sustainability

Article

Multi-Objective Optimization of Service Selection and Scheduling in Cloud Manufacturing Considering Environmental Sustainability

Dong Yang, Qidong Liu *, Jia Li and Yongji Jia

School of Business and Management, Donghua University, Shanghai 200051, China; yangdong@dhu.edu.cn (D.Y.); jia.li@mail.dhu.edu.cn (J.L.); yjjia@dhu.edu.cn (Y.J.)

* Correspondence: 2180983@mail.dhu.edu.cn

Received: 25 August 2020; Accepted: 16 September 2020; Published: 18 September 2020

Abstract: Cloud manufacturing is an emerging service-oriented paradigm that works by taking advantage of distributed manufacturing resources and capabilities to collaboratively perform a manufacturing task, with the consideration of QoS (Quality of Service) requirements such as cost, time and quality. Incorporating environmental concerns and sustainability into cloud manufacturing to produce a much greener product has become an urgent issue since there is fierce market competition and an increasing environment consciousness from customers. In this paper, we present a multi-objective optimization approach to selecting and scheduling cloud manufacturing services from the viewpoints of the economy and environment including carbon emissions and water resource. Subject to the carbon cap regulation, a multi-objective model for a cloud manufacturing task is built with the aim of minimizing total costs, carbon emissions, and water resource use. Transportation mode selections and carbon emissions from both cloud manufacturing services and transportation activities are taken into account in this model. The ε-constraint method is employed to obtain the exact Pareto front of optimal solutions. A case study from automobile cloud manufacturing is used to illustrate the effectiveness of the presented approach. Numerical experiments are conducted to compare the presented approach and the simple additive weighting method. The results show that the presented ε-constraint method can obtain a better and more diverse Pareto set of solutions and that it can solve the models in a reasonable time.

Keywords: cloud manufacturing; multi-objective optimization; carbon emissions; sustainable manufacturing

1. Introduction

With the development of modern information technology, fierce market competition, and diverse customer needs, cloud manufacturing has emerged as a convenient manufacturing mode for organizing global manufacturing resources and capabilities to perform a specific manufacturing task by means of cloud computing [1]. Cloud manufacturing can be regarded as a specific kind of SaaS (Software as a Service), i.e., manufacturing as a service, and users can use cloud manufacturing services based on a pay-as-you-go mode [2]. Under the paradigm of cloud manufacturing, enterprises encapsulate their manufacturing resources into cloud manufacturing services with standard information description, namely, virtualization of manufacturing resources [1,2]. Then, the manufacturing services are published on a cloud manufacturing platform. Once a user submits the request for a manufacturing task, the cloud platform performs service discovery, service matching, and service selection and scheduling such that the user requirements for QoS (Quality of Service) like cost and due date are satisfied. A vast number of studies on cloud manufacturing have been carried out during recent years [3,4] on issues like service discovery [5–7], service matching [8–10], and service selection and scheduling [11–13]. Among these, the service selection and scheduling problem, which is the problem of assigning candidate cloud manufacturing services to each subtask and scheduling them with specified start and end times, is most challenging in cloud manufacturing [4]. Owing to various QoS requirements for user tasks, such as cost, due date, and reliability, service selection and scheduling is a multi-objective optimization problem; generally, in most studies, a simple additive weighting (SAW) method is used to convert a multi-objective optimization problem to a single objective one [14,15]. Researchers have proposed a variety of QoS like makespan, cost, reliability of service, and pass rate in specifying their optimization objectives. However, existing research on cloud manufacturing rarely considers the sustainability of cloud manufacturing, especially environmental impacts of cloud manufacturing services [16]. Consequently, there exists a significant gap between cloud manufacturing and sustainable manufacturing. In this paper, we propose a multi-objective optimization approach for optimally selecting and scheduling cloud manufacturing services according to task requirement from a user and with consideration of environmental impacts, i.e., greenhouse gas emissions and water resource, in cloud manufacturing services. We further consider the carbon cap (i.e., limitation) regulation [17,18], one of main policies to reduce CO_2 emissions under the United Nations Framework Convention on Climate Change (UNFCCC) and the Paris Agreement, in the optimization model as a hard constraint that restricts the total carbon emissions for all selected cloud manufacturing services so that they cannot exceed the specified amount of CO_2 emissions. Unlike existing studies that use a simple additive weighting to solve multi-objective optimization, we further develop an exact method to obtain the Pareto-optimal front by using the ε-constraint method [19] without a pre-determined weight for each objective.

Our contributions are three-fold. Firstly, carbon emissions by both manufacturing services and transportation activities between the services are taken into account when optimally assigning cloud services to subtasks, subject to the carbon cap regulation. As far as we know, few studies address the carbon emissions in cloud manufacturing within the framework of the carbon cap regulation. Secondly, minimization of water resources when configuring cloud manufacturing tasks is another contribution. Typical manufacturing processes for cloud services have to consume a lot of water during handling, cleansing, and lubricating activities. To make cloud manufacturing sustainable, taking into account water resources during manufacturing tasks is indispensable. Finally, all previous researches regarding cloud manufacturing either assume pre-determined weights for each objective or make an effort to derive a near Pareto front for the multi-objective optimization using heuristic algorithms such as NSGA-II. In contrast, we can obtain an exact Pareto-optimal front by means of the developed ε-constraint method, thereby ensuring rigorous decisions for decision makers.

The paper is organized as follows. Section 2 reviews related work. In Section 3, the service selections and scheduling problem in cloud manufacturing in relation to carbon emissions and sustainability considerations is described, and a multi-objective mixed-integer programming model is built. Section 4 addresses the ε-constraint method algorithm for solving the multi-objective optimization model. A case study on automobile cloud manufacturing is used to illustrate the effectiveness of the presented approach. Numerical experiments are conducted to compare the presented approach and the simple additive weighting method. Finally, Section 6 summarizes conclusions and future research directions.

2. Literature Review

The studies on cloud manufacturing mainly include service discovery, service matching, and service selection and scheduling. Below, the literatures about the three aspects are elaborated. For a comprehensive review, readers can be referred to the literatures in [3,14].

Due to a vast number of enterprises involved in a cloud manufacturing platform, a mechanism of dynamically discovering the encapsulated services is needed to match the services with the submitted subtasks. Although there is ambiguity between service discovery and service matching and the two terms are used interchangeably in some literature, we distinguish the two. Service discovery is the means of dynamically discovering the cloud manufacturing services in terms of the functions that a subtask requires, whereas service matching means matching a service to a subtask in terms of non-function attributes. To discover the cloud manufacturing services dynamically, formal representation of functions and inputs and outputs of services are requisite, considering various slight variations in the information representation provided by various services. A knowledge representation such as ontology acts as one of the main means to formalize the functions and inputs and outputs of a service. Wang et al. (2014) proposed a modeling framework of cloud manufacturing tasks using ontology to formally express the classification of the manufacturing tasks and modeling requirements [5]. Zhang et al. (2016) implemented a proactive service discovery by using OWL (Ontology Web Language) to represent function attributes of services [6]. With regard to service matching, it aims to find suitable services from the discovered candidate service with the same functions by comparing their non-functional attributes, such as product quality and delivery time. Li et al. (2019) developed a two-side matching model for non-function attributes of services such as product quality and supply capacity using dual hesitant fuzzy sets [8]. Tao et al. (2009) presented a framework for integrating service search and service matching by considering similarity degrees between services and subtasks. The similarity degree is calculated by summing the basic similarities in function, input and output, and other non-functional attributes [9]. Liu and Chen (2019) proposed an approach to recommending cloud manufacturing services by clustering and similarity-based recommending. The similarity degree takes into account both task similarity and QoS similarity [10].

