sustainability

MDPI

Article

Optimization of the Product–Service System Configuration Based on a Multilayer Network

Zaifang Zhang [1,*], Darao Xu [1], Egon Ostrosi [2] and Hui Cheng [3]

1 School of Mechatronic Engineering and Automation, Shanghai University, Shanghai 200444, China; xudarao@163.com
2 Université de Bourgogne Franche-Comté, UTBM, Pôle Industrie 4.0, Pôle ERgonomie et COnception des Systèmes ERCOS/ELLIADD EA4661, 25000 Belfort, France; egon.ostrosi@utbm.fr
3 Shanghai Aerospace Equipments Manufacturer Co., Ltd., Shanghai 200245, China; cheng_and_hui@163.com
* Correspondence: zaifangzhang@shu.edu.cn

Received: 18 November 2019; Accepted: 10 January 2020; Published: 20 January 2020

Abstract: Product–service systems (PSS) accelerate the transition of value creation patterns for manufacturing industries, from product design and production to the delivery of overall solution integrating products and services. Existing PSS configuration solutions provide customers with preferable product modules and service modules characterized by the module granularity. Every service module is essentially a whole service flow. However, the performance of the PSS configuration solution is greatly influenced by service details. In summary, this paper studied the configuration optimization of product-oriented PSS using a fine-grained perspective. A multilayer network composed of (i) a product layer, (ii) a service layer, and (iii) a resource layer was constructed to represent the elements (product parts, service activities, resources) and relationships in PSS. Service activities selection and resource allocation were considered jointly to construct the mathematical model of PSS configuration optimization, thus enabling the calculation of optimizing objectives (time, cost, and reliability) under constraints closer to the actual implementation. The importance degree of service activity was considered to improve the performance of service activities with higher importance. Corresponding algorithms were improved and applied for obtaining the optimal solutions. The case study in the automotive industry shows the various advantages of the proposed method.

Keywords: product service system; configuration optimization; multilayer network; service activities selection; resource allocation

1. Introduction

Manufacturing industries are experiencing a transition of value creation patterns from the design and production of products to the design and the delivery of services based on products, to accommodate increasingly global competition and customer-centered business settings [1]. As an overall solution that effectively integrates products and services [2], product–service systems (PSS) [3–5] play a vital role in accelerating this transition for manufacturers. By prolonging the service life of products and commoditizing the related services, PSS can change the revenue models and pricing capabilities for manufacturers [6,7]. This exemplifies a new source of competitive advantage and differentiation. Furthermore, compared to traditional business models, PSS has the potential to reduce environmental impact, thus receiving much attention as one strategic alternative for the sustainable development of firms [8–10].

PSS can be divided into three types: (a) product-oriented PSS, (b) use-oriented PSS, and (c) result-oriented PSS [5]. In product-oriented PSS, customers buy the product as usual; a service is also sold to customers additionally, which then will be operated on products or provided to customers [11].

This means that the product is the principal purpose, so the service is rendered to help customers achieve a better user experience. This paper focuses extensively on product-oriented PSS.

Services help to attain higher margins, but performing the transition or maintaining the success of service strategy is still a challenge for companies [12]. With the increase in service offers, higher costs or failures in achieving expected returns sometimes occur [13,14]. To overcome this glitch in the profit generation and commercial success, companies should target the role of PSS design [15,16]. Many studies have focused on PSS design to provide support, such as design frames [17–19], design methods [20], modularization approaches [21,22], configuration and evaluation methods [23–25].

Product–service integration solutions provided to customers include product modules and service modules [26]. There are usually many possible PSS solutions. A PSS configuration is the construction of technical systems by selecting and assembling preferable modules from a predefined product and service library according to customers' needs under certain constraints [27]. Thus, PSS configuration is an imperative part of PSS design. Studies on PSS configuration have mainly focused on the PSS configuration framework [24,28], customer perception in PSS configuration [29], the extraction of PSS configuration rules [30–33], and PSS configuration optimization methods [23,34]. Most studies on PSS configuration, especially research on PSS configuration optimization, commonly deal with some predefined level of module granularity. Overall, the product module granularity is introduced to achieve flexible manufacturing systems for product functions realization. With regard to service module granularity, every service module is essentially a whole service flow, depending on the specific module's granularity level. However, it presently lacks the PSS configuration on a finer granularity level, which can inspire the final configuration's optimization.

