J. Electrical Systems 19-4 (2023):418-434

[1] Ali Vaziri

[2]Pardis Sadatian Moghaddam

[3]Anita Ershadi Oskouei

Development of Service Compositions in Cloud Manufacturing Processes Based on System Sustainability Components

Journal of Electrical Systems

Abstract: - With the emergence of information and communication technologies, manufacturing industries have shifted from centralized production architectures to architectures based on global services. This shift is highly beneficial for identifying emerging customer needs in the market, such as rapid response to custom production, autonomy in organizational network resources, flexibility, and efficiency in manufacturing. Consequently, several leading countries have given special attention to cloud manufacturing in their national development programs. Notable examples include the Industrial Internet program in the United States, the Industry 4.0 framework in Germany, and the Made in China 2025 initiative. Cloud manufacturing adopts the cloud computing paradigm as a foundation for providing shared manufacturing services on demand. The result is customer-centric supply chains that can be tailored based on cost, quality, speed, and customization.

In this study, considering cloud manufacturing systems, the economic, social, and environmental sustainability components were evaluated and analyzed. A sustainable mathematical model was developed and extended in this field. In the conducted analysis, using a small-scale approach with a metaheuristic algorithm, optimal solutions for economic, social, and environmental sustainability were evaluated. Finally, using the NSGA-II metaheuristic algorithm, larger dimensions of the problem were studied and analyzed. In the performed analysis, the algorithm parameters were initially tuned using the Taguchi method, followed by an evaluation and analysis of the problem dimensions based on exact and metaheuristic solutions. It was demonstrated that the development of cloud manufacturing systems is highly practical and executable within analytical models.

Keywords: Service Composition, Cloud Manufacturing Process, System Sustainability Components

1. Introduction

With the advent of information and communication technology, manufacturing industries have transitioned from centralized production models to a service-oriented global architecture (Hamidi, 2021). This shift has significantly enhanced the ability to identify emerging customer demands at various market levels, enabling faster responses to customized production, increased autonomy in organizational network resources, as well as improved adaptability and efficiency (Song, 2023).Consequently, some advanced countries have focused on large-scale production in their national development programs. Among these are the Industrial Internet Program in the United States, the Fourth Industrial Revolution initiative in Germany, and the 2025 Manufacturing Plan in Germany. Large-scale production emphasizes a cloud-based approach as a fundamental framework for delivering shared manufacturing services and adopting a demand-driven strategy. The outcome of this approach involves customer-centric supply chains, which can optimize costs, quality, speed, and customization (Wang, 2022). While current emphasis is on the technical capabilities required for large-scale production, numerous questions remain regarding the positive and negative impacts on consumer behavior or support for large-scale production services. In the presence of security, welfare, and user experience in cloud production environments, human factors can play a significant role in the success and adoption of large-scale production (Song, 2022) .

With the expansion of the concept of cloud computing to include all production items and characteristics, it can be perceived as a singular service. The main issue in today's construction and production space revolves around the supply and demand of service-oriented models. In service production systems (such as large-scale production),

[1] School of Systems and Enterprises, Stevens Institute of Technology, Hoboken, NJ, USA.

avaziri1@stevens.edu

[2]Computer Science Department Georgia State University Atlanta, Georgia, USA

psadatian1@student.gsu.edu

[3]School of Systems and Enterprises, Stevens Institute of Technology, Hoboken, NJ, USA.

aershadi@stevens.edu

Copyright © JES 2024 on-line: journal.esrgroups.org

solving issues related to supply-demand alignment for various resources and capabilities as services remains a primary requirement for maintaining system effectiveness and efficiency (Wang, 2022). Particularly, the current state of the universal cloud environment and its social implications have made the issues of aligning supply and demand more complex and diverse (Salmasnia, 2023). Moreover, in service production systems, all production activities and inter-company collaborations are based on service demand alignment, which still presents opportunities for development. The results of such issues (supply-demand alignment) serve as practical guidelines for companies to create collaborative activities and theoretical bases for evaluating the impact of company collaborations. One of the challenges related to network models is how to distribute and share production resources, which in this context highlights the transition from product-based to service-based production. To facilitate service-oriented systems and address challenges in network production, cloud computing presents new ideas for researchers (Hu, 2022).

Cloud manufacturing is a new concept that is currently under development. It is a model that provides networked, on-demand access to a shared pool of configurable manufacturing resources. For example, software tools for production, manufacturing equipment, and manufacturing capabilities that can be quickly provisioned and released. In essence, cloud manufacturing is a type of service-oriented and network-based manufacturing model(Rashidifar, 2023). In cloud manufacturing systems, each company can make its resources and capabilities available as cloud manufacturing services to other companies or consumers. These services can be accessed, invoked, and deployed in target companies. Different users can search for these services, call upon and combine them according to their needs through different manufacturing clouds, and either form a virtual manufacturing enterprise or complete a part of their own production process using the services provided in this manner (Xie, 2021).

On the other hand, existing analyses regarding the competition between small and medium-sized enterprises (SMEs) and market-leading companies indicate that SMEs have been able to compete with major market players by using a shared environment and forming strategic alliances with each other. This alliance can be realized by sharing each SME's capabilities through information and communication technology in a space known as cloud computing .

The ultimate goal of developing a cloud manufacturing model is to respond to customer needs by optimizing interactions and integrating supply, production, and logistics processes. However, there are fundamental differences between supply chain management and the cloud manufacturing model(Li,2022). Cloud manufacturing is a service-oriented model, while a supply chain is a product-centric system. This means that all mechanisms of a supply chain are designed to deliver specific products (or services), whereas in the cloud manufacturing model, added value is created solely through the sale of services. As a result, the cloud manufacturing model is not necessarily created to produce a specific product or set of products. It allows for the sale of individual services and also enables the satisfaction of a wider range of complex and customized demands through the integration and combination of services(Pang,2023).

Cloud manufacturing is a highly flexible production model that plans production immediately upon receiving a customer demand, whereas many supply chains are envisioned with push or push-pull structures. In a given supply chain, every entity plays a role as a customer, manufacturer, or supplier. However, a supplier in the cloud manufacturing model may act as a customer at another time. The centralized management system of cloud manufacturing negates the need for contracts, such as those in supply chains, or at least alters the structure of these contracts.

Information required by members of a supply chain may not always be available to them, whereas in cloud manufacturing, using advanced technologies such as the Internet of Things (IoT), all stakeholders can access the necessary information as soon as it is needed.

Although recently the issue of service composition in cloud manufacturing has attracted the attention of researchers, the available studies in this field do not fully meet the needs of the real world. Therefore, in this article, various objectives, including quality, time, and cost—which their combination significantly contributes to approaching reality and addressing real-world needs—are considered simultaneously. The problem studied in this research is the composition of services in cloud manufacturing, which, although has recently garnered a lot of attention from researchers, currently, only a limited number of studies are available in this area. The composition of cloud services is one of the fundamental approaches to achieving the added value of resources, playing a crucial role in the implementation of the cloud manufacturing process. When a customer's demand cannot be satisfied by a specific standalone service, a combination of cloud services is likely required to meet the customer's needs. Therefore, this research proposes a sustainable multi-objective model for the cloud manufacturing problem.

2. Introduction to the Problem Space

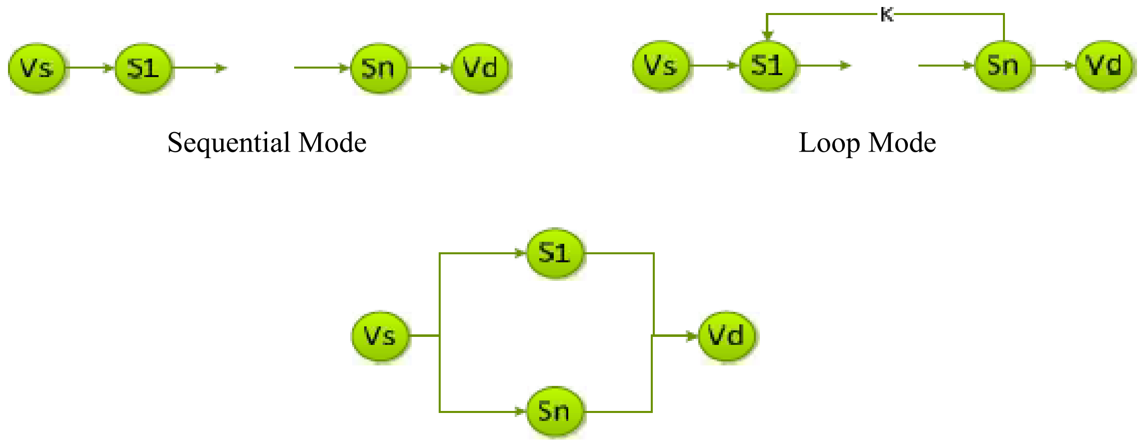
With the growth of information and communication technology and the advent of the Fourth Industrial Revolution, significant changes have occurred across various industries. In this domain, the widespread adoption of cloud computing and the Internet of Things (IoT) has led to the emergence of a new concept known as cloud manufacturing. This technology, which delivers manufacturing operations in the form of services within the cloud computing environment, has given rise to virtual manufacturers who are, in fact, consumers of cloud manufacturing services. This trend is highly beneficial for identifying emerging customer needs in the market, such as rapid response to customized production, autonomy in organizational network resources, flexibility, and efficiency in manufacturing. Consequently, some leading countries have given special attention to cloud manufacturing in their national development plans.

Although recently, the issue of service composition in the cloud manufacturing space has garnered the attention of researchers, the available studies in this field do not adequately address the real-world needs for improving the quality, cost, and time of cloud manufacturing. Therefore, the aim of this article is to provide a mathematical model for the multi-objective service composition problem in cloud manufacturing systems, which simultaneously considers various objectives, including sustainability, cost reduction, and quality enhancement. The combination of these objectives significantly helps in approaching reality and meeting real-world needs.

Multi-objective service composition problems in cloud manufacturing systems are generally NP-hard. Therefore, the proposed model is solved using precise hyper-heuristic algorithms and the NSGA-II meta-heuristic algorithm.

The problem under investigation in this research is the service composition in the cloud manufacturing space. Although this topic has recently garnered significant attention from researchers, currently, only a limited number of studies are available in this field. The composition of cloud services plays a crucial role in the implementation of the cloud manufacturing process. In the cloud manufacturing model, a customer's demand (for example, the manufacturing of a complex mechanical part) is submitted to cloud manufacturing, and cloud managers are responsible for examining, interpreting, executing, and delivering the customer's requirements while considering service quality. If a customer's demand cannot be satisfied using a single service from a cloud manufacturing perspective, a combination of cloud services is likely to be used to meet the customer's needs. Since a set of cloud services may have similar performance, multiple services could potentially be envisioned to fulfill the customer's requirements. Thus, the fundamental question in the problem of service composition is how to select the optimal service that will achieve an optimal combination of services. This issue can be modeled as a multi-objective optimization problem with a set of design constraints, quality, cost, time, etc., and is classified as an NP-complete problem.

In this problem, there are n services in a composition (series, parallel, or series-parallel) (S1,…,Sn), where each has candidate services with similar performance but differing in quality, cost, and time. Four common control structures in service composition include sequential, loop, parallel, and composite modes. In a sequential structure, services are performed in a specific order, and the next service cannot be started until the previous one is completed. In contrast, in a parallel service composition mode, multiple services can be executed simultaneously. The loop structure is essentially the same as the sequential structure but repeats for a predetermined number of times. In a loop structure, after completing the last service, the cycle restarts, repeating this process K times. A problem that results from the integration of at least two of the sequential, parallel, and loop structures is considered to have a composite structure.



Sequential Mode

Loop Mode

Parallel Mode

Figure 1. Structure of the Problem in Sequential, Parallel, and Loop Modes

As shown in Figure 1, the sequential, parallel, and loop structures are specific cases of the service composition problem, presented alongside the structure shown in Figure 2, which is a composite structure. This research focuses on the mathematical modeling of the composite structure of the problem. For the purpose of modeling the problem, the parameters and decision variables in the composite structure are defined as presented in Figure 2. It is worth mentioning that the objectives considered in the service composition problem include goals such as stability, cost minimization and economic aspects, minimization of environmental issues, maximization of social impact, and maximization of quality.

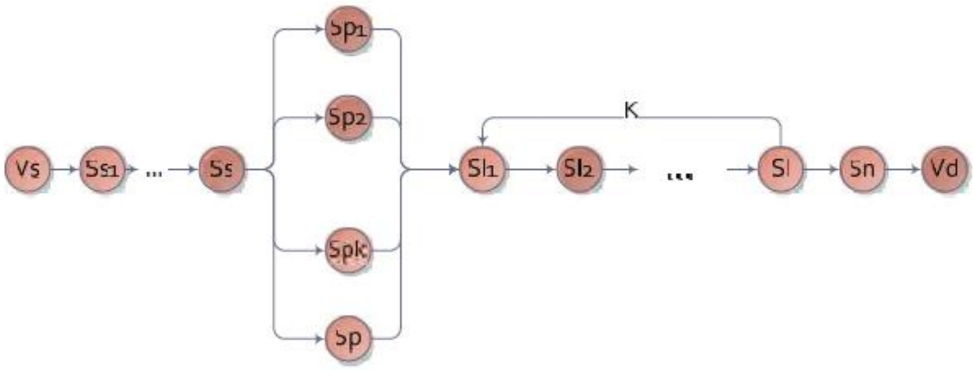


Figure 2. Composite Structure in the Service Providers Problem

3. Mathematical Model

To present a three-objective mathematical model for the described service composition problem, the symbols and notations are introduced as follows:

Indices

i', i	Activity Index
j', j	Service Provider Index
e	Environmental Performance Index
c	Economic Performance Index
so	Social Performance Index
d	Service Requester Index

Decision Variables

sd_d	Total number of requesters for the service
$tmcs_{ij}$	Time of activity i to receive service j

$pmcs_{ij}$	Cost of executing activity i from service provider j
$ttcs_{iji'j'}$	Time to transfer activity i from service j to activity i′ from service j’
$ptcs_{iji'j'}$	Cost to transfer activity i from service j to activity i′ from service j'
$EnCS_{ij}$	Level of environmental components created for activity i from service provider j as obtained from equation 3-1.
$EneCS_{eij}$	Level of environmental performance components e for activity i in service j
$Enemin_{eij}$	Minimum level of environmental performance components e for activity i in service j
$Enemax_{eij}$	Maximum level of environmental performance components e for activity i in service j
$landa_{e}$	If the environmental index e is positive, then 1; otherwise, -1
$EcCS_{ij}$	Level of economic components for activity i in service j as obtained from equation 3-2.
$EccCS_{cij}$	Level of economic performance components c for activity i in service j
$Eccmin_{cij}$	Minimum level of economic performance components c for activity i in service j
$Eccmax_{cij}$	Maximum level of economic performance components c for activity i in service j
$landa_{c}$	If the economic index c is positive, then 1; otherwise, -1
sCS_{ij}	Level of social components for activity i in service j as obtained from equation 3-3.
$ssCS_{sij}$	Level of social components s for activity i in service j
$ssmin_{sij}$	Minimum level of social components s for activity i in service j
$ssmax_{sij}$	Maximum level of social components s for activity i in service j
$landa_{s}$	If the social index s is positive, then 1; otherwise, -1
Enmin	Minimum environmental level
Ecmin	Minimum economic level
spmin	Minimum social level
$tmax_{d}$	Maximum availability time of service providers to offer services to requester d
$pmax_{d}$	Maximum cost of service providers to offer services to requester d
$qmin_{d}$	Minimum service level of service providers to offer services to requester d
$pd_{iji'j'}$	Distance of service location for activity i by service j from service location for activity i′ by service j'
$plc_{iji'j'}$	Logistic cost of activity i by service j from service location of activity i′ by service j'
Np_{i}	Number of service requests for activity i
M	Large positive number

1
$$EnCS_{ij} = \frac{(\sum_e landa_e * EneCS_{eij} - Enemin_{eij})}{Enemax_{eij} - Enemin_{eij}} \qquad \forall\ i \text{ و } j$$

2
$$eccS_{ij} = \frac{(\sum_c landa_c * EccCS_{cij} - Eccmin_{cij})}{Eccmax_{cij} - Eccmin_{cij}} \qquad \forall\ i \text{ و } j$$

3
$$scS_{ij} = \frac{(\sum_s landa_s * ssCS_{sij} - ssmin_{sij})}{ssmax_{sij} - ssmin_{sij}} \qquad \forall\ i \text{ و } j$$

Mathematical Model

4
$$\text{Min Som} = \left\{\frac{1}{EN} ، \frac{1}{Ec} ، \frac{1}{S}\right\}$$

5
$$\text{Max Qos} = \text{Min}\left\{t_d ، p_d ، \frac{1}{q_d}\right\}$$

6 $$\text{Min Tlc} = \sum_{j'=1}\sum_{i=0}\sum_{j=0}\sum_{i'=1} plc_{iji'j'} * z_{iji+1j'} * tmcs_{ij} + \sum_{j'}\sum_{\substack{i\\ i>1}}\sum_{\substack{i'\\ i'=i+1}}\sum_{j} plc_{iji'j'} * z_{iji+1j'} * pd_{iji'j'} + \sum_{j'}\sum_{i}\sum_{i'}\sum_{j} plc_{iji'j'} * z_{iji+1j'}$$

7 $$en = \sum_{d}\sum_{i}\sum_{j} EnCS_{ij} * x_{ij}$$

8 $$ec = \sum_{d}\sum_{i}\sum_{j} EcCS_{ij} * x_{ij}$$

9 $$s = \sum_{d}\sum_{i}\sum_{j} scS_{ij} * x_{ij}$$

10 $$t_d = \sum_{i}\sum_{j} tmcs_{ij} * x_{ij} + \sum_{i}\sum_{j}\sum_{j'} z_{iji+1j'} * ttcs_{iji+1j'} \qquad \forall d$$

11 $$p_d = \sum_{i}\sum_{j} pmcs_{ij} * x_{ij} + \sum_{i}\sum_{j}\sum_{j'} z_{iji+1j'} * ptcs_{iji+1j'} \qquad \forall d$$

12 $$q_d = \prod_{i}\sum_{j} qcs_{ij} * x_{ij} \qquad \forall d$$

13 $$en \geq enmin$$

14 $$EcCS_{ij} * x_{ij} \geq ecmin * x_{ij} \qquad \forall i \text{ و } j$$

15 $$sCS_{ij} * x_{ij} \geq spmin * x_{ij} \qquad \forall i \text{ و } j$$

16 $$t_d \leq tmax_d \qquad \forall d$$

17 $$p_d \leq pmax_d \qquad \forall d$$

18 $$q_d \geq qmin_d \qquad \forall d$$

19 $$\sum_{j} x_{ij} = 1 \qquad \forall i$$

20 $$z_{iji'j'} \leq \frac{x_{ij} + x_{i'j'}}{2} \qquad \forall\ i \text{و} j \text{و} i' \text{و} j'$$

21 $$z_{iji'j'} \geq x_{ij} + x_{i'j'} - 1 \qquad \forall\ i \text{و} j \text{و} i' \text{و} j'$$

22 $$t_d = positive \text{ ، } p_d = positive \text{ ، } q_d = positive$$

Equation 4 represents the first objective function related to optimizing sustainability components (environmental, economic, and social dimensions). Equation 5 is the second objective function aimed at optimizing service quality (time, price, and quality), and Equation 6 represents the third objective function, which is focused on minimizing logistical costs. Constraint 7 pertains to the environmental level of activities using cloud service providers. Constraint 8 is related to the economic level of activities using cloud service providers. Constraint 9 addresses the social level of activities using cloud service providers. Constraint 10 pertains to the execution time of activities using cloud service providers. Constraint 11 concerns the total cost of the system for activities using cloud service providers. Constraint 12 relates to the quality level of activities using cloud service providers. Constraint 13 ensures that the environmental level of cloud production exceeds the minimum level considered. Constraint 14 ensures that the costs of cloud production do not exceed the initial budget. Constraint 15 states that social components must exceed the minimum level determined. Constraint 16 ensures that the duration range of cloud production is less than the maximum level considered. Constraint 17 relates to the cost range of cloud production, which must be less than the maximum level set. Constraint 18 pertains to the quality range of cloud production. Constraint 19 states that each activity can only be connected to one service provider. Constraints 20 and 21 are used for linearization purposes. Constraint 22 specifies the status of the decision variables in the problem.

4. Solution Approach

Meta-Goal Programming

Multi-objective mathematical model problem-solving strategies have changed a lot in the last decade. One way to solve multi-objective programming problems is to use the meta-goal programming method. Meta-goal programming is one of the most widely used solution methods in multi-objective mathematical modeling problems (Charnes and Cooper, 1961), first proposed by Charnes and Cooper. The general form of a multi-goal programming model is as follows (Diaz et al., 2003):

$$f_i(x) + n_i + p_i = t_i \quad i = 1,\dots,s$$

$$g_i(x) \le b_i \quad j = 1,\dots,m \tag{23}$$

$$x \in R^n$$

There are s goals and m system constraints in this model. The functions $f_i(x)$ are concave and the functions $g_i(x)$ are convex in most cases. The goals include an unwanted component called deviation variables, which are n_i and p_i, used for negative and positive deviations from the desired goal level, respectively, and are minimized by one of the priority, weight, or mini-max methods (Oriya et al., 2002).

So, this study uses the meta-goal programming strategy according to the level of considered goals as well as the programming problem. A list of all the signs and indicators used in the study along with a brief description of each of them can be seen in Table 1.

Table 1. A list of signs used in the study

Model Variables	
n_i:	Undesirable negative deviation variables concerning goal i
p_i:	Undesirable positive deviation variables concerning goal i
$f_i(x)$:	The concave functions of the goals
$g_i(x)$:	The convex functions of the constraints
s:	The number of goals
:m	The number of system constraints
ω_i:	Preferred weights
: r_i	The goal corresponding to each ideal
D:	The maximum Percentage of Weighted Deviation
y_i:	Binary Variables
$S_k^{(1)}$:	A set of type I meta-goals
$S_l^{(2)}$:	A set of type II meta-goals
$S_r^{(3)}$:	A set of type III meta-goals
$Q_k^{(1)}$:	The limit of deviation from the type I meta-goal
$Q_l^{(2)}$:	The limit of deviation from the type II meta-goal
$Q_r^{(3)}$:	The limit of deviation from the type III meta-goal

If the results are not satisfactory to the decision-maker after solving the model by one of these single-objective solution methods, the three-objective mathematical model can be converted into three single-objective models and satisfactory levels can be obtained for each of the objectives based on the model solution. When the meta-goal model is solved by the weighted method using the single-objective mathematical model, the function of achieving the least undesirable deviations is defined as Equation 24 (Rahnam and Romero, 1978; Nili et al., 1976):

$$h(n) = \sum_{i=1}^{s} \omega_i \frac{n_i}{r_i} \tag{24}$$

Where ω_i represents the preferred weights given by the decision-maker to each of the undesirable deviation variables, which is normalized by dividing by the corresponding goal of each ideal. Other methods such as normalized Euclidean distance should be used to normalize undesirable deviation variables if the corresponding goal of each ideal is zero or negative (Tamizou et al., 1998). The optimal solutions in the weighting method are obtained by minimizing the percentage sum of the weighted values of the undesirable deviation variables.

The mini-max method can also be used to minimize undesirable deviations from the acceptable level of each goal. Then, the function of achieving the minimum undesirable deviations will be as Equation 25 (Oriya et al., 2002):

$$\max_{i=1,\dots,s} \left\{ \omega_i \frac{n_i}{r_i} \right\} \tag{25}$$

In this type of meta-goal programming, the maximum undesirable weighted deviation from the desired goal is minimized until the final solution is achieved. It is assumed that the decision-maker is less flexible about the final values of the functions for achieving the minimum undesirable deviations and wants to consider goal levels for these values so that these values do not exceed a certain goal level or change in a certain range. This assumption gives rise to other goals that can usually be part of the model's core goals. Accordingly, they are called meta-goal models. So, three types of meta-goals are conceivable, each of which is described below.

Type I meta-goal

This meta-goal adds another constraint to the main program, according to which the sum of the weighted values of the normalized undesirable deviation variables should not exceed a critical limit of Q_1. This meta-goal is developed as follows:

$$\sum_{i=i}^{s} \omega_i \frac{n_i}{r_i} \le Q_1 \tag{26}$$

Table 1 describes each of the formula components.

Type II meta-goal

According to this meta-goal, the maximum weighted deviations of the normalized undesirable variables should not exceed a critical limit of Q_2 (Oriya et al., 2002):

$$\max_{i=1,..,s} \left\{ \omega_i \frac{n_i}{r_i} \right\} \le Q_2 \Leftrightarrow \begin{cases} \omega_i \frac{n_i}{r_i} - D \le 0 \ , \ i = 1,\dots,s \\ D \le Q_2 \end{cases} \tag{27}$$

Where D indicates the maximum percentage of weighted deviations.

Type III meta-goal

In addition to the above two meta-goals, another type of meta-goal has been defined which does not directly affect the calculation of the functions of achieving the minimum undesirable deviations. According to this meta-goal, the sum of unfulfilled goals should not exceed a critical limit of Q_3 (Oriya et al., 2002):

(28)

$$\begin{cases} n_i - M_i y_i \leq 0 \\ \dfrac{\sum_{i=1}^{s} y_i}{s} \leq Q_3 \end{cases}$$

Where y_i is a binary variable and M_i is an arbitrarily large number that is usually considered equal to the corresponding goal of each ideal. So, the final value of $\sum_{i=1}^{s} y_i$ in the optimal solution indicates the number of goals that are not fully met. Most formulations require specific subsets that contain all the goals so that the decision-maker can create each of the defined meta-goals. For example, if type I meta-goal is defined by set $S_k^{(1)} \subset \{1,2,\dots,s\}$, its goal is rewritten as Equation 29:

(29)

$$\sum_{i \in s_k^{(1)}} \omega_i \frac{n_i}{r_i} \leq Q_k^{(1)}$$

Similarly, if set $S_l^{(2)}$ is defined for the type II meta-goal, its goal will be as Equation 30:

(30)

$$\omega_i \frac{n_i}{r_i} - D \leq 0,\ i \in s_l^{(2)},\ D \leq Q_l^{(2)}$$

Finally, by defining set $S_r^{(3)}$ as the previous sets for the type III meta-goal, its goal will be as Equation 31:

(31)

$$\begin{cases} n_i - M_i y_i\ ,\ i \in s_r^{(3)} \\ \dfrac{\sum_{i \in s_r^{(3)}} y_i}{card\left(s_r^{(3)}\right)} \leq Q_r^{(3)} \\ y_i \in \{0,1\}\ ,\ i \in s_r^{(3)} \end{cases}$$

In some cases, after solving the model using the defined constraints, an empty feasible set will be created, which is likely to be inefficient, even if there is a feasible solution. One possible way to solve the problem in this situation is to present a meta-goal model as follows, in which it is assumed that type I meta-goal r_1, type I meta-goal r_2, and type I meta-goal r_3 are available to the decision-maker. The meta-goal model obtained from this method ($[GP]^2$) is defined as Equation 32 (it should be noted that card (x) is the counter of the number of subsets in the set x):

(32)

$$[GP]^2 \quad MIN \left\{\beta_1^{(1)},\ldots,\beta_{r_1}^{(1)},\ldots,\beta_1^{(2)},\ldots,\beta_{r_2}^{(2)},\ldots,\beta_1^{(3)},\ldots,\beta_{r_3}^{(3)}\right\}$$

$$f_i(x)+n_i+p_i=t_i \ , \ i=1,\ldots,s$$

$$g_i(x)\le b_i \ , \ j=1,\ldots,m$$

$$\sum_{i\in s_k^{(1)}} \omega_i \frac{n_i}{r_i}+\alpha_k^{(1)}-\beta_k^{(1)}=Q_k^{(1)} \ , \quad k=1,\ldots,r1$$

$$\omega_i \frac{n_i}{r_i}-D\le 0 \ , i\in S_l^{(2)}, l=1,\ldots,r2$$

$$D_l+\alpha_l^{(2)}-\beta_l^{(2)}=Q_l^{(2)} \ , \ l=1,2,\ldots,r2$$

$$n_i-M_i y_i\le 0 \ , \ i\in S_r^{(3)} \ , \quad r=1,2,\ldots,r3$$

$$\frac{\sum_{i\in s_r^{(3)}} y_i}{card\left(s_r^{(3)}\right)}+\alpha_r^{(3)}-\beta_r^{(3)}=Q_r^{(3)} \ , \ r=1,2,\ldots,r3$$

$$y_i\in\{0,1\} \ , \ i\in S_r^{(3)} \ , \ r=1,2,\ldots,r3$$

$$n_i, p_i\ge 0 \ , \quad i=1,\ldots,s$$

$$X\in R^n$$

$$\alpha_k^{(1)},\beta_k^{(1)},\alpha_l^{(2)},\beta_l^{(2)},\alpha_r^{(3)},\beta_r^{(3)}\ge 0$$

Due to the importance of the perishable product supply chain (PPSC) problem, in this study, type I to III meta-goals concern this production input, and the permissible deviation from these two meta-goals is also zero. However, the effect of changes in the permissible deviation from the type IV meta-goal, which is related to other constraints, can be examined at three levels of 0, 0.1, and 1 on changes in the optimal cloud manufacturing model. The results of estimating the meta-goal model at different levels of the type I meta-goal in the third priority (type IV meta-goal) can be seen in Table 2 . $Q_2^{(1)}=0$, $Q_2^{(1)}=0.1$, $Q_2^{(1)}=1$).

Different levels $Q_2^{(1)}$ refer to the permissible limits of deviation from the main goal. For example, ($Q_2^{(1)}=0.1$) means that the main goal can be exceeded by as much as 10%, which indicates the deviation from the sum of the values of the normalized undesirable deviation variables, assuming the other meta-goals are constant for the type IV meta-goal ($Q_2^{(1)}$). As can be seen, both have economic and environmental models. Although the minimum environmental level is at both $Q_2^{(1)}=1$ and $Q_2^{(1)}=0.5$, the ideal goal is 436 units at $Q_2^{(1)}=1$. So, $Q_2^{(1)}=0$ has the highest level of implementation of activities in the most optimal economic and environmental conditions, but the highest level of time of implementation of activities and costs is allocated to $Q_2^{(1)}=0.1$ and $Q_2^{(1)}=1$. As the permissible level of deviation from the above meta-goal increases, the total stability increases. Here, too, this level increases for the execution of activities by the servers. Although stability is constant at $Q_2^{(1)}=0.1$ and $Q_2^{(1)}=0$, it decreases at $Q_2^{(1)}=1$.

Table 2. The results of estimating the meta-goal programming model at different levels $Q_2^{(1)}$

Product	***Variable***	$Q_2^{(1)}=1$	$Q_2^{(1)}=0.1$	$Q_2^{(1)}=0$
Product 2	*x_1 (environmental level)*	-	-	-
Product 2	*X_2 (Economic level)*	*5226*	*5289*	*5289*
Product 3	*X_3 (environmental level)*	8566	7796	5321
Product 3	*X_4 (Economic level)*	*175*	-	-
Product 4	*X_5 (environmental level)*	-	-	-
Product 4	*X_6 (Economic level)*	*1397*	*2237*	*6297*
Product 5	*X_7 (environmental level)*	*6710*	*5810*	*4209*
Product 5	*X_8 (Economic level)*	-	-	-
Total	**X**	*22074*	***21134***	***21117***

In addition, real test problems are generally big in their size and bigger the size of the problem, bigger the complexity (Andisheh Abdi et al., 2020; Anita Abdi et al., 2019). In this regard and due to the Np-*hardness* of the mentioned problem, it would be impossible to solve the model using exact methodologies. Utilizing exact methods could be time-consuming and one cannot guarantee the best solution in a given time especially in more complex test problems (Amiri et al., 2020). Therefore, here in this paper, four different metaheuristic approaches have taken into account. non-dominated ranking genetic algorithm (NSGA-II) (Deb et al., 2002) are exerted to find best Pareto solutions for the proposed multi-objective model. The utilized encoding and decoding strategy for the concerned metaheuristics is developed in the following subsection.

4.1. Encoding and decoding strategy

There are a number of developed strategies in order to encode the solutions in various formulations. These strategies are ranging from Michalewicz matrix Michalewicz et al., 1991), Prufer numbers (Prüfer, 1918), and priority-based method (Gen et al., 2006). In this study, the application of and priority-based method has taken into account and small-sized example is utilized to illustrate the considered chromosome. This method is also leading to satisfying all the considered constraints. The schematic plan to encode and decode each chromosome is depicted in Fig. 4.

Section 1				**Section 2**										
j		*k*		*l*		*j+k+l+c+p+t+r+v*								
0.90	0.52	0.68	0.95	0.19	0.07	0.63	0.59	0.35	0.58	0.74	0.96	0.98	0.78	0.53
1	2	2	1	1	2	7	6	8	5	1	2	4	9	3

Section 3							**Section 4**												
k			*l+c+p+t*				*j+k+l+c+p+t+r*											*v*	
0.31	0.12	0.75	0.74	0.75	0.39	0.57	0.50	0.80	0.36	0.71	0.01	0.29	0.54	0.96	0.41	0.57	0.26	0.58	0.08
3	1	2	2	1	4	3	8	2	4	10	7	1	9	3	6	11	5	1	2

Section 5			
l		*c*	
0.80	0.55	0.13	0.09
1	2	1	2

Fig. 1. Encoding and decoding plan for sections 1-5.

In Fig. 1. encoding and decoding plan within each chromosome is depicted. In each section, the first row defines the flow between considered centers. Second and third rows are random numbers in the interval [0,1] and decoding plan priorities respectively. After defining the encoding and decoding plan, in the next subsections, the proposed metaheuristics are defined.

4.2. Proposed NSGA-II

In this subsection, the application of NSGA-II is taken into account. NAGA-II proposed by Deb et al. (2002) respectively. Both algorithms are identical in most of their procedure. They initiate by producing the population. Then this population is evaluated. In the next steps, two operators including sorting and crowing are applied to make the selected population even better. To avoid trapping in the local optimum answer in these algorithms, a uniform mutation crossover is applied for both one for the aim of algorithm diversification. Fig. 5 simultaneously depicted the pseudocode of algorithms.

```
1. Initialize Population;
2. Generate random population;
3. Evaluate Objectives Values;
4.   For each Parent and Child in Population do
5.      Assign Rank (level) based on Pareto;
6.      Generate sets of nondominated solutions;
7.      Determine Crowding distance
8.      Loop (inside) by adding solutions to next generation
9.   End
10. Determine population front;
11. For each determined front
12.    Perform Binary tournament selection Rolette wheel selection (NSGA-II);
13. Generate new population with mutation and crossover;
14 End
```

Fig. 2. Pseudocode of NSGA-II.

5. Parameter tuning and computational results

In this section, the efficiency of the proposed metaheuristics is considered. In order to obtain the best results from the employed metaheuristic approaches, after generation random input data, the application of the Taguchi method is taken into account. This method is introduced by Taguchi (1986) which offers various test experiments along with parameter tuning for each algorithm in order to get the best output results in the proposed problem.

5.1. Parameter settings

In order to illustrate the applicability and relevance of the proposed model, the problem must be solved under various conditions and settings. Hence, here, 12 test problems with various sizes and parameter settings are considered. The test problems along with its corelateded parameters are presented in Table 3. According to this table, as the number of test problems increase, so the complexity of the model as larger problem sizes offers more NP-*hardness* settings. Using these test problems, the efficiency and applicability of the utilized metaheuristics could be easily investigated in various size problems.

The dimensions of the problem are depicted in the table

Table 3 Test problems settings.

Test problem	*j*	*k*	*l*	*c*	*p*	*t*	*r*	*v*
1	8	2	2	2	1	1	1	1

2	16	4	4	4	2	2	2	2
Case study	25	6	7	6	5	4	6	2
4	30	8	8	6	6	6	6	4
Case study	34	12	12	10	10	6	15	6
6	50	20	20	18	15	10	15	10
7	70	30	30	22	25	15	20	12
8	90	40	40	35	35	25	30	14
9	110	50	50	45	45	30	40	16
10	130	60	60	50	55	35	50	18
11	150	70	70	55	65	35	55	20
12	200	85	85	65	80	45	70	25

5.2. Tuning the parameters of the algorithms and achieved results

In this subsection, the application of Taguchi approach is employed to set the algorithms' parameters in order to get the optimum results. Using Taguchi approach would decrease the number of total experiments by eliminating unnecessary ones. In this regard, it uses cluster of factors which are based on orthogonal arrays. These factors are categorized into two essential groups namely control and noise factor. Hence, to evaluate this response variation, a method is needed to verify signal to noise ratio. It should be noted that, the Taguchi setting is related to the type of response. In this study, the response type of "the smaller is better" is exerted to designate the best settings in each considered levels of proposed metaheuristics. The considered levels for proposed metaheuristics are represented in Table 4.

Table 4 Algorithm factors alongside with their levels.

Algorithms	Factor	Factors levels		
		Level 1	Level 2	Level 3
NSGA-II	A: Pc	0.75	0.85	0.95
	B: Pm	0.06	0.11	0.16
	C: N-pop	60	120	180
	D: Max-iteration	2x	3x	4x

The initial step to implement the Taguchi experimental design is to identify levels for each factor of the algorithm (see Table 4). The next step would be using Minitab software to analyses the experiment with its Taguchi experiment toolbar. In this respect, the L_9 design was used for NSGA-II algorithm. As aforementioned, to identify the best levels for each algorithm, the evaluation of signal to noise ratio is required. Equation (33) represents the selected signal to noise ratio and its evaluation method. This quantity identifies the variation in response relative to the target value and under various noise conditions.

$$Signal/Noise = -10\ log\left(\sum (Y^2)\ /n\right) \tag{33}$$

Where Y and n are the response time and number of orthogonal arrays respectively. To verify the best levels for each algorithm, the signal to noise plot for each algorithm is depicted in Figs. 5 .

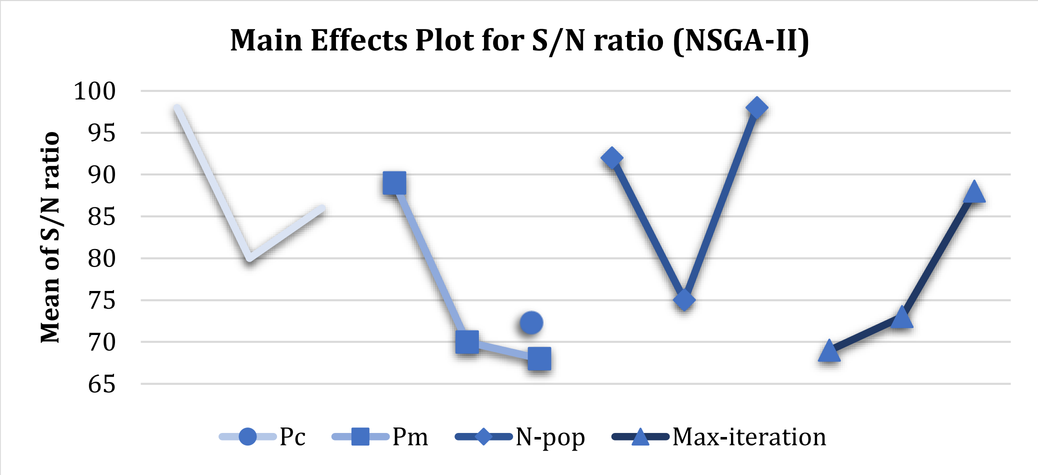


Fig. 1. S/N ratio for NSGA-II.

Considering Fig. 5, the best levels for each algorithm factors are described in Table 5.

Table 5 Tuned parameters of the algorithm.

Algorithm	Parameters
NSGA-II	Pc=0.75; Pm=0.06; N-pop=180; Max-iteration=4x

Table 1 Results from exact method.

Test problem	Objective function	Weight of first objective function (w1)								
		0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
1	Total cost	1.43E+10	1.43E+10	1.43E+10	1.43E+10	1.43E+10	1.43E+10	1.43E+10	1.43E+10	1.43E+10
	environmental	2.90E+04	2.91E+04	2.92E+04	2.96E+04	3.03E+04	3.07E+04	3.07E+04	3.08E+04	3.08E+04
2	Total cost	1.97E+10	1.97E+10	1.97E+10	1.96E+10	1.95E+10	1.87E+10	1.86E+10	1.83E+10	1.78E+10
	environmental	3.49E+04	3.49E+04	3.49E+04	3.50E+04	3.52E+04	3.87E+04	3.90E+04	3.90E+04	3.90E+04
3	Total cost	8.00E+10	7.99E+10	7.98E+10	7.97E+10	7.96E+10	7.95E+10	7.95E+10	7.66E+10	7.31E+10
	environmental	5.81E+07	5.89E+07	5.91E+07	5.94E+07	5.98E+07	6.02E+07	6.02E+07	6.04E+07	6.07E+07

In addition, the Pareto solutions of the considered metaheuristics for test problems 3, 5, and 10 are depicted in Figs. 6-8.

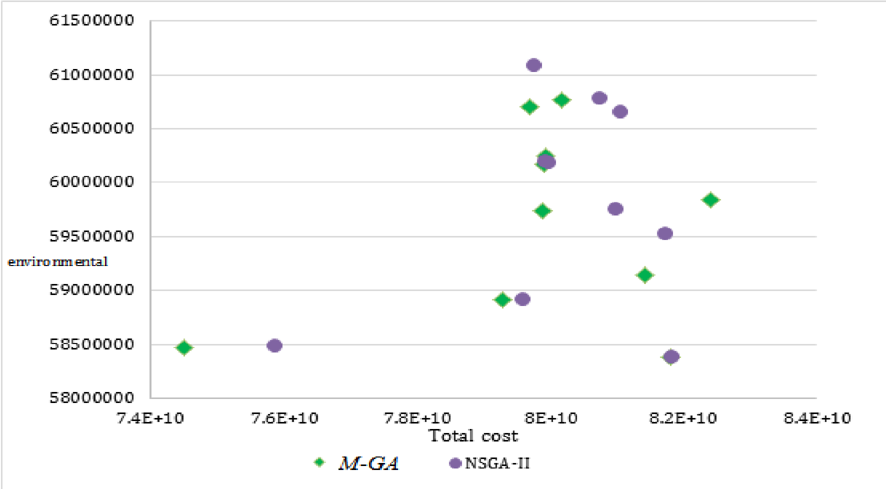


Fig. 6. Pareto solution of test problem 3.

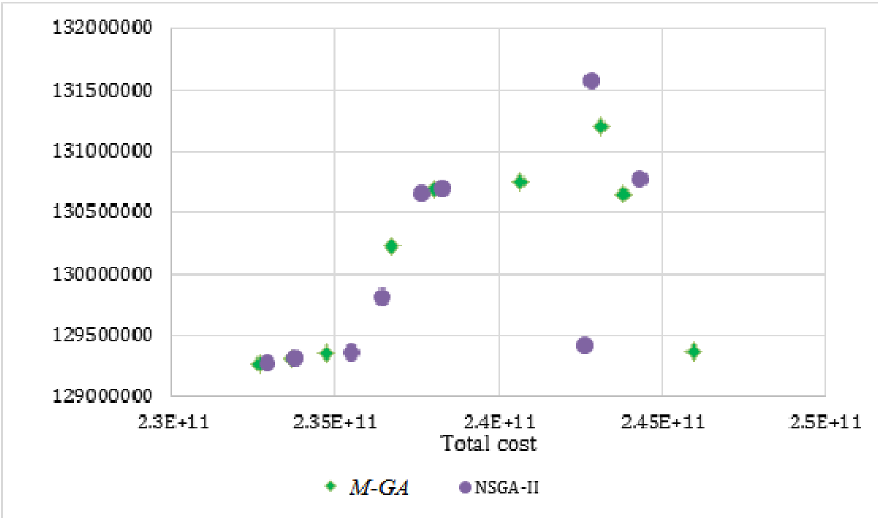


Fig. 7. Pareto solution of test problem 5.

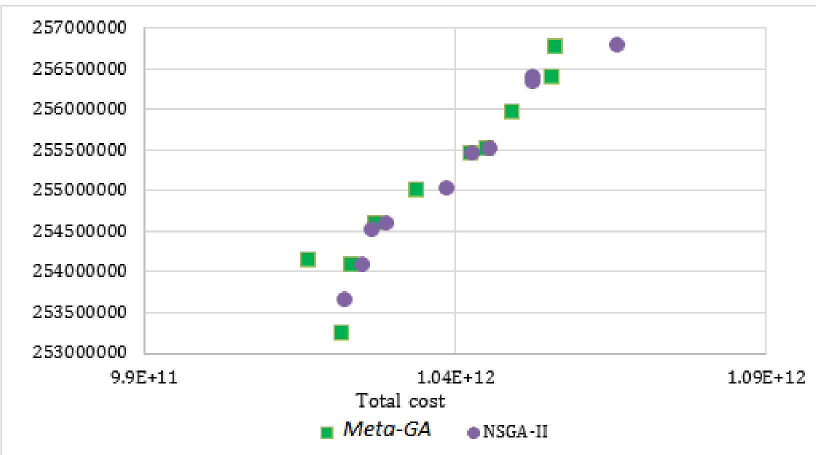


Fig. 8. Pareto solution of test problem 10.

6. Conclusion and Future Suggestions:

In recent years, advancements in microelectronics technology have led to the development of cheap and fast processors in cloud computing. Currently, multi-processor networks in cloud computing have become highly prevalent. The special applications of such technologies become more prominent when these processors play a key role in them. The reliability of multi-processors in cloud computing has enabled the development and creation of real-time systems, which involve the design of processor placement and single-chip architectures with a limited number of processors up to large-scale signal processing systems. Consequently, the problem of scheduling synchronous tasks in single-processor cloud systems has been extensively studied and evaluated by researchers, and the synchronization available in multi-processor cloud systems has been introduced in recent years.

A real-time system aims to have an actual impact within a specific timeframe. Typically, a real-time system includes a guiding system (computer) and a controlled system (environment). The guiding system interacts and coordinates with the surrounding environment based on information. For example, in a real-time system analyzing a laptop or a processing unit, sensors create readings at regular intervals, and the computer must react by sending pulses to cloud computing actuators. There might be irregular or abnormal strategies that also need to generate a response. In all cases, a time frame is created within which a response must be generated. The capability of cloud computing to fulfill these needs depends on its capacity to process the necessary computations within the required timeframe.

If multiple events occur simultaneously, to ensure each process is executed within the required time frames, the system engages in agent scheduling. In some cases, the system might not be able to perform all possible unforeseen operations. In this situation, it is stated that the system does not have sufficient cloud computing resources. A system with unlimited processing capabilities and infinite speed can execute any time frame. Failure to meet time frames for a response can lead to various consequences; in the first scenario, the event might not occur, or the effects could be negligible or adjustable, or the results might be unacceptable. Each task occurring in cloud computing processing has multiple time-related attributes. These attributes must be considered when scheduling tasks on a cloud computing processor.

In a cloud manufacturing system, there are often many required services that need to be executed in a short time. The urgency of these requests may vary depending on their nature (meaning that a security-related request might be much more critical than a request for accessing information) or based on the availability period or execution time. Therefore, the allocation of processors to the system requires the scheduling of all requests that need to be analyzed considering their execution time. These activities are usually implemented through a cloud manufacturing planning program. In other words, a scheduling strategy is determined to define how a system’s processor is allocated to the program.

Scheduling strategies can be mathematically evaluated and analyzed; therefore, the precision of formal characteristics and the steps for improving scheduling can be completed through temporal analysis of program features. As a result, in this research, considering the cloud manufacturing system based on cost, time, and quality components, and taking into account the dimensions of economic, social, and environmental sustainability, an evaluation and analysis were conducted, and a mathematical model was developed in this area. In the analysis conducted on small-scale problems, optimal solutions were evaluated using a metaheuristic algorithm, and finally, large-scale problems were studied and analyzed using the NSGA-II metaheuristic algorithm. In the conducted analysis, first, the parameters of the algorithm were set based on the Taguchi method, and then the dimensions of the problem were evaluated and analyzed based on exact and metaheuristic solutions. It was shown that the development of cloud manufacturing systems patterns, considering analytical models, is highly practical and executable.

Based on the research findings and to further extend this scientific research, the following are Suggestions:

- Considering the risk of some service providers going offline during operations in the mathematical model.
- Evaluating irregular queue theory in the processing of activities.
- Using the concept of fuzzy uncertainty in the analysis of the mathematical model and comparing the obtained results with robust optimization in examining uncertainties.
- Evaluating and developing fuzzy Bayesian networks and grey algorithms in analyzing the communication network.

Resources

[1] Hamidi Moghaddam, S., Akbaripour, H., & Houshmand, M. (2021). Integrated forward and reverse logistics in cloud manufacturing: an agent-based multi-layer architecture and optimization via genetic algorithm. *Production Engineering*, *15*, 801-819.

[2] Hu, Y., Wu, F., Yang, Y., & Liu, Y. (2022). Tackling temporal-dynamic service composition in cloud manufacturing systems: A tensor factorization-based two-stage approach. Journal of Manufacturing Systems, 63, 593-608

[3] Li, W., Fan, Q., Dang, F., Jiang, Y., Wang, H., Li, S., & Zhang, X. (2022). Multi-Objective Optimization of a Task-Scheduling Algorithm for a Secure Cloud. *Information*, *13*(2), 92.

[4] Pang, S., Guo, S., Wang, X. V., Wang, L., & Wang, L. (2023). Dual-dimensional manufacturing service collaboration optimization toward industrial internet platforms. Engineering, 22, 34-48.

[5] Rashidifar, R. (2023). *Optimization of Multi-Objective Resource Scheduling in Cloud Manufacturing Environment via Integrating Reinforcement Learning and Deep Neural Network* (Doctoral dissertation, The University of Texas at San Antonio).

[6] Salmasnia, A., & Kiapasha, Z. (2023). Integration of sub-task scheduling and logistics in cloud manufacturing systems under setup time and different task arrival times. *International Journal of Computer Integrated Manufacturing*, *36*(7), 985-1008.

[7] Song, C., Zheng, H., Han, G., Zeng, P., & Liu, L. (2022). Cloud edge collaborative service composition optimization for intelligent manufacturing. IEEE Transactions on Industrial Informatics, 19(5), 6849-6858.

[8] Song, H., Lu, X., Zhang, X., Tang, X., & Zhang, Q. (2023). Collaborative optimization for energy saving and service composition in multi-granularity heavy-duty equipment cloud manufacturing environment. *Journal of Industrial and Management Optimization*, *19*(4), 2742-2771.

[9] Wang, T., Zhang, P., Liu, J., & Gao, L. (2022). Multi-user-oriented manufacturing service scheduling with an improved NSGA-II approach in the cloud manufacturing system. *International journal of production research*, *60*(8), 2425-2442.

[10] Xie, N., Tan, W., Zheng, X., Zhao, L., Huang, L., & Sun, Y. (2021). An efficient two-phase approach for reliable collaboration-aware service composition in cloud manufacturing. Journal of Industrial Information Integration, 23, 100211..