loops, BCD algorithm solves the partitioned elements, while fixing the penalty parameters. The second aspect is the working mechanism for the middle elements O_4^1 and O_4^2. It can be stated as follows. O_4^1 and O_4^2 compete with each other to complete S-T_4 while maintaining their

Table 3
Information of candidate services.

Sub-task	Service information					
	O_i^j	pc_i^j	pt_i^j	lc_i^j	lt_i^j	E_i^j
1	O_1^1	16	10	5	1	3
	O_1^2	20	5	5	1	3
	O_1^3	13	25	8	2	2
2	O_2^1	405	120	18	9	10
	O_2^2	380	130	15	7	5
	O_2^3	410	100	15	7	8
3	O_3^1	40	5	8	6	110
	O_3^2	20	30	10	8	120
	O_3^3	26	16	10	8	100
	O_3^4	31	15	12	10	100
	O_3^5	24	25	8	6	110
4	O_4^1	220	60	20	16	130
	O_4^2	230	35	16	10	130
5	O_5^1	310	90	12	8	180
	O_5^2	340	60	12	8	180
	O_5^3	320	80	15	10	190
6	O_6^1	355	40	16	10	245
	O_6^2	320	70	12	7	250
	O_6^3	330	50	13	8	230

Table 4
Optimization results of AIO method and ALC method.

Sub-task	AIO method	ALC method
	Service option	Service option
$S\text{-}T_1$	O_1^1	O_1^1
$S\text{-}T_2$	O_2^3	O_2^3
$S\text{-}T_3$	O_3^1	O_3^1
$S\text{-}T_4$	O_4^2	O_4^2
$S\text{-}T_5$	O_5^2	O_5^2
$S\text{-}T_6$	O_6^3	O_6^3
Total manufacturing time	1435	1435
Total manufacturing cost	306	306
Computational time	52.291s	133.733 s (45.941 s)

autonomous decision rights to allocate service to $S\text{-}T_3$. In each iteration, the alternative elements execute their optimizations within their decision scopes and get the values of the objective function (50) respectively. The alternative with the smallest value of the objective function is regarded as the best service for $S\text{-}T_4$ in this iteration. Then, the values of TC_2^{M}, TT_2^{M},TC_4^{M}, and TT_4^{M} in the selected element will be sent to the master element. When the convergence conditions are satisfied, the selected service in the final iteration will complete $S\text{-}T_4$.

5. Results and discussions

The optimization results are discussed from the motivating case mentioned in Section 4.1 where the customer submits the demanded manufacturing requirements to the CMfg platform. Through analyzing the real-time manufacturing capabilities of services in the service pool, the CMfg platform acquires all the available services for each sub-task. Table 3 lists the candidate services which come from Guangzhou Zhaoqing Automotive parts association. Note that the data shown in Table 3 for each service is representative to keep the confidentiality of their key business.

The software MATLAB R2012b running on a PC at 2.2 GHz with 6 GB RAM is adopted to obtain the optimization results. The weight coefficients are set as $w_c = 0.3$, $w_t = 0.7$. The initial values of Lagrangian multiplier estimates are all set to 0. The initial values of penalty weights are all set to 1. TC_2, TT_2, TC_4, and TT_4 are initialized by 0. TC_2^U, TT_2^U, TC_2^M, TT_2^M,TC_4^M, TT_4^M, TC_4^D, and TT_4^D are set to 416, 116, 450, 166, 684, 175, 744, and 280. Parameters β, γ, and ε are set to 2.2, 0.5, and 0.01. The maximum number of outer loop is set to 50.

5.1. Effectiveness of ALC method in solving MSA problem

To verify the effectiveness of ALC method in solving MSA problem, an AIO method (i.e. Particle Swarm Optimization) was also tested in the same computing environment. For better comparison, the autonomous decision rights of service providers are not considered in this circumstance. Then, elements O_4^1 and O_4^2 can be treated as one middle element.

The MSA results obtained by PSO-based AIO method and ALC method are contrasted in Table 4. As it can be seen, ALC achieves the same allocation results as the AIO method with different computational time. Two different strategies are adopted to compare the computational time of these two methods. In the first strategy, ALC method is implemented on a single PC without a distributed computing environment. It takes 133.733 s to obtain the optimal results, which is much longer than the PSO-based AIO method (i.e. 52.291s). The main reason leading to this is the adopted computing mode. As a distributed method, ALC needs a distributed computing environment to implement its optimization process. However, ALC method is executed on a single PC, which could hardly show its advantage in computing efficiency. To address this reason, a distributed computing environment is constructed in the second strategy. According to ALC model of the MSA problem, it is partitioned into the master element, upstream element, middle element, and downstream element. Therefore, it can be separated onto four distributed pieces (i.e. P_1, P_2, P_3, and P_4). P_1 is to solve the master element. P_2, P_3, and P_4 are parallel pieces and employed to solve the upstream element, middle element, and downstream element respectively. Let CT_i represent the computation time for solving P_i. The whole computation time CT can be calculated as $CT = CT_1 + \max\left\{CT_2, CT_3, CT_4\right\}$. For this MSA problem, CT_1, CT_2, CT_3, and CT_4 are 0.003s, 42.354s, 45.938s, and 45.415s. CT is calculated as 45.941s. In the second strategy, ALC outperforms the AIO method. That implies the computing efficiency can be improved when ALC method is performed in a distributed computing environment.

Some limitations of this case study should be noted. Firstly, it didn't consider how other important optimization criterion would influence the MSA results. The "quality" index is vital in the production of engines. When the "quality" index is included in the objective function, the MSA results will be different. Some other important indexes could also influence the results. For example, "energy consumption" is a key criterion to evaluate the cleaner production of engines. Secondly, the example studied in this paper is based on a real-life scenario from Guangdong Zhaoqing Automotive Parts Industry Association, where a small-scale case is implemented. Manufacturing services and tasks are limited, thus, the proposed method on a large-scale example involving more candidate services should be conducted in the future.

5.2. Maintaining decision autonomy

One of the most important reasons to adopt ALC method to solve the MSA problem is its characteristic of maintaining decision autonomy. In this case, O_4^1, O_3^1, O_3^2, and O_3^3 are in the same service alliance, while O_4^2, O_3^4, and O_3^5 are in another. Service providers of O_4^1 and O_4^2 are independent decision makers. They prefer to keep the autonomous decision rights to allocate service to $S\text{-}T_3$ within their own service alliances respectively.

Table 5 shows the MSA results with and without decision autonomy of service providers. As it can be seen, service option for $S\text{-}T_3$ is different in these two circumstances. When decision autonomy of service providers is not considered in the MSA process, the optimal service options for $S\text{-}T_3$ and $S\text{-}T_4$ are O_3^1 and O_4^2 which come from different service alliance. When the decision autonomy is considered in the MSA process, the optimal service option for $S\text{-}T_3$ and

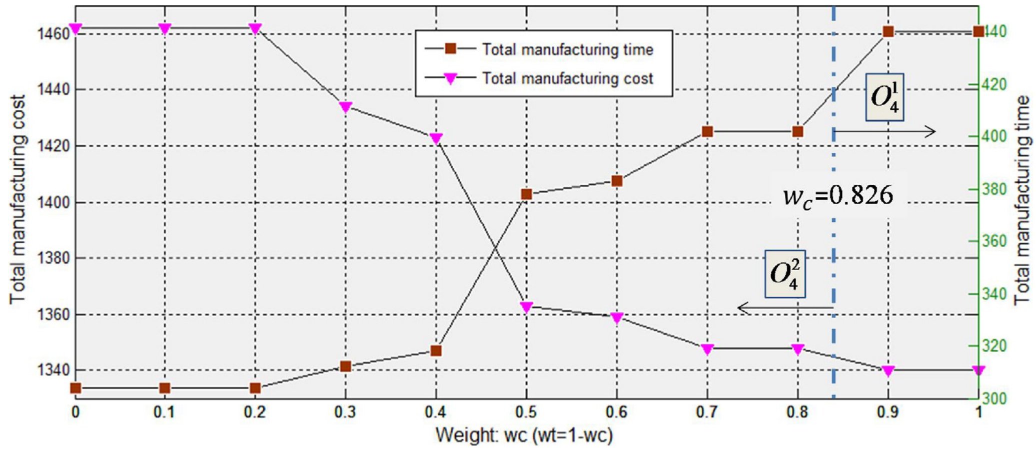

Fig. 7. Effect of weight adjustment to MSA result.

Table 5
Optimization results of with and without decision autonomy.

Sub-task	Without decision autonomy			With decision autonomy		
	Service option	TT_i	TC_i	Service option	TT_i	TC_i
$S\text{-}T_1$	O_1^1	14	21	O_1^2	9	25
$S\text{-}T_2$	O_2^3	121	446	O_2^3	116	450
$S\text{-}T_3$	O_3^1	132	494	O_3^4	141	493
$S\text{-}T_4$	O_4^2	177	740	O_4^2	186	739
$S\text{-}T_5$	O_5^2	248	1092	O_5^2	254	1091
$S\text{-}T_6$	O_6^3	306	1435	O_6^3	312	1434

$S\text{-}T_4$ are O_3^4 and O_4^2 which come from the same service alliance. It is observed that decision autonomy of service provider can be achieved by the ALC method. However, the total manufacturing time is increased from 306 to 312. That because the maintaining autonomous decision rights of the service provider of O_4^2 brings extra manufacturing time to complete $S\text{-}T_3$. O_3^4 takes 25 to complete $S\text{-}T_3$, and O_3^1 just needs 11. Though O_3^4 takes much more time than O_3^1, it saves manufacturing cost (i.e. the cost of O_3^4 is 43, and the cost of O_3^1 is 48).

Though ALC method can be helpful for decision autonomy of service provider, it will influence the open and transparent characteristics of services. Some further considerations are important. Firstly, limited information of services can be shared between service providers and the CMfg platform. In such case, the CMfg platform may not be able to provide comprehensive suggestions for improving enterprises' services. Secondly, the central management of distributed manufacturing resources in CMfg will be affected. In such case, there will be extra manufacturing cost and time for completing the tasks. Therefore, how to balance the decision autonomy of service provider and open characteristics of services should be studied in the future.

5.3. Sensitivity analysis

As the independent decision makers, service providers of O_4^1 and O_4^2 have their own service chains. They compete with each other and only one of them will be selected. The service option for $S\text{-}T_4$ will change along with the changing of weight coefficients (i.e. w_c, w_t). Meanwhile, the formation of the optimal MSA results will change accordingly. In order to find out how the changing of weight coefficients will affect the MSA results, a set of sensitivity analysis tests are conducted. In these tests, w_c ranges from 0 to 1, and $w_t = 1 - w_c$. Fig. 7 gives the MSA results from where some key observations are as follows.

(1) The descend curve shows the change of total manufacturing cost, and ascend curve shows the change of total manufacturing time. Along with the increasing of w_c, the value of total manufacturing cost falls gradually from 1462 to 1340, and the value of total manufacturing time rises from 304 to 440.

(2) The selection of O_4^1 or O_4^2 is indicated by the right arrow or left arrow respectively. As it can be seen, the selection of O_4^1 leads to lower total manufacturing cost and higher total manufacturing time. Correspondingly, the selection of O_4^2 leads to lower total manufacturing time and higher total manufacturing cost. If $w_c < 0.826$, O_4^2 will be selected; if $w_c > 0.826$, O_4^1 will be selected. Especially, if $w_c = 0.826$, the selection of O_4^1 or O_4^2 will lead to the same value of the sum of weighted manufacturing cost and time.

The managerial implications from the case study can be described as follows. For the CMfg platform, it should provide detailed guidance for assisting service transaction. For example, valuable suggestions for customers on how to set accurate weight values for their objectives are recommended so that customers can get the most suitable services for their tasks. Meanwhile, suggestions on how to improve the quality of services offered by various service providers could be worked out by the CMfg platform. For the service providers, services' quality such as reducing manufacturing cost and time is significant to enhance the competitiveness. They can put forward innovative service strategies to increase customers' satisfaction. For the customers, they should figure out what are their objectives (e.g. cost or time). If the required products have an early due time, they can set higher priority. Additionally, if customers want to save costs, they can set higher weight value for the manufacturing cost. For the cloud-based manufacturing, flexibility is one of its most important characteristics. The case study shows ALC method is feasible to be adopted in the optimal allocation of manufacturing services. In particular, it is shown that the proposed method has potential to facilitate service providers maintaining their desired level autonomy over their services and contribute to keep manufacturers' flexibility. This can promote manufacturing enterprises' shift to service-oriented business.

6. Conclusions

This paper attempts to adopt the ALC method to solve the MSA problem in cloud manufacturing. Four steps are followed: (1) decomposition of the MSA problem according to the decision rights of service providers; (2) introduction of auxiliary variables and consistency constraints to relax the decomposed individual elements; (3) ALC formulation of the relaxed elements; and (4) ALC solutions for the MSA problem. The major conclusions of this research are as follows.

- Compared with the centralized optimization strategy, the distributed strategy is much more practical because the autonomous decision rights of service providers can be maintained, which is essential for keeping their desired level of autonomy over their services.
- ALC method can not only achieve the same MSA results as AIO method but also demonstrate high computing efficiency under a distributed computing environment. Meanwhile, the effectiveness of ALC method in maintaining the autonomous decision rights of service providers is also verified.
- The observation shows the dynamic formation of MSA results along with the changing of weight coefficients. It also provides the implication for the selection of service providers with decision autonomy.

The future work may follow several aspects. Firstly, how to extend the ALC method to deal with the MSA problem when a large-scale task involving more candidate services is submitted to the CMfg platform? This paper investigates the MSA problem with a small-scale problem, but real-life case can be much more complex. Secondly, how to encapsulate the ALC method into a service and published on the CMfg platform? In CMfg, all resources can be considered as services, including the optimization methods. The encapsulation of ALC method will truly unfold its full potential in solving the MSA problem. Thirdly, how different optimization criterions influence the MSA results and how to construct a comprehensive optimization model for MSA problems should be the emphases of future research. Fourthly, how to develop a sustainable strategy to achieve the re-allocation of manufacturing service? When exceptions happen to a service which has gained one task, re-allocation is quite necessary.

Acknowledgements

The authors would like to acknowledge the financial supports of National Science Foundation of China (51675441), the 111 Project Grant (B13044), the Fundamental Research Funds for the Central Universities (3102017jc04001), and </gs4>China Scholarship Council<gs4>. Special thanks go to Guangdong Zhaoqing Automotive Parts Industry Association.

References

[1] Tao F, Zhang L, Liu Y, Cheng Y, Wang L, Xu X. Manufacturing service management in cloud manufacturing: overview and future research directions. J Manuf Sci Eng 2015;137:40912, http://dx.doi.org/10.1115/1.4030510.

[2] Zhong RY, Xu X, Klotz E, Newman ST. Intelligent manufacturing in the context of industry 4.0: a review. Front Mech Eng 2017;3:1–15.

[3] Costantino N, Dotoli M, Falagario M, Fanti MP, Mangini AM. A model for supply management of agile manufacturing supply chains. Int J Prod Econ 2012;135:451–7, http://dx.doi.org/10.1016/j.ijpe.2011.08.021.

[4] Nomden G, van der Zee DJ. Virtual cellular manufacturing: configuring routing flexibility. Int J Prod Econ 2008;112:439–51, http://dx.doi.org/10.1016/j.ijpe.2007.04.010.

[5] Tao F, Zhao D, Hu Y, Zhou Z. Resource service composition and its optimal-selection based on particle swarm optimization in manufacturing grid system. IEEE Trans Ind Informatics 2008;4:315–27, http://dx.doi.org/10.1109/TII.2008.2009533.

[6] Wang L, Haghighi A. Combined strength of holons, agents and function blocks in cyber-physical systems. J Manuf Syst 2016;40:25–34, http://dx.doi.org/10.1016/j.jmsy.2016.05.002.

[7] Wu D, Liu X, Hebert S, Gentzsch W, Terpenny J. Democratizing digital design and manufacturing using high performance cloud computing: performance evaluation and benchmarking. J Manuf Syst 2017;43:316–26, http://dx.doi.org/10.1016/j.jmsy.2016.09.005.

[8] Zhong RY, Xu C, Chen C, Huang GQ. Big data analytics for physical internet-based intelligent manufacturing shop floors. Int J Prod Res 2017;55:2610–21, http://dx.doi.org/10.1080/00207543.2015.1086037.

[9] Zhang Y, Ren S, Liu Y, Sakao T, Huisingh D. A framework for Big Data driven product lifecycle management. J Clean Prod 2017;159:229–40, http://dx.doi.org/10.1016/j.jclepro.2017.04.172.

[10] Zhang Y, Qian C, Lv J, Liu Y. Agent and cyber-physical system based self-organizing and self-adaptive intelligent shopfloor. IEEE Trans Ind Informatics 2017;13:737–47, http://dx.doi.org/10.1109/TII.2016.2618892.

[11] Zhen L. An analytical study on service-oriented manufacturing strategies. Int J Prod Econ 2012;139:220–8, http://dx.doi.org/10.1016/j.ijpe.2012.04.010.

[12] Girbea A, Suciu C, Nechifor S, Sisak F. Design and implementation of a service-oriented architecture for the optimization of industrial applications. IEEE Trans Ind Informatics 2014;10:185–96, http://dx.doi.org/10.1109/TII.2013.2253112.

[13] Xu X. From cloud computing to cloud manufacturing. Robot Comput Integr Manuf 2012;28:75–86, http://dx.doi.org/10.1016/j.rcim.2011.07.002.

[14] Wu D, Greer MJ, Rosen DW, Schaefer D. Cloud manufacturing: strategic vision and state-of-the-art. J Manuf Syst 2013;32:564–79, http://dx.doi.org/10.1016/j.jmsy.2013.04.008.

[15] Zhang Y, Zhang G, Qu T, Liu Y, Zhong RY. Analytical target cascading for optimal configuration of cloud manufacturing services. J Clean Prod 2017;151:330–43, http://dx.doi.org/10.1016/j.jclepro.2017.03.027.

[16] Laili Y, Tao F, Zhang L, Sarker BR. A study of optimal allocation of computing resources in cloud manufacturing systems. Int J Adv Manuf Technol 2012;63:671–90, http://dx.doi.org/10.1007/s00170-012-3939-0.

[17] Tao F, Zuo Y, Da Xu L, Zhang L. IoT-based intelligent perception and access of manufacturing resource toward cloud manufacturing. IEEE Trans Ind Informatics 2014;10:1547–57, http://dx.doi.org/10.1109/TII.2014.2306397.

[18] Zhang Y, Zhang G, Liu Y, Hu D. Research on services encapsulation and virtualization access model of machine for cloud manufacturing. J Intell Manuf 2017;28:1109–23, http://dx.doi.org/10.1007/s10845-015-1064-2.

[19] Cheng Y, Tao F, Zhao D, Zhang L. Modeling of manufacturing service supply-demand matching hypernetwork in service-oriented manufacturing systems. Robot Comput Integr Manuf 2017;45:59–72, http://dx.doi.org/10.1016/j.rcim.2016.05.007.

[20] Lu Y, Xu X. A semantic web-based framework for service composition in a cloud manufacturing environment. J Manuf Syst 2017;42:69–81, http://dx.doi.org/10.1016/j.jmsy.2016.11.004.

[21] Huang B, Li C, Yin C, Zhao X. Cloud manufacturing service platform for small- and medium-sized enterprises. Int J Adv Manuf Technol 2013;65:1261–72, http://dx.doi.org/10.1007/s00170-012-4255-4.

[22] Zhou J, Yao X. Multi-population parallel self-adaptive differential artificial bee colony algorithm with application in large-scale service composition for cloud manufacturing. Appl Soft Comput J 2017;56:379–97, http://dx.doi.org/10.1016/j.asoc.2017.03.017.

[23] Zhou J, Yao X. A hybrid artificial bee colony algorithm for optimal selection of QoS-based cloud manufacturing service composition. Int J Adv Manuf Technol 2017;88:3371–87, http://dx.doi.org/10.1007/s00170-016-9034-1.

[24] Liu Y, Zhang L, Tao F, Wang L. Resource service sharing in cloud manufacturing based on the Gale?Shapley algorithm: advantages and challenge. Int J Comput Integr Manuf 2017;30:420–32, http://dx.doi.org/10.1080/0951192X.2015.1067916.

[25] Thekinen J, Panchal JH. Resource allocation in cloud-based design and manufacturing: a mechanism design approach. J Manuf Syst 2016;43:327–38, http://dx.doi.org/10.1016/j.jmsy.2016.08.005.

[26] Li H, Chan KCC, Liang M, Luo X. Composition of resource-Service chain for cloud manufacturing. IEEE Trans Ind Informatics 2016;12:211–9, http://dx.doi.org/10.1109/TII.2015.2503126.

[27] Zhang Y, Wang J, Liu S, Qian C. Game theory based real-Time shop floor scheduling strategy and method for cloud manufacturing. Int J Intell Syst 2017;32:437–63, http://dx.doi.org/10.1002/int.21868.

[28] Liu W, Liu B, Sun D, Li Y, Ma G. Study on multi-task oriented services composition and optimisation with the Multi-Composition for Each Task pattern in cloud manufacturing systems. Int J Comput Integr Manuf 2013;26:786–805, http://dx.doi.org/10.1080/0951192X.2013.766939.

[29] Zheng H, Feng Y, Tan J. A fuzzy QoS-aware resource service selection considering design preference in cloud manufacturing system. Int J Adv Manuf Technol 2016;84:371–9, http://dx.doi.org/10.1007/s00170-016-8417-7.

[30] Cheng Y, Tao F, Liu Y, Zhao D, Zhang L, Xu L. Energy-aware resource service scheduling based on utility evaluation in cloud manufacturing system. Proc Inst Mech Eng Part B-J Eng Manuf 2013;227:1901–15, http://dx.doi.org/10.1177/0954405413492966.

[31] Xiang F, Hu Y, Yu Y, Wu H. QoS and energy consumption aware service composition and optimal-selection based on Pareto group leader algorithm in cloud manufacturing system. Cent Eur J Oper Res 2014;22:663–85, http://dx.doi.org/10.1007/s10100-013-0293-8.

[32] Xu W, Tian S, Liu Q, Xie Y, Zhou Z, Pham DT. An improved discrete bees algorithm for correlation-aware service aggregation optimization in cloud manufacturing. Int J Adv Manuf Technol 2016;84:17–28, http://dx.doi.org/10.1007/s00170-015-7738-2.

[33] Lartigau J, Xu X, Nie L, Zhan D. Cloud manufacturing service composition based on QoS with geo-perspective transportation using an improved Artificial Bee Colony optimisation algorithm. Int J Prod Res 2015;53:4380–404, http://dx.doi.org/10.1080/00207543.2015.1005765.

[34] Liu Y, Xu X, Zhang L, Wang L, Zhong RY. Workload-based multi-task scheduling in cloud manufacturing. Robot Comput Integr Manuf 2017;45:3–20, http://dx.doi.org/10.1016/j.rcim.2016.09.008.

[35] Tosserams S, Etman LFP, Rooda JE. An augmented Lagrangian decomposition method for quasi-separable problems in MDO. Struct Multidiscip Optim 2007;34:211–27, http://dx.doi.org/10.1007/s00158-006-0077-z.

[36] Tosserams S, Etman LFP, Rooda JE. Augmented Lagrangian coordination for distributed optimal design in MDO. Int J Numer Methods Eng 2008;73:1885–910, http://dx.doi.org/10.1002/nme.2158.

[37] Qu T, Nie DX, Chen X, Chen XD, Dai QY, Huang GQ. Optimal configuration of cluster supply chains with augmented Lagrange coordination. Comput Ind Eng 2015;84:43–55, http://dx.doi.org/10.1016/j.cie.2014.12.026.

[38] Allison JT, Papalambros PY. Consistency constraint allocation in augmented lagrangian coordination. J Mech Des 2010;132:71007, http://dx.doi.org/10.1115/1.4001525.

[39] Tosserams S, Etman LFP, Papalambros PY, Rooda JE. An augmented Lagrangian relaxation for analytical target cascading using the alternating direction method of multipliers. Struct Multidiscip Optim 2006;31:176–89, http://dx.doi.org/10.1007/s00158-005-0579-0.