After service discovery and matching is finished, a number of legitimate cloud manufacturing services are available, and they differ in QoS requirements such as costs, prices, and time. Consequently, how to select the services, assign the services to the subtasks, and schedule them in a timely manner to satisfy the QoS requirements of user tasks becomes a challenging problem [4]. Due to the fact that a cloud manufacturing task (i.e., a user) generally has various service requirements like due date, cost, reliability, and reputation, service selecting and scheduling is regarded as a multi-objective optimization problem in which it is impossible to achieve all optimal results simultaneously in multiple objectives, and an improvement in one objective may lead to the deterioration in another objective. To solve the multi-objective problem, the main methods in existing studies include the simple additive weighting (SAW) method [11] and meta-heuristics algorithms, like the genetic algorithm (GA) [20] and ant colony optimization (ACO) [21]. The studies mainly differ in the chosen criteria for measuring QoS, like cost, time, reputation, energy usage, and pass rate. Akbaripour employed SAW to solve the multi-objective manufacturing service selection problem with a pre-determined weight for each objective, and an exact mathematical programming solver GAMS was used to obtain optimal results [11]. Aghamohammadzadeh et al. (2019) presented an approach to composing and selecting cloud service providers in a cloud manufacturing environment by using cloud-entropy, an index for measuring the complexity of the composed cloud services, and other objectives like cost and time [20]. They solved the multi-objective optimization model using Non-dominated sorting genetic algorithm II (NSGA_II). Li et al. (2018) considered multi-task scheduling in cloud manufacturing services by optimizing multiple objectives including makespan, time, and quality of service such that the dependencies among subtasks of a task were observed [22]. Moreover, both ACO and NSGA_II were employed to solve the multi-objective model, and the experiments showed that ACO can obtain a more diverse set of Pareto solution, thus offering more selection alternatives for customers. Zhang et al. (2019) constructed a comprehensive cloud manufacturing platform for injection molding, and the optimal service selections were derived using GA [23].

However, logistics and transportation as important activities in cloud manufacturing are ignored in the literature until recently [4,11,14]. Lartigau et al. (2015) proposed to take into account the transportation activities between physical locations of cloud services when composing a cloud manufacturing task [21]. Liu et al. (2016) proposed a multi-task cloud manufacturing model for assigning multiple tasks to cloud manufacturing services including logistic time by satisfying the QoS requirements for tasks [24]. Zhou et al. (2020) investigated a logistics scheduling problem for cloud services to reduce the average delivery time from manufacturers to customers [25].

It becomes gradually recognized that cloud manufacturing should increase its sustainability by reducing its environmental impacts, reducing its waste, and reusing and recycling renewable resources [16]. Nevertheless, there exist a significant gap between cloud manufacturing and sustainable manufacturing. The only paper considering carbon emissions in cloud manufacturing is that by He et al. (2019) [26]. Our study is different from their studies, which only address carbon emissions in manufacturing activities, because we further take into account carbon emissions in transportation activities between cloud manufacturing services, namely carbon footprint from manufacturing to transportation. Moreover, we give a hard carbon emission cap, i.e., carbon cap regulation, for total carbon emissions when selecting, composing, and scheduling cloud manufacturing services. The reason is that an increasing number of countries around the world have imposed carbon emission regulations like carbon caps and carbon taxes [17] since the Paris Agreement. Therefore, it is instructive to consider carbon regulations in the studies on cloud manufacturing. To keep cloud manufacturing environmentally friendly and sustainable, minimization of water resource usage in cloud manufacturing tasks is taken into account in our study.

3. Problem Description

A cloud manufacturing framework consists of a central cloud manufacturing platform, small and medium-sized enterprises (SMEs) offering various cloud manufacturing services, and manufacturing task requests from users, as shown in Figure 1. An enterprise encapsulates its manufacturing services or capabilities, registers them to the cloud manufacturing platform, and then publishes them. All registered and published services are put into the service pool managed by the platform. Once a user submits its manufacturing task to the platform, the task is decomposed into a set of the subtasks with precedence relations. The platform is responsible for discovering and matching candidate manufacturing services for the corresponding subtasks. Then, the services are selected from the candidate service set and assigned to the subtasks. The start and end time of each subtask are determined and scheduled according to the user's required due date. It is assumed that a task has been decomposed into the subtasks and that services can be discovered and matched in accordance with the well-known method mentioned in the literature. Therefore, our study only focuses on the service selection and scheduling problem.

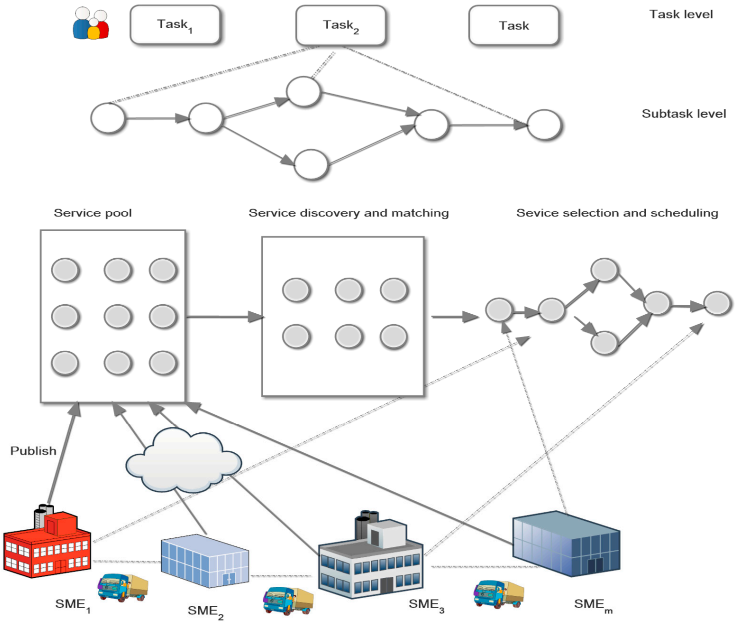

Figure 1. The framework of a cloud manufacturing platform.

Suppose a manufacturing task TA is submitted to the cloud manufacturing platform and is decomposed into a set of the subtasks ST_i $(i = 1, 2, \cdots, I)$, namely, $TA = \{ST_1, ST_2, \cdots, ST_i, \cdots\}$. An enterprise SME_j $(j = 1, 2, \cdots, J)$ can provide several cloud manufacturing services $MS = \left\{MS_{j1}, MS_{j2}, \cdots, MS_{jk}, \cdots\right\}$ where MS_{jk} represents kth manufacturing service by the jth enterprise. The transportation activities and between SMEs are considered, and there exist several transportation modes m $(m = 1, 2, \ldots.)$ for transporting a product from one service site to another site. For example, land transportation and air transportation have different transportation costs and per-unit carbon emissions. Although air transportation has relatively higher transportation costs and carbon emissions when compared to land transportation, it offers much shorter transportation time. The carbon footprint from manufacturing to transportation is also taken into account. It is required by the user that the total carbon emissions for the configured cloud services cannot exceed the carbon allowance e^{cap}, which is specified according to an annual plan for total carbon amount for this enterprise. The aim of this study is to select and assign the cloud manufacturing services to the subtasks with the multiple objectives of minimizing total cost, carbon emissions, and water usage.

4. Cloud Manufacturing Service Selection and Scheduling Model and Solution Method

4.1. Assumptions

(1) The task has been decomposed into the subtasks.
(2) Service discovery and matching has been carried out beforehand.
(3) A subtask cannot be interrupted once it begins to execute.
(4) Carbon cap (i.e., limitation) is known and given by the user according to a yearly planning for the reduction of CO_2 emissions.
(5) Each cloud manufacturing service provides its environment and resource data in its service description, such as CO_2 emissions in manufacturing activities and water usage in production processes.
(6) We focus on discrete cloud manufacturing tasks where CO_2 emissions by electricity account for most of greenhouse gas emissions, although in continuous cloud manufacturing such as biomass products, other greenhouse gases such as NO_2 and Sulfur dioxide should be dealt with.

4.2. Notations

SC_{ijk}	Service cost of service MS_{jk}, if assigned to subtask ST_i;
ST_{ijk}	Service time of service MS_{jk}, if assigned to subtask ST_i;
Q_{ijk}	Service quality of service MS_{jk}, if assigned to subtask ST_i;
WA_{ijk}	Amount of water used by service MS_{jk}, if assigned to subtask ST_i;
PC_{ijk}	Setup cost of service MS_{jk}, if assigned to subtask ST_i;
em_{ijk}	Unit carbons emission of manufacturing activities by MS_{jk}, if assigned to subtask ST_i;
et_m	Unit carbons emission by transportation activity using transportation mode m;
e^{cap}	Maximum carbon emission cap;
CT_m	Unit transportation cost using transportation mode m;
BAT	Batch of manufacturing;
$Dist_{jp}$	Geographical distance between enterprise j and enterprise p;
QA_{min}	Minimum acceptable quality level specified by the user;
W_i	Weight of a semi-finish product for subtask ST_i;
$TT^m_{ijk,lpk'}$	Transportation time between service MS_{jk} of subtask i and $MS_{pk'}$ of subtask l; using transportation mode m;
SOT_{jk}	Beginning time for time window of cloud service MS_{jk};
EOT_{jk}	End time for time window of cloud service MS_{jk};
TD	Due date for a cloud manufacturing task;
x_{ijk}	=1, if subtask i is assigned to cloud manufacturing service MS_{jk}; = 0, otherwise;
$y^m_{ijk,lpk'}$	=1, if transportation model m is selected for the transportation between service MS_{jk} of subtask i and $MS_{pk'}$ of subtask l;
T_i	Start time of subtask ST_i.

4.3. Model Formulation

4.3.1. Decision Variables

Three types of decisions needs to be made in the service selection and scheduling problem in cloud manufacturing. The variable x_{ijk} represents task assignment decisions and is defined as:

$$x_{ijk} = \begin{cases} 1, \ if\ subtask\ i\ is\ assigned\ to\ cloud\ manufacturing\ service\ MS_{jk} \\ 0, \ Otherwise \end{cases} \quad (1)$$

The logistic decisions for transportation means are denoted by the variable $y^{m}_{ijk,lpk'}$ which is defined as follows:

$$y^{m}_{ijk,lpk'} = \begin{cases} 1, \ if\ transportation\ model\ m\ is\ selected\ for\ shippping\ between \\ \quad service\ MS_{jk}\ of\ subtask\ i\ and\ service\ MS_{pk'}\ of\ subtask\ l \\ 0, \ Otherwise \end{cases} \quad (2)$$

The scheduling decisions determine the start time of each subtask such that the delivery date can be met. The variable T_i defines the start time of each subtask in the problem, namely,

$$T_i \geq 0 \quad (3)$$

4.3.2. Objectives

(1) Total cost

The total cost consists of service costs, setup costs of cloud manufacturing services, and transportation costs between two consecutive manufacturing services. The service cost and setup cost of a cloud manufacturing service is formulated as:

$$SC_{ijk}BAT + PC_{ijk}$$

where SC_{ijk} and PC_{ijk} represent service cost and setup cost of cloud manufacturing service MS_{jk}, respectively, and BAT is batch of the manufacturing. The transportation cost between two consecutive manufacturing services MS_{jk} and $MS_{pk'}$ is calculated as:

$$Dist_{jp}CT_m W_i y^{m}_{ijk,lpk'}$$

where $Dist_{jp}$, CT_m, and W_i are the geographical distance between enterprise j and enterprise p, unit transportation cost, weight of transported semi-product or product for subtask i, respectively. $y^{m}_{ijk,lpk'}$ denotes transportation model m is selected for shipment between service MS_{jk} of subtask i and $MS_{pk'}$ of subtask l. Therefore, the total cost (TOC) in cloud manufacturing is formulated as:

$$TOC = \sum_i \sum_j \sum_k \left(SC_{ijk}BAT + PC_{ijk}\right)x_{ijk} + \sum_i \sum_{(i,l)\in PR} \sum_m Dist_{jp}CT_m W_i y^{m}_{ijk,lpk'} \quad (4)$$

where PR is the set of a pair of subtasks with immediate precedence relations.

(2) Total carbon emissions

The total carbon emissions in cloud manufacturing include the CO_2 emissions in manufacturing services and those in transportation activities between them. The CO_2 emissions of a manufacturing service MS_{jk} in processing, cutting, and forming material can be calculated as:

$$em_{ijk}BATx_{ijk}$$

where em_{ijk} represents per-unit carbons emission for manufacturing activities MS_{jk} performing subtask i, and BAT is the batch of manufacturing. The CO_2 emissions by transportation activity between two manufacturing services are formulated as:

$$Dist_{jp}et_mW_iy^m_{jkpk'}$$

where et_m represent per-unit carbons emissions by transportation activity with transportation mode m. Consequently, the total carbon emissions (TCE) are calculated as:

$$TCE = \sum_i\sum_j\sum_k em_{ijk}BATx_{ijk} + \sum_i \sum_{(jk,pk')\in PR} \sum_m Dist_{jp}et_mW_iy^m_{ijk,lpk'} \quad (5)$$

(3) Water resource

The manufacturing processes within a cloud manufacturing services need to consume a lot of water in processing, lubricating, and cooling activities. The total amount of water usage (TAW) in cloud manufacturing amounts to:

$$TAW = \sum_i\sum_j\sum_k WA_{ijk}BATx_{ijk} \quad (6)$$

where WA_{ijk} denotes the amount of water consumed by service MS_{jk} to perform subtask i.

(4) Multi-objective optimization

With the three objective expressions for total cost, total carbon emissions, and total water usage, the multi-objective optimization for sustainable cloud manufacturing can be formulated as:

$$\begin{gathered}\text{Minimize } f_1 : TOC \\ \text{Minimize } f_2 : TCE \\ \text{Minimize } f_3 : TAW\end{gathered}$$

Before conducting multi-objective optimization, some constraints need to be considered in this study.

4.3.3. Constraints

(1) Subtask assignment constraint

A subtask is only assigned to a cloud manufacturing service to carry out the manufacturing task. The assignment constraints are formulated as follows:

$$\sum_{j=1}^{J}\sum_{k=1}^{K} x_{ijk} = 1\ \forall i \quad (7)$$

(2) Precedence constraint

The tasks with the precedence relations must be scheduled such that a succeeding task can only be executed if all its proceeding tasks have been finished and the corresponding semi-finished product has been shipped. The precedence constraint is described as:

$$T_p - T_i \geq \sum_{j=1}^{J}\sum_{k=1}^{K} x_{ijk}ST_{ijk} + + \sum_{j=1}^{J}\sum_{k=1}^{K}\sum_m TT^m_{ijk,pj'k'}y^m_{ijk,pj'k'}\ \forall p,\ \forall i \in PRE(p) \quad (8)$$

where ST_{ijk} is the service time of service MS_{jk} w.r.t. subtask ST_i and PRE is the set of all preceding task of the task p.

(3) Time window constraint

We consider in this study that a cloud manufacturing service can specify its service time window by the start and end time of the service to avoid waiting and time conflicts between customers, owing to the fact that a cloud service could serve multiple customers. Consequently, a subtask can only be assigned to a service if its start time and finish time fall in the time window. The constraints can be represented as:

$$\sum_{j=1}^{J}\sum_{k=1}^{K} SOT_{jk}x_{ijk} \leq T_i \ \forall i \tag{9}$$

$$T_i + \sum_{j}^{J}\sum_{k}^{K} ST_{ijk}x_{ijk} \leq \sum_{j=1}^{J}\sum_{k=1}^{K} EOT_{jk}x_{ijk} \ \forall i \tag{10}$$

where SOT_{ijk} and EOT_{jk} are the start time and end time of a time window by cloud service MS_{jk}.

(4) Service quality constraint

Service quality is calculated by considering average pass rate for the selected cloud manufacturing services. It should satisfy the minimal pass rate specified by a user task.

$$\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{J}\sum_{k=1}^{K} Q_{ijk}x_{ijk} \geq QA_{min} \tag{11}$$

where Q_{ijk} is the service quality of service MS_{jk} w.r.t. subtask ST_i.

(5) Due date constraint

The finish time of the final subtask must satisfy the delivery time of a user task. The constraint is elaborated as follows:

$$T_N + \sum_{j}^{J}\sum_{k}^{K} ST_{Njk}x_{ijk} \leq T_{max} \tag{12}$$

where T_N represents the start time of the final subtask. T_{max} is the due date for the submitted user task.

(6) Carbon cap regulation

The carbon cap regulation enforces that the total amount of CO_2 emission must not exceed a specified carbon allowance. The constraint is described as follows:

$$TCE \leq e^{cap} \tag{13}$$

(7) Budget constraint

The service cost of composite services can be less than or equal to the budget specified by a user task. The budget constraint is formulated as:

$$TOC \leq C_{max} \tag{14}$$

(8) Logical constraint

Generally there exist logical constraints between decision variables. The logical constraints between transportation decisions and subtask assignment decisions should be imposed to ensure that a transportation mode can only be selected if the corresponding services are assigned to the two consecutive subtasks.

$$\sum_{m=1}^{M} y_{ijk,lpk'}^{m} = x_{ijk}x_{lpk'} \ \forall (i,l) \in PR, \ \forall j \forall p \forall k$$

Due to the non-linear term, i.e., multiplication of two decision variables, in this formula, a linearization technique must be adopted in this study to ensure that mathematical programming can be used to solve the mathematical model. An auxiliary binary variable $z_{rkk'}$ is introduced and defined below.

$$z_r = x_{ijk}x_{lpk'} \tag{15}$$

$$z_r \leq x_{ijk} \tag{16}$$

$$z_r \leq x_{lpk'} \tag{17}$$

$$z_r \leq x_{ijk} + x_{lpk'} - 1 \tag{18}$$

$$z_r \in \{0,1\} \tag{19}$$

$$\sum_{m=1}^{M} y_{ijk,lpk'}^{m} = z_r \tag{20}$$

4.3.4. Multi-Objective Optimization Model

With decision variables and constraints defined above, the multi-objective optimization model for service selection and scheduling in cloud manufacturing is formulated as:

$$\begin{array}{ll} (\mathrm{MO-SSS}): & \\ & \text{Minimize } f_1 : TOC \\ & \text{Minimize } f_2 : TCE \\ & \text{Minimize } f_3 : \ TAW \\ \text{s.t.} & \\ & \text{Contraints } (7)-(20) \\ & x_{ijk},\ y_{ijk,lpk'}^{m} = 0,\ 1 \\ & T_i \geq 0 \end{array} \tag{21}$$

where the objectives TOC, TCE, and TAW represent total cost, total carbon emissions, and total amount of water usage, respectively, as defined in Equations (4)–(6).

4.4. Solution Algorithm for Multi-Objective Optimization

The aim of multi-objective optimization is to obtain Pareto-optimal solutions for the concerned problem, owing to the nonexistence of a solution that has all optimal values in every objective for the minimization problem. A Pareto-optimal solution set is defined to be the set of all non-dominated solutions. A solution x is called non-dominated (for minimization problem), if it satisfies the following inequalities:

$$\forall y\ f_i(x) \leq f_i(y)\ \forall i \in \{1,2,\ldots,o_1\} \text{ and } \exists j \in \{1,2,\ldots,o_2\},\ f_j(x) < f_j(y)$$

where $f_i(x)$ is the ith objective function to be optimized. In other words, a solution is a non-dominated one if it is not much worse in all objectives and has one better value in at least one objective than all other solutions.

There exist several limited studies on cloud manufacturing which apply a pre-determined additive weighting method (SAW) for multi-objective optimization. The advantage of the Pareto-optimal based method is that decision makers can dynamically make optimal polices according to the real situation without a pre-determined weight. In this study, we employ the ε-constraint method [19] to obtain exact Pareto-optimal solutions, which differs from NSGA-II with approximate Pareto-optimal methods.

In the ε-constraint method, one of multiple objectives is selected as the main objective to be optimized, and other objectives are transformed into the constraints with corresponding ε as the upper bounds of the constraints. As a consequence, a multiple objective optimization problem can be handled by iteratively solving a serial of single-objective optimization problems by varying the values of ε such that the optimal Pareto-front can be derived. With the ε-constraint method, the model $(\mathrm{MO-SSS})$ for service selection and scheduling in this study can be converted to a single objective optimization problem with other objectives moving down to constraints. The transformed model is formulated as the model $\mathrm{MO-SSS}(\varepsilon)$:

$$\begin{array}{c} \mathrm{MO-SSS}(\varepsilon): \\ \text{Minimize } f_1 : TOC \\ \text{s.t.} \\ f_2 = TCE \le \varepsilon_2 \\ f_3 = TAW \le \varepsilon_3 \\ \text{Contraints } (7)-(20) \\ x_{ijk},\ y^{m}_{ijk,lpk'} = 0,\ 1 \\ T_i \ge 0 \end{array} \tag{22}$$

where the optimized objective is f_1, i.e., total cost, and the objectives f_2 and f_3, i.e., total carbon emissions and water resource, are converted as corresponding constraints with specified upper bounds ε_1 and ε_2, respectively. By varying the values of ε_1 and ε_2 according to a step size, a serial of single-objective optimization problems can be solved, and thus the optimal Pareto front can be obtained.

To determine the range of the value of ε_2 and ε_3, the ideal points and nadir points of the objectives f_2 and f_3 should be obtained, and they correspond to the minimal values and maximal values of f_2 and f_3, respectively. The ideal points of f_1, f_2 and f_3, i.e., f_1^I, f_2^I and f_3^I, can be derived by solving the following linear programming problems:

$$\begin{array}{ll} (M1): & f_1^I = \text{Minimize } TOC \\ & \text{s.t.} \\ & \text{Contraints } (7)-(20) \\ (M2): & f_2^I = \text{Minimize } TCE \\ & \text{s.t.} \\ & \text{Contraints } (7)-(20) \\ (M3): & f_3^I = \text{Minimize } TAW \\ & \text{s.t.} \\ & \text{Contraints } (7)-(20) \end{array}$$

Similarly, by adding the constraints restricting other objectives equal to their ideal points, the nadir points of f_2 and f_3, i.e., f_2^N and f_3^N, can be obtained by solving the following linear programming problems:

$$\begin{array}{ll} (N1): & f_1^N = \text{Minimize } TOC \\ & \text{s.t.} \\ & f_2 = f_2^I \\ & f_3 = f_3^I \\ & \text{Contraints } (7)-(20) \\ (N2): & f_2^N = \text{Minimize } TCE \\ & \text{s.t.} \\ & f_1 = f_1^I \\ & f_3 = f_3^I \\ & \text{Contraints } (7)-(20) \\ (N3): & f_3^N = \text{Minimize } TAW \\ & \text{s.t.} \\ & f_1 = f_1^I \\ & f_2 = f_2^I \\ & \text{Contraints } (7)-(20) \end{array}$$

Based on the solutions above, we can obtain the range of ε_2 taking values, namely, the interval $[f_2^I,\ f_2^N]$. Similarly, the range of ε_3, i.e., $[f_3^I,\ f_3^N]$, can be determined. The complete ε-constraint algorithm for solving service selection and scheduling in cloud manufacturing is described as follows (Algorithm 1):

Algorithm 1. The ε-constraint algorithm for solving multi-objective optimization

```
Step 1. Obtain the ideal points of the objectives $f_2$ and $f_3$ by solving the models M1-M3.
Step 2. Obtain the nadir points of the objectives $f_2$ and $f_3$ by solving the models N1-N3.
Step 3. Set the step size $\Delta$ as an initial value. Initialize the set of Pareto-optimal set $\Omega = \phi$.
Step 4. While $\varepsilon_2 \le f_2^N$
                Add Constraint $f_2 = TCE \le \varepsilon_2$ to model $\mathrm{MO-SSS}$;
                    While $\varepsilon_3 \le f_3^N$
                        Add constraints $f_3 = TAW \le \varepsilon_3$;
                        Solve the model $\mathrm{MO-SSS}(\varepsilon)$ and obtain $f_1(\varepsilon),\ f_2(\varepsilon),\ f_3(\varepsilon)$;
                Add the triple tuple $(f_1(\varepsilon),\ f_2(\varepsilon),\ f_3(\varepsilon))$ to the set $\Omega$;
                    Update $\varepsilon_3 \leftarrow \varepsilon_3 + \Delta$;
            End
              Update $\varepsilon_2 \leftarrow \varepsilon_2 + \Delta$;
      End
Step 5. Output the Pareto-optimal set $\Omega$.
```

As described above, the algorithm begins with obtaining ideal points and nadir points of each objective, respectively. The step size Δ should be set and initialized. When ε_2 and ε_3 are less than or equal to their nadir values (i.e., upper bounds for the minimization problem), new constraints, i.e., ε-constraints, are added and thus the model $\mathrm{MO-SSS}(\varepsilon)$ is derived. By solving this model, its optimal objective $f_1(\varepsilon)$ can be obtained and the expressions $f_2(\varepsilon),\ f_3(\varepsilon)$ can be calculated. A triple tuple $(f_1(\varepsilon),\ f_2(\varepsilon),\ f_3(\varepsilon))$ is added to the Pareto-optimal set Ω. Then the step size is updated and the process iterates until the corresponding condition does not hold. Finally, the Pareto-optimal set Ω, i.e., Pareto front, is derived.

5. Computational Experiments and Results

5.1. Case Study for a Small Example

To verify the effectiveness of the suggested models and solution algorithms, the production of automobile engine parts including valve (VAL), Exhaust Gas Re-circulation (EGR) passage, crankcase (CRK), gear housing (GHS), and oil pan (OIP) from reference [27] is used an example. The manufacturing task is decomposed into five subtasks. Overall, the task has a sequential structure. SMEs offer cloud manufacturing services for performing corresponding subtasks. It is possible that a subtask may be performed by the services of several SMEs with differentiated QoS, and a SME offers several cloud manufacturing services for different subtasks. Table 1 shows the data about SMEs and their cloud services, including the manufacturing cost, time, pass rate, CO_2 emissions, water usage, and time windows of each service. As shown in the table, 13 enterprises can offer one or two cloud manufacturing services with specified service data for corresponding subtasks. For example, SME3 offers two cloud manufacturing services for the subtasks EGR and GHS, respectively. The geographical distances between enterprises are shown in Table A1 (see Appendix A). Other parameters for the manufacturing example include cost budget, delivery deadline, quality level, and carbon emission factors and are listed in Table 2. Two kinds of transportation models, i.e., $m = 2$, such as land and air transportation are assumed in the example.

With the data as an input, the multi-objective model for cloud manufacturing, i.e., $\mathrm{MO-SSS}(\varepsilon)$ in (22), is solved using the ε-constraints method, and the model is programmed in Cplex C++ 12.7.0

with the computational time 817.69 s. The optimization results, including all Pareto-optimal solutions, are shown in Table 3. For example, the solution R4 indicates that the objective values for cost, carbon emissions, and water usage are 265,341, 14,683.7, and 1212, respectively, if the five subtasks are assigned to SME7, SME3, SME4, SME13, and SME12, respectively, and the transportation mode 2 is selected from SME7 to SME3, and mode 1 is used for shipping between other SMEs. There are 18 non-dominated solutions, as shown in Table 3, and decision makers can select one of the solutions as their decisions according to their preference with regards to economic, environmental, and water resources. One can selects the solution R1 if the minimal cost is preferably considered, whereas the solution R7 should be adopted if the decision makers focus on the least CO_2 emissions for cloud manufacturing. Figure 2 graphically displays the scheduling results for the two candidate solutions, namely solution R1 with the minimal cost and solution R7 with the least CO_2 emissions.

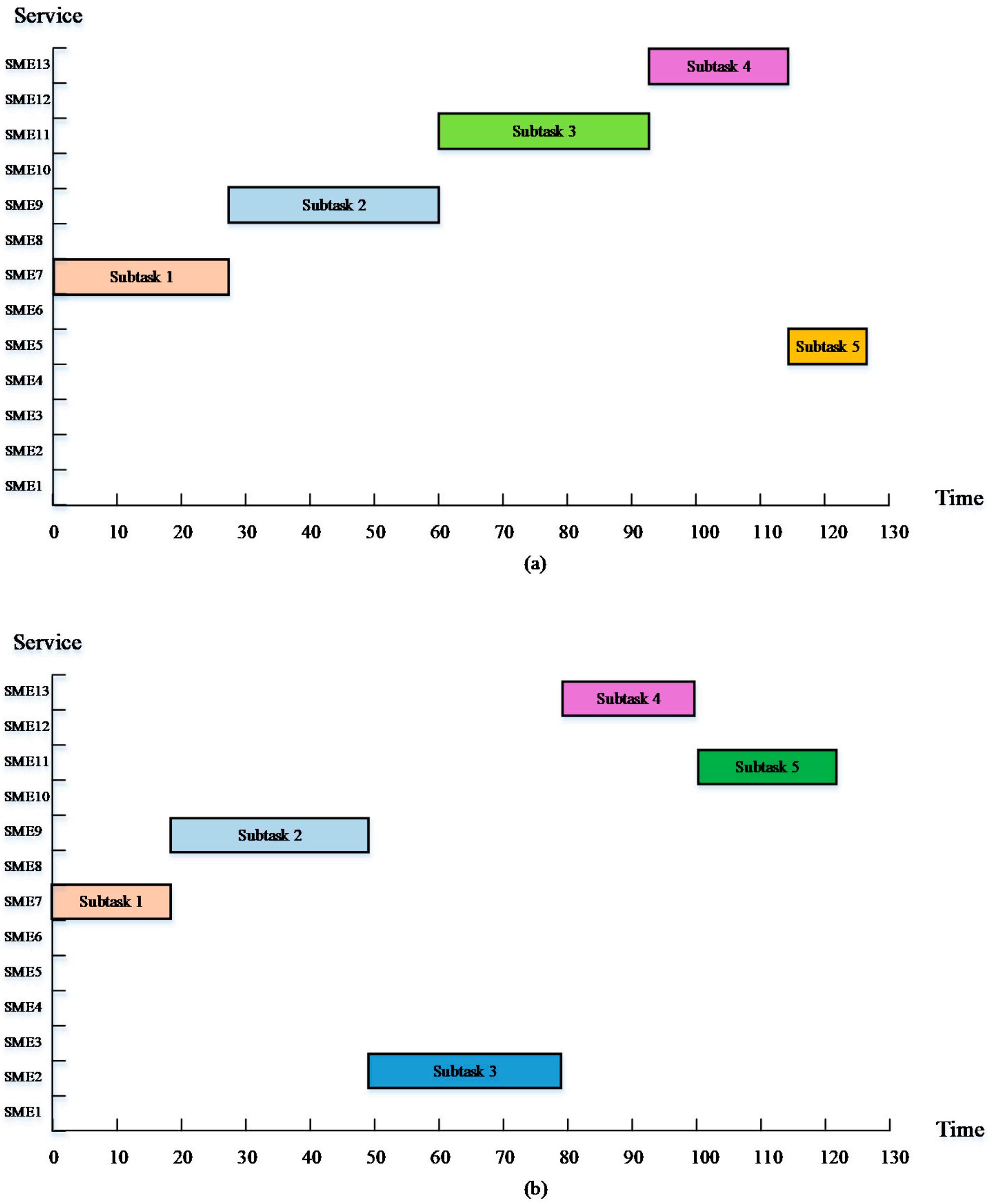

Figure 2. Alternative service selection and scheduling results: (**a**) Solution R1; (**b**) Solution R7.

Table 1. Service providers and their cloud manufacturing services for the example case.

SME	Time Windows	Manufacturing Cost(RMB)/Manufacturing Time/Pass Rate/Carbon Emissions(kgCO_2-e)/Water Usage (ton) for Five Subtasks				
		VAL	EGR	CRK	GHS	OIP
SME1	[20,100]		506/34/0.80/41.39/3.35			
SME2	[0,80]	161/20/0.81/21.4/1.81		1165/22/0.98/31.07/4.14		
SME3	[30,150]		561/24/0.89/30.82/2.5		311/30/0.87/45.03/4.61	
SME4	[60,100]			1120/28/0.95/34.96/4.69		
SME5	[10,150]			917/32/0.83/54.32/6.42		76/12/0.89/10.59/0.27
SME6	[30,120]		532/31/0.82/35.66/3.5			67/17/0.85/13.59/0.45
SME7	[0,50]	176/17/0.87/16.55/1.07			416/16/0.96/28.02/3.37	
SME8	[30,80]			1138/25/0.97/33.96/4.5		77/11/0.91/11.53/0.25
SME 9	[10,60]		585/20/0.92/30.81/2.17			75/14/0.88/13.23/0.31
SME 10	[30,100]	172/19/0.83/17.52/1.34			352/20/0.90/40.73/3.92	
SME 11	[0,150]	116/28/0.80/23.31/2.05		1007/30/0.87/43.28/5.78		55/25/0.80/15.64/0.57
SME12	[0,120]	139/24/0.82/22.68/1.96				63/20/0.84/14.75/0.49
SME13	[50,120]				387/18/0.93/35.74/3.59	

Table 2. Parameters for clouding manufacturing task in the example.

Parameters	Values
C_{max}	300,000
T_{max}	150
QA_{min}	0.90
BAT	100
W_i	{1.2, 5.0, 20.0, 30.0, 1.0}
$TC_m(m = 1, 2)$	{0.0069, 0.0104}
$TE_m(m = 1, 2)$	{0.0006, 0.0008}

Table 3. Optimal service selections.

Solutions	**(*f*1, *f*2, *f*3)**	Cloud Services and Transportation Selections for Subtasks VAL-EGR-CRK-GHS-OLP
R1	(256,583, 15,069.8, 1266)	SE7-(1)-SE9-(1)-SE11-(1)-SE13-(1)-SE5
R2	(260,477, 14,279.4, 1187)	SE7-(1)-SE9-(1)-SE4-(1)-SE13-(1)-SE11
R3	(265,333, 14,542.4, 1179)	SE7-(1)-SE9-(1)-SE4-(1)-SE13-(1)-SE12
R4	(265,341, 14,683.7, 1212)	SE7-(2)-SE3-(1)-SE4-(1)-SE13-(1)-SE12
R5	(265,974, 14,467.1, 1168)	SE7-(2)-SE9-(1)-SE8-(1)-SE13-(1)-SE11
R6	(266,458, 15,004.7, 1220)	SE7-(2)-SE1-(2)-SE2-(1)-SE13-(1)-SE5
R7	(268,137, 14,121.6, 1132)	SE7-(2)-SE9-(1)-SE2-(1)-SE13-(1)-SE11
R8	(268,994, 14,892.4, 1212)	SE7-(1)-SE3-(1)-SE4-(2)-SE13-(1)-SE12
R9	(269,403, 14,915.8, 1212)	SE7-(2)-SE3-(1)-SE4-(2)-SE13-(1)-SE12
R10	(269,480, 14,198.3, 1132)	SE7-(1)-SE9-(2)-SE2-(1)-SE13-(1)-SE11
R11	(271,039, 15,009.3, 1212)	SE7-(2)-SE3-(1)-SE4-(1)-SE13-(2)-SE12
R12	(272,993, 14,384.6, 1124)	SE7-(2)-SE9-(1)-SE2-(1)-SE13-(1)-SE12
R13	(274,334, 14,810.0, 1190)	SE7-(2)-SE3-(1)-SE4-(2)-SE13-(1)-SE5
R14	(277,925, 14,278.8, 1102)	SE7-(2)-SE9-(1)-SE2-(1)-SE13-(1)-SE5
R15	(279,267, 14,355.5, 1102)	SE7-(1)-SE9-(2)-SE2-(1)-SE13-(1)-SE5
R16	(284,636, 14,802.4, 1135)	SE7-(1)-SE3-(2)-SE2-(2)-SE13-(1)-SE5
R17	(284,803, 14,671.8, 1102)	SE7-(2)-SE9-(2)-SE2-(2)-SE13-(1)-SE5
R18	(286,774, 14,784.5, 1102)	SE7-(1)-SE9-(2)-SE2-(1)-SE13-(2)-SE5

5.2. Comparison with the Weighted Sum Approach

To investigate with the effectiveness of the presented approach, the ε-constraints method is compared with simple additive weighting (SAW) approach. SAW is to assign a weight to each objective such that a multi-objective optimization problem can be converted into a single objective one. For cloud manufacturing, multi-objective optimization for service selection and scheduling with SAW approach is represented below.

$$
\begin{array}{ll}
(\mathrm{MO-SAW}): & \\
& \text{Minimize } w_1 f_1' + w_2 f_2' + w_3 f_3' \\
\text{s.t.} & \\
& \text{Contraints } (7)-(20) \\
& x_{ijk},\ y_{ijk,lpk'}^{m} = 0,\ 1 \\
& T_i \geq 0
\end{array}
\quad (23)
$$

where f_1', f_2', f_3' are the normalized objectives for cost, carbon emissions, and water usage, respectively. They are defined as $f_f' = \frac{f_r - f_r^{min}}{f_r^{max} - f_r^{min}}$ $(r = 1, 2, 3)$ where f_r^{min} and f_r^{max} are minimal and maximal values of the objective f_r, respectively. w_1, w_2, and w_3 are corresponding weights for these three objectives. The decision variables are the same with the model in (21).

The process of solving the multi-objective optimization model by the simple additive weighting (SAW) method is depicted in Figure 3. The process begins with the initialization of the weights, i.e., $w_1 = w_2 = 0, w_3 = 1$. During each iteration, the weights are updated by increasing or decreasing a step size and the model is then solved by the commercial solver Cplex 12.7.0. The optimal solution is added to the Pareto set. When one of weights is larger than 1, the process ends and the Pareto-optimal set is obtained. For the small cloud manufacturing example mentioned above, the Pareto-optimal solutions by SAW with equal step size of weights are listed in Table 4. The coverage ratio is defined as N_w/N_ε, where N_w is the number of the Pareto-optimal solutions by SAW and N_ε is that of the Pareto-optimal solutions by the ε-constraints method. As shown in the table, the Pareto-optimal solutions by SAW can only obtain seven solutions, i.e., (R1, R2, R7, R8, R11, R14, R18). By contrast, the ε-constraints method can derive all 18 solution from R1 to R18, which have been shown in Table 3. The maximal coverage ratio achieves its maximal value of 38.89% even if step size of the weights becomes smaller than 0.01. Figure 4 graphically shows all Pareto fronts, i.e., Pareto-optimal points, by both the ε-constraints and SAW. Obviously, the ε-constraint approach finds all eighteen solutions and SAW can obtain only seven solutions of these solutions. Therefore, the ε-constraints method can achieve more Pareto-optimal points and more diverse Pareto distributions than the simple weighted sum approach.

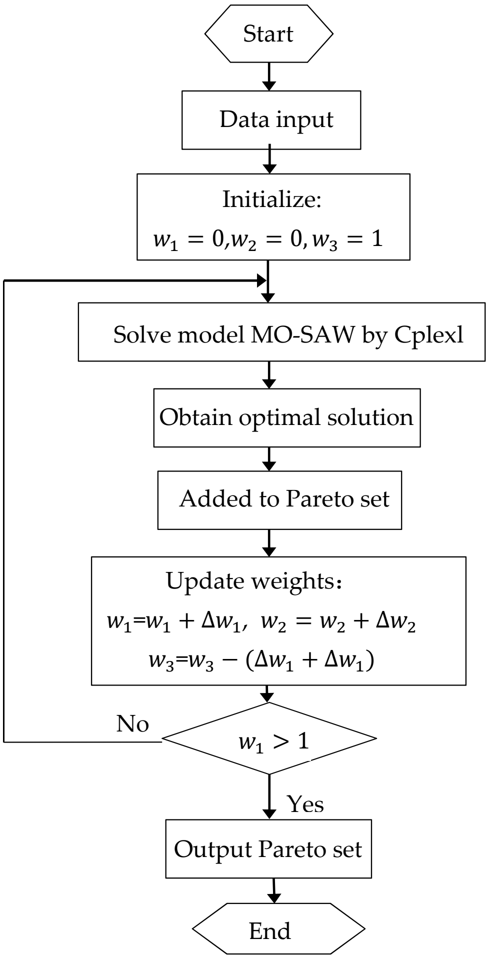
Start
Data input
Initialize:
$w_1 = 0, w_2 = 0, w_3 = 1$
Solve model MO-SAW by Cplexl
Obtain optimal solution
Added to Pareto set
Update weights：
$w_1 = w_1 + \Delta w_1,\ w_2 = w_2 + \Delta w_2$
$w_3 = w_3 - (\Delta w_1 + \Delta w_1)$
No
$w_1 > 1$
Yes
Output Pareto set
End

Figure 3. Multi-objective optimization by simple weighted sum (SAW).

Table 4. Pareto-optimal solutions by simple additive weighting (SAW).

Step Size (Δw_1=Δw_2)	Solutions	Coverage Ratio	CPU Time (s)
0.1	R1, R2, R7, R14, R18	27.78%	8.64
0.05	R1, R2, R7, R14, R18	27.78%	28.87
0.01	R1, R2, R7, R8, R11, R14, R18	38.89%	476.69
0.005	R1, R2, R7, R8, R11, R14, R18	38.89%	1764.84
0.001	R1, R2, R7, R8, R11, R14, R18	38.89%	9865.64

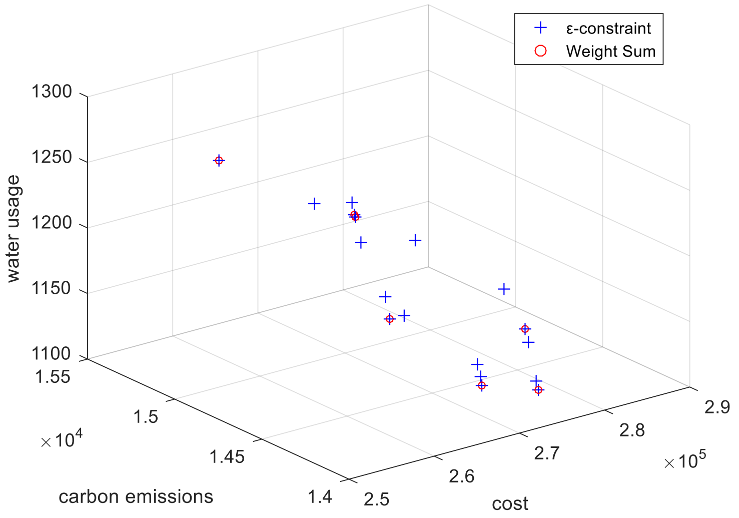

Figure 4. Pareto-optimal fronts by the two approaches.

To investigate the effectiveness and efficiency of the presented approach under medium-sized or larger-sized cases, numerical experiments are conducted to compare the ε-constraint method and SAW. Each case is denoted by case ID, namely $I * J * K$, where I, J, K represent the numbers of subtasks, SMEs, and services offered by a SME, respectively. We assume that the number of available cloud services provided by a SME is the same for all SMEs. By generating the numbers randomly, we can obtain the same case ID with different case data. Service costs and transportation cost per unit obey uniform distributions U[50, 1000] and U[0.005, 0.02], respectively. Carbon emissions and water usage are subject to normal distribution N[10, 48] and N[0.5, 6.5], respectively. For each case, 10 experiments are randomly generated and the average coverage ratio and computational times are derived based on the 10 experiments. Table 5 lists the experimental results for the ε-constraint method and SAW with different scales of cases. The number of subtask varies from 5 to 20, and the number of SMEs takes a value from 10 to 20. For possible cloud services that a SME could provide, the value varies from 2 to 4 since in reality a SME only focuses one or several core services. It can be seen from the table that the coverage ratio by SAW is only 24.7~68.4% of that derived by the ε-constraint method, whatever the step size of weight update is. Consequently, the ε-constraint method can obtain more Pareto-optimal points than SAW within a reasonable time for most of the cases. However, the computational results cannot be solved within 3600 s. when the numbers of both subtask and SMEs are equal to or greater than 20. Therefore, there is a potential improvement in applying multi-objective evolutionary algorithms such as NSGA-II in solving the large-sized problems.

Table 5. Experimental comparisons between ε-constraint and SAW.

I	**J**	**K**	**CPU Time (s)**			**Coverage**	
			ε-Method	**SAW Δw=0.05**	**SAW Δw=0.01**	**SAW Δw=0.05**	**SAW Δw=0.01**
5	10	2	10.636	17.109	258.573	68.4%	68.4%
5	15	3	22.58	29.444	887.396	37.5%	37.5%
5	20	4	96.823	95.98	3587.88	46.2%	51.3%
10	10	2	9.983	12.943	245.637	37.5%	37.5%
10	15	3	28.872	32.346	960.424	52.6%	60.5%
10	20	4	173.506	156.972	4950.34	35.2%	59.2%
15	10	2	12.399	18.979	336.286	47.6%	57.1%
15	15	3	91.151	98.179	1066.91	27.4%	39.3%
15	20	4	541.17	585.507	4587.88	31.7%	37.6%
20	10	2	40.464	41.887	1134.71	33.3%	43.6%
20	15	3	263.787	267.358	7473.9	27.3%	38.6%
20	20	4	962.17	842.641	- [1]	24.7%	-

[1] The symbol '-' means that the computational results cannot be derived in 3600 s.

6. Conclusions

For service selection and scheduling in cloud manufacturing, an exact approach for solving multi-objective optimization is proposed in this study with the consideration of the sustainability of cloud manufacturing. Especially, the carbon emissions of both cloud manufacturing services and transportation activities are taken into account in the optimization model. Furthermore, we consider the decisions on transportation modes since in fact several transportation models such as highway, land, air transportation, and third-party logistics are available for cloud service enterprises, which has an effect on the quality of manufacturing services. Additionally, water usage, a large quantity of which may be consumed in manufacturing processes, is incorporated into the optimization model to keep cloud manufacturing resource conservative. The multi-objective model is solved with the ε-constraint approach, which can generate an exact Pareto front for the concerned problem and achieve much better and diverse Pareto solutions than the simple weighted sum approach. Experimental results show that it is effective and efficient for small or medium-size problem instances and can solve them in a reasonable time. Nevertheless, for large-scale problems, applying meta-heuristic algorithms such as NSGA-II should be addressed in further study.

Another research direction which is worth exploring is the uncertainty in service time and transportation time of a cloud service when the selection and scheduling in cloud manufacturing is addressed.

Author Contributions: D.Y., Q.L., and J.L. conceived and designed the research; D.Y. and J.L. conducted the research and drafted the manuscript. Q.L. designed the computational experiments and Y.J. revised the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the MOE (Ministry of Education in China) Project of Humanities and Social Sciences under Grant No. 18YJA630129, the National Natural Science Foundation of China under Grant No. 71971053, 71832001 and the Shanghai philosophy and Social Sciences Planning Project under Grant No. 2018BGL002.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Table A1. Geographical distances between SMEs.

SME	1	2	3	4	5	6	7	8	9	10	11	12	13
1	0	43	1417	215	106	755	743	858	1049	1699	158	1044	686
2	43	0	1460	187	134	727	715	830	1039	1690	201	1034	722
3	1417	1460	0	1357	1432	1818	975	1466	2103	2765	1251	668	778
4	215	187	1357	0	299	633	540	663	929	1579	277	892	580
5	106	134	1432	299	0	852	836	942	1084	1735	175	1080	714
6	755	727	1818	633	852	0	846	447	329	979	900	1176	1065
7	743	715	975	540	836	846	0	511	1134	1796	750	334	354
8	858	830	1466	663	942	447	511	0	702	1351	929	813	854
9	104	1039	2103	929	1084	329	1134	702	0	672	1199	1465	1336
10	169	1690	2765	1579	1735	979	1796	1351	672	0	1850	2127	1986
11	158	201	1251	277	175	900	750	929	1199	1850	0	347	347
12	104	1034	668	892	1080	1176	334	813	1465	2127	347	0	542
13	686	722	778	580	714	1065	354	854	1336	1986	347	542	0

References

1. Wu, D.; Greer, M.J.; Rosen, D.W.; Schaefer, D. Cloud manufacturing: Strategic vision and state-of-the-art. *J. Manuf. Syst.* **2013**, *32*, 564–579. [CrossRef]
2. Xu, X. From cloud computing to cloud manufacturing. *Robot. Comput. Integr. Manuf.* **2012**, *28*, 75–86. [CrossRef]
3. Tao, F.; Zhang, L.; Liu, Y.; Cheng, Y.; Wang, L.; Xu, X. Manufacturing Service Management in Cloud Manufacturing: Overview and Future Research Directions. *ASME J. Manuf. Sci. Eng.* **2015**, *137*, 040912. [CrossRef]
4. Liu, Y.; Wang, L.; Wang, X.V.; Xu, X.; Zhang, L. Scheduling in cloud manufacturing: State-of-the-art and research challenges. *Int. J. Prod. Res.* **2019**, *57*, 4854–4879. [CrossRef]
5. Wang, T.R.; Guo, S.S.; Lee, C.G. Manufacturing task semantic modeling and description in cloud manufacturing system. *Int. J. Adv. Manuf. Technol.* **2004**, *71*, 2017–2031. [CrossRef]
6. Zhang, Y.X.; Li, R.D. Task-driven manufacturing cloud service proactive discovery and optimal configuration method. *Int. J. Adv. Manuf. Technol.* **2016**, *84*, 29–45. [CrossRef]
7. Yuan, M.; Cai, X.; Zhou, Z.; Sun, C.; Gu, W.; Huang, J. Dynamic service resources scheduling method in cloud manufacturing environment. *Int. J. Prod. Res.* **2019**. [CrossRef]
8. Li, B.; Yang, Y.; Su, J.; Zhang, N.; Wang, S. Two-sided matching model for complex product manufacturing tasks based on dual hesitant fuzzy preference information. *Knowl.-Based Syst.* **2019**, *186*, 104989. [CrossRef]
9. Tao, F.; Hu, Y.F.; Zhao, D.M.; Zhou, Z.D. Study on resource service match and search in manufacturing grid system. *Int. J. Adv. Manuf. Technol.* **2009**, *43*, 379–399. [CrossRef]
10. Liu, J.; Chen, Y. A personalized clustering-based and reliable trust-aware QoS prediction approach for cloud service recommendation in cloud manufacturing. *Knowl.-Based Syst.* **2019**, *174*, 43–56. [CrossRef]
11. Akbaripour, H.; Houshmand, M.; Woensel, T.V.; Mutlu, N. Cloud manufacturing service selection optimization and scheduling with transportation considerations: Mixed-integer programming models. *Int. J. Adv. Manuf. Technol.* **2018**, *95*, 43–70. [CrossRef]
12. Zhang, W.; Yang, Y.; Zhang, S.; Yu, D.; Xu, Y. A new manufacturing service selection and composition method using improved flower pollination algorithm. *Math. Probl. Eng.* **2016**. [CrossRef]
13. Cao, Y.; Wang, S.L.; Kang, L.; Gao, Y. ATQCS-based service selection and scheduling strategy in cloud manufacturing. *Int. J. Adv. Manuf. Technol.* **2016**, *82*, 235–251. [CrossRef]
14. Bouzary, H.; Chen, F. Service optimal selection and composition in cloud manufacturing: A comprehensive survey. *Int. J. Adv. Manuf. Technol.* **2018**, *97*, 795–808. [CrossRef]
15. Chen, F.Z.; Dou, R.L.; Li, M.Q.; Wu, H. A flexible QoS-aware web service composition method by multi-objective optimization in cloud manufacturing. *Comput. Ind. Eng.* **2016**, *99*, 423–431. [CrossRef]
16. Fisher, O.; Watson, N.; Porcu, L.; Bacon, D.; Rigley, M.; Gomes, R. Cloud manufacturing as a sustainable process manufacturing route. *J. Manuf. Syst.* **2018**, *47*, 53–68. [CrossRef]
17. Carl, J.; Fedor, D. Tracking global carbon revenues: A survey of carbon tax versus cap-and-trade in the real world. *Energy Policy* **2016**, *96*, 50–77. [CrossRef]
18. Chelly, A.; Nouira, I.; Frein, Y.; Hadj-Alouane, A.B. On The consideration of carbon emissions in modelling-based supply chain literature: The state of the art, relevant features and research gaps. *Int. J. Prod. Res.* **2019**, *57*, 4977–5004. [CrossRef]
19. Mavrotas, G. effective implementation of the ε-constraint method in multi-objective mathematical programming problems. *Appl. Math. Comput.* **2009**, *213*, 455–465. [CrossRef]
20. Aghamohammadzadeh, E.; Malek, M.; Valilai, O.F. A novel model for optimisation of logistics and manufacturing operation service composition in Cloud manufacturing system focusing on cloud-entropy. *Int. J. Prod. Res.* **2019**, *58*, 1987–2015. [CrossRef]
21. 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*, 4380–4404. [CrossRef]
22. Li, F.; Zhang, L.; Liao, T.W.; Liu, Y. Multi-objective optimisation of multi-task scheduling in cloud manufacturing. *Int. J. Prod. Res.* **2019**, *57*, 3847–3863. [CrossRef]
23. Zhang, Y.; Xi, D.; Yang, H. Cloud manufacturing based service encapsulation and optimal configuration method for injection molding machine. *J. Intell. Manuf.* **2019**, *30*, 2681–2699. [CrossRef]

24. Liu, Y.; Xu, X.; Zhang, L.; Tao, F. An Extensible Model for Multi-Task Service Composition and Scheduling in a Cloud Manufacturing System. *J. Comput. Inform. Sci. Eng.* **2016**, *16*, 041009. [CrossRef]
25. Zhou, L.; Zhang, L.; Fang, Y. Logistics service scheduling with manufacturing provider selection in cloud manufacturing. *Robot. Comput. Integ. Manuf.* **2020**, *65*, 101914. [CrossRef]
26. He, W.; Jia, G.; Zong, H.; Kong, J. Multi-Objective Service Selection and Scheduling with Linguistic Preference in Cloud Manufacturing. *Sustainability* **2019**, *11*, 2619. [CrossRef]
27. Liang, H.; Wen, X.; Liu, Y.; Zhang, H.; Zhang, L.; Wang, L. Logistics-involved QoS-aware service composition in cloud manufacturing with deep reinforcement learning. *Robot. Comput. Integr. Manuf.* **2021**, *67*, 101991. [CrossRef]