Considering the balance between differentiated customization and mass production, inappropriate product module granularity will increase managerial difficulty and manufacturing cost [35]. Services are generally under-designed in comparison to physical products [36]. Also, service modules are not restricted to physical artefacts. Thus, service modules should be designed considering a finer granularity level, so they can integrate tangible and non-tangible artefacts. Recent studies on service modules in PSS have mainly focused on the identification and evaluation of customer requirements [37,38], modularization and related quantification [39,40], and the evaluation of internal and external performance (such as cost, customer satisfaction, service cycle time, availability, and reliability) [11,41]. Modularization can normalize the PSS configuration and improve its suitability. However, a service module in PSS is essentially a composition of service activities. It is coarse-grained to use a whole service flow as a service module unit in PSS configuration. Therefore, configuring a PSS in finer granularity by using service activity as the minimum configuration unit is proposed and developed in this paper.

There are various methods to optimize the composition of service activities, which can be embodied in the following three approaches [42]:

(1) Service composition solver based on artificial intelligence planning, which can automatically generate service activities composition according to customer demands;

(2) The method based on semantics, studying the service matching and composability;

(3) The service workflow-based semiautomatic service composition method, which is the most widely used one.

In this context, Zhu et al. [43] proposed an ontology-based service process model to support decision-making for maintenance, repair, and overhaul services in the aerospace industry. Cao et al. [44] transformed the service process model into a process structure tree to consider service process structuration and service activities selection simultaneously. Wang et al. [45] advocated a model of service activities composition based on stochastic Petri net. Hu et al. [46] modeled the service process by logical Petri net to study the structural transformation of the service process.

The above-mentioned studies only focused on service activities selection. The impact of resource allocation on service activity has not been considered during the service activities selection. Service activities selection and resource allocation were consequently divided into two independent processes.

In a practical situation, the performance of service activity operation will be altered based on different resource allocation instances. It is necessary to combine service activities selection and resource allocation into a whole process during the PSS configuration optimization. Furthermore, most of the previous studies highlighted multi-objective optimization of the overall solution, but high-importance service activities usually exert a greater impact on the solution, compared to low-importance services. Therefore, the importance degree of service activity is also taken into consideration in this paper, to improve the performance of service activities with higher importance.

Based on the above analysis, it was found that: (a) PSS configuration is an imperative part of PSS design; (b) existing PSS configuration solutions are obtained in the module granularity; and (c) the significant impact of service details on the performance of a PSS configuration solution has not been given sufficient attention in recent research studies. PSS configuration optimization is, therefore, studied from a fine-grained perspective in this paper. The optimization is performed by combining service activities selection and resource allocation. As a result, the PSS configuration solutions provided to the customer will be more refined and precise. In particular, the optimizing objectives, which present the performance of the solutions, will be calculated closer to the actual implementation.

The complex network is chosen as the framework for the PSS modeling because its structural properties enable the exploration of the interactions between products, resources, and service activities. The mathematical characteristics of complex networks are very useful for analyzing the characteristics of complex systems. However, it is difficult for a general complex network to effectively reveal the multilayered or multidimensional relationships in a complex system when the scale of the system increases substantially. Many approaches have been proposed to enrich the representation of networks. Hypernetworks [47–49] are developed based on the definition of the hypergraphs to generalize the multidimensional relationships, in which multi-elements are connected rather than only two elements. Multilayer networks [50] are applied to describe the complex structures of systems, which include different types of relationships between their components. Originally, multilayer networks were proposed to express different types of relationships between one single type of elements in different layers [51,52]. More recently, they have been used to model other systems, including those with different types of elements. Omodei et al. [53] proposed a method based on the multilayer networks of citations and disciplines to assess the interdisciplinary importance of scholars, institutions, and countries. Additionally, Pasqual and Weck [54] introduced a multilayer network model integrating three coupled layers (product layer, change layer, and social layer) to analyze and manage engineering change propagation. Leng and Jiang [55] also offered a multilayer network model comprising processes, machines, and work in processes to assess dynamic scheduling in a radio frequency identification-driven discrete manufacturing system.

Based upon the comparison of the general complex network, hypernetwork, and multilayer network, it is obvious that a multilayer network can perfectly accommodate multiple heterogeneous elements and provide a better description of relationships between heterogeneous elements than a general complex network or hypernetwork. Therefore, a multilayer network, as a special type of complex network, is applied for the modeling of PSS in this paper.

The organization of this paper is as follows. In Section 2, this paper's research problem is described and the framework is then introduced. The PSS multilayer network model is constructed in Section 3. Section 4 establishes the mathematical model of PSS configuration. This section also delineates the improved optimization algorithms and their applications for obtaining the optimal solutions. Moreover, a case is studied in Section 5 to discuss the feasibility and effectiveness of this method. Section 6 highlights the benefits and the contributions to the sustainability of the proposed approach. Finally, conclusions are stated in Section 7.

2. Problem Description and Framework

In product-oriented PSS, products are delivered to customers as tangible objects in a definite time, just after the PSS configuration solution is accepted. However, that is not the case with services.

Services are provided to customers only when the customer requests or the service premises are satisfied and operation of services is affected by the real-time conditions of PSS. Service details, which include service activities selection and resource allocation, are consequently not strictly determined when the PSS configuration solution is accepted. Contracts to implement corresponding service functions are provided to the customer. But service details definitely impact the performance of the PSS configuration solution. Service time depends on service activities selection, cost and reliability are accurately influenced by resource allocation. Thus, the PSS configuration should be optimized by combining service activities selection and resource allocation. In this way, the PSS solutions obtained after optimization will be more precise, and the results based on the optimizing objectives (executing time, executing cost, and executing reliability) will be closer to the actual implementation.

The framework of this fine-grained configuration optimization of the product-oriented PSS is shown in Figure 1. A PSS model is constructed based on a three-layer network, which contains product and services selected by the customer, and the resources in the system. A node in every layer (product layer, service layer, resource layer) of the multilayer network respectively represents an element (product, service, resource). The relationships between homogeneous elements in a layer and the relationships between heterogeneous elements in different layers are represented as edges. With appropriate decomposition, PSS can be clearly described by the proposed multilayer network. Elements and relationships in PSS can be expressed by connection matrixes, which make the description of the mathematical model and realization of optimization more convenient. By using this PSS multilayer network model, the importance degrees of service activities are analyzed based on the attributes of the multilayer network. A mathematical model based on the connection matrixes is constructed to express the objectives and constraints of PSS configuration optimization, and the corresponding solution algorithm is proposed to solve the mathematical model of optimization.

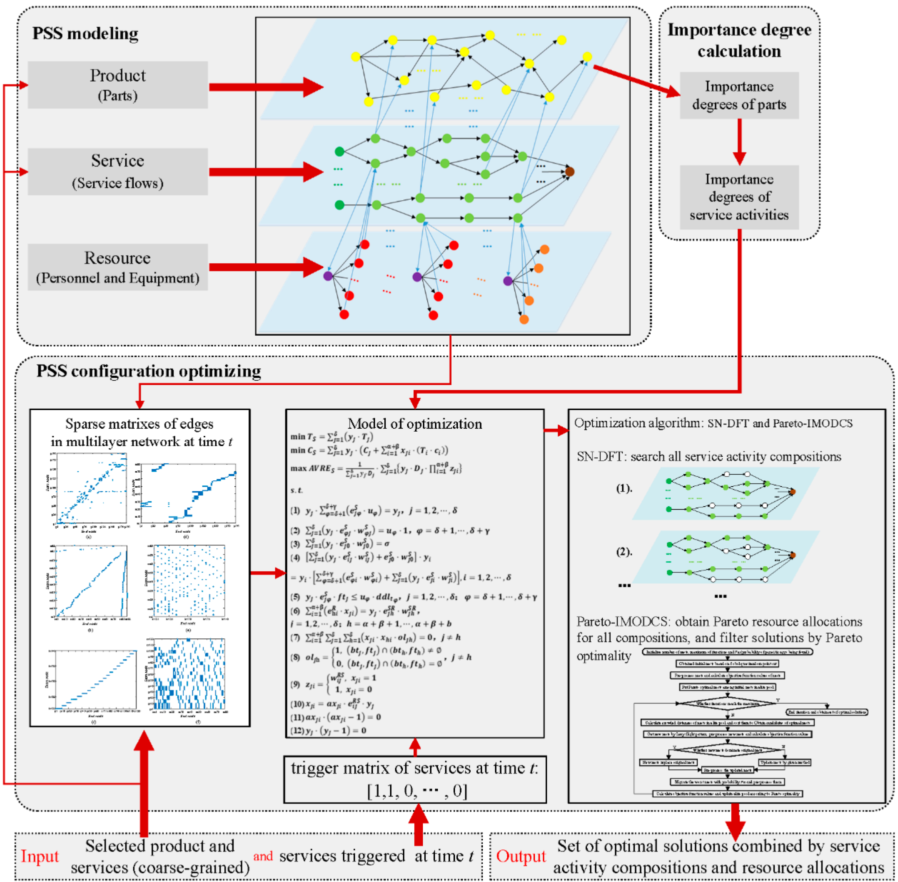


Figure 1. Framework of the proposed method.

3. PSS Multilayer Network Model

The PSS multilayer network (NET_{PSS}) is composed of the product layer (Net_P), service layer (Net_S), resource layer (Net_R), and set of inter-layer edges (E_I) which connect layers to each other. Therefore, there are three types of nodes and five kinds of edges considered in NET_{PSS}. Specifically, nodes in Net_P, nodes in Net_S, and nodes in Net_R are denoted as sets P, S, and R, respectively; edges in Net_P, edges in Net_S, edges in Net_R, inter-layer edges between Net_S and Net_P, and inter-layer edges between Net_S and Net_R are denoted as sets E_P, E_S, E_R, $E_{P,S}$, and $E_{R,S}$, respectively. The inter-layer edges between Net_R and Net_P are not considered in this model.

Then, NET_{PSS} is described by Formula (1):

$$NET_{PSS} = \left\langle V(P, S, R), E(E_P, E_S, E_R, E_{P,S}, E_{R,S}) \right\rangle. \tag{1}$$

A product can be decomposed into parts according to required granularity. Net_P is formed by part nodes (p) connecting to each other by interrelationships between parts. Net_S consists of service flows, each of which is described by a service trigger node (ss), numbers of service activity nodes (sa), and a service end node (se) logically associated with each other. Net_R contains resources required by service activities, including personnel resources (pr) and equipment resources (er), as resource types are denoted as kr. E_I contains $E_{P,S}$, $E_{R,S}$, and the weights on inter-layer edges. Detailed descriptions of Net_P, Net_S, Net_R, and E_I are given in the following subsections.

3.1. Product Layer

A node p_i in Net_P is described by Formula (2):

$$p_i = \langle p\_BAS_i, p\_REC_i, p\_ST_i \rangle, \tag{2}$$

where p_BAS_i is the set of basic information, including information such as name, quality, and material. p_REC_i is the set of accepted service records. p_ST_i is the set of working statuses, including various data of physical status, material status, and environmental status. All statuses vary with time t. For any p_i whose status data cannot be monitored, $p\_ST_i = \varnothing$.

There are four types of interactions among parts [56]: spatial-type interaction, energy-type interaction, information-type interaction, and material-type interaction. These interactions could be extracted from the product design manually. If p_i provides spatial reference (adjacency or orientation) to p_j, then $se_{ij} = 1$. If p_i provides or transmits energy to p_j, then $ee_{ij} = 1$. If p_i provides or transfers information to p_j, then $ie_{ij} = 1$. If p_i determines or impacts the material selection of p_j, then $me_{ij} = 1$. If there are one or more interactions from p_i to p_j, then $e^P_{ij} = 1$, denotes the directed edge from p_i to p_j; otherwise, $e^P_{ij} = 0$. Obviously, $e^P_{ij} \neq e^P_{ji}$.

In order to describe the overall strength of the relationship between p_i and p_j, the weight on e^P_{ij} is represented by the value w^P_{ij}, following Formula (3):

$$w^P_{ij} = w_s \cdot se_{ij} + w_e \cdot ee_{ij} + w_i \cdot ie_{ij} + w_m \cdot me_{ij}, \tag{3}$$

where w_s, w_e, w_i, and w_m, respectively, represent the proportion of se_{ij}, ee_{ij}, ie_{ij}, and me_{ij} in the edge e^P_{ij}. w^P_{ij} is defined in [0,1]. If the four types of interactions are equal in importance, then $w_s = w_e = w_i = w_m = 0.25$.

Net_P is consequently represented by Formula (4):

$$\begin{aligned} Net_P &= \langle P, E_P \rangle; \\ P &= \{p_1, p_2, \cdots, p_n\}; \\ E_P &= \left\{e_{ij}^P\right\}; \\ W_P &= \left\{w_{ij}^P\right\}. \end{aligned} \tag{4}$$

3.2. Service Layer

Services in a product lifecycle can be classified into two types: production services and product services. All services in Net_S are product services.

3.2.1. Service Trigger Node

A node ss_i in Net_S is as described by Formula (5):

$$\begin{aligned} ss_i &= \langle ssP_i, Tri_i, s\_st_i \rangle; \\ Tri_i &= \left\{tri_i\left(p_j\right) \middle| p_j \in ssP_i\right\}; \\ s\_st_i &= \{on(start\_t_i, ddl\_t_i) \vee off\}; \end{aligned} \tag{5}$$

where ssP_i is the set of parts whose working statuses are monitored by ss_i. Tri_i is the set of triggering conditions of ss_i, $tri_i\left(p_j\right)$ is the triggering condition for working statuses p_ST_i of p_i to trigger ss_i. s_st_i denotes the status of ss_i: if triggering conditions are satisfied, the status will be $on(start\_t_i, ddl\_t_i)$, where $start\_t_i$ and ddl_t_i are, respectively, the start time and the deadline of the entire service set according to customer requirements; otherwise, the status will be off.

3.2.2. Service Activity Node

The function of service activity is to change the status of the service activity receiver [57]. Service activity is operated by the specified resources and executed on the specified component. By decomposing the service flow at specified granularity, the corresponding set of service activities could be obtained. A service activity node sa_i in Net_S is as described by Formula (6):

$$\begin{aligned} sa_i &= \langle a\_REC_i, T_i, C_i, RE_i, O_i, a\_st_i \rangle; \\ a\_st_i &= \{off \vee on(bt_i, ft_i)\}; \end{aligned} \tag{6}$$

where a_REC_i is the set of service execution records of sa_i. T_i and C_i are the theoretical executing time and fixed cost for operating sa_i, which are extracted from historical data. RE_i denotes the reliability of that sa_i which can be completed in time T_i. It is represented by the ratio between the number of times that sa_i has been completed in time T_i and the total number of times that sa_i has been executed. $RE_i \in (0, 1]$. O_i is set of service objects of sa_i. a_st_i denotes the status of sa_i: off is idle status, $on(bt_i, ft_i)$ denotes working status. bt_i is the beginning time of sa_i, ending time is defined as $ft_i = bt_i + T_i$.

3.2.3. Service End Node

A service end node se_0 is put in Net_S to distinguish whether the service flow is finished at the end of sa_i. An edge between sa_i and se_0 means that sa_i is the last service activity in the service flow.

3.2.4. Edges in Service Layer

Regarding ss_j and se_0 as the activities of "trigger service" and "end service", respectively, then ss_j, se_0, and sa_i are homogeneous. Edges between sa_i and sa_j, edges between sa_i and ss_j, and edges between sa_i and se_0 are all denoted by e^S; $e_{ij}^S \neq e_{ji}^S$, $e_{ii}^S = 0$. w_{ij}^S defined in the interval [0,1] denotes weight on e_{ij}^S. $w_{ij}^S = 0$ when $e_{ij}^S = 0$. The description of an edge e^S between two nodes considers the following cases:

1. Between ss_j and sa_i: if sa_i is the first one executed in the service flow triggered by ss_j, then $e^S_{ji} = 1$, $bt_i = start\_t_j$. sa_i can participate in different service flows triggered by ss_j, denoted as $e^S_{ij} = 1$ with $w^S_{ij} \to 0$, sa_i will not output any weight to ss_j; $e^S_{ij} = 0$ between ss_i and ss_j;
2. Between se_0 and sa_i: if sa_i is the last one in the service flow being executed, then $e^S_{i0} = 1$; $e^S_{0i} = 0$ for any sa_i and $e^S_{i0} = e^S_{0i} = 0$ between se_0 and ss_i;
3. Between sa_j and sa_i: if sa_i and sa_j are in same service flow and are executed in time sequence, then $e^S_{ij} = 1$. Otherwise, $e^S_{ij} = 0$.

There are three types of correlations among service activities [58]: correlation of function similarity, correlation of function complementary, and no correlation. If two service activities have a correlation of function similarity, their function might be identical, inclusive, or partially similar. If two service activities have a correlation of function complementary, the composable situation might be accurate composition, inclusive composition, blocked composition, or crossed composition.

According to the above correlations, logical relationships among service activity nodes are divided into serial relationship, parallel relationship, and selective relationship. Two activities in a serial relationship are sequentially executed in chronological order. Two activities in a parallel relationship are executed at the same time, and only when they are fully completed will the execution of subsequent activities begin. In a selective relationship, there are multiple activities for selection: if only one of them is executed, then the execution of subsequent activities will begin.

The in-degree and out-degree of sa_i in Net_S are denoted as inw_i and $outw_i$ which are described by Formulas (7) and (8):

$$inw_i = \sum\nolimits_j e^S_{ji} \cdot w^S_{ji}, \tag{7}$$

$$outw_i = \sum\nolimits_j e^S_{ij} \cdot w^S_{ij}. \tag{8}$$

Then, the description of the three types of logical relationships is expanded as follows:

- If sa_h and sa_l are in a serial relationship, and $inw_h = x$, then $e^S_{hl} = 1$, $w^S_{hl} = x$, $bt_l = ft_h$;
- If n nodes $(sa_1, sa_2, \cdots, sa_n)$ are in a selective relationship, and $inw_h = x$, then $e^S_{hi} = e^S_{hj} = e^S_{il} = e^S_{jl} = 1$, $w^S_{hi} = w^S_{hj} = w^S_{il} = w^S_{jl} = x$, $outw_l = x$. In this case, $outw_h > inw_h$ and $outw_l < inw_l$. If sa_i is selected, then $bt_i = ft_h$, $bt_l = ft_i$;
- If n nodes $(sa_1, sa_2, \cdots, sa_n)$ are in a parallel relationship, and $inw_h = x$, then $e^S_{hi} = e^S_{hj} = e^S_{il} = e^S_{jl} = 1$, $w^S_{hi} = w^S_{hj} = w^S_{il} = w^S_{jl} = x/n$, $outw_l = x$, and $bt_j = bt_i = ft_h$, $bt_l = \max\left(ft_i, \cdots, ft_j\right)$. In this case, $outw_h = inw_h$ and $outw_l = inw_l$.

Connections of these logical relationships are shown in Figure 2. If sa_h is replaced by a service trigger node, then $x = 1$.

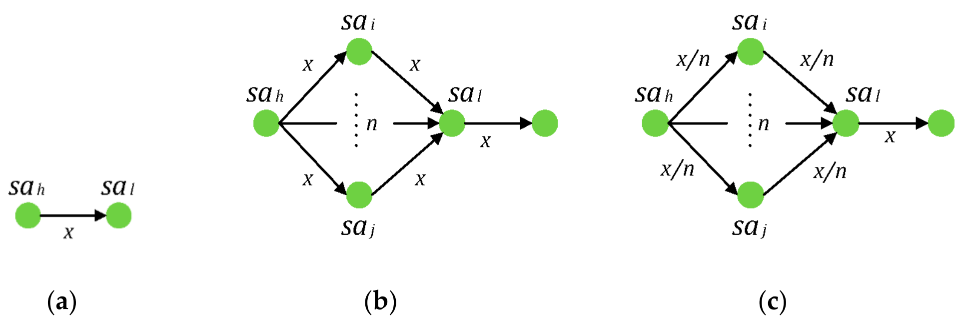


Figure 2. Logical relationships among service activity nodes. (**a**) Serial relationship; (**b**) selective relationship; (**c**) parallel relationship.

Then, the logical relationship between nodes in Net_S can be distinguished by values of *inw* and *outw*. However, if sa_h has multi inputs and multi outputs, as depicted in Figure 3a, the logical relationship between sa_h and other nodes will be uncertain. The virtual service activity node sa_m is applied to prevent that. Then, sa_h will become a node with a single input or single output, as shown in Figure 3b,c. There is not any essential service content in a virtual service activity node, so $T_m = 0$, $C_m = 0$, $RE_m = 1$, $a\_REC_m = \varnothing$, $O_m = \varnothing$, and sa_m does not require any resources.

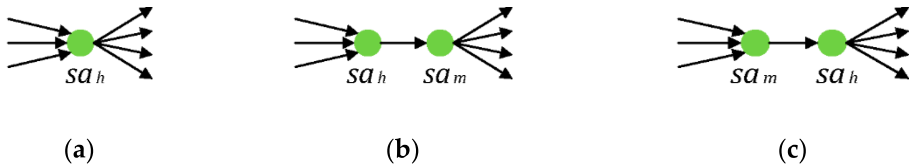


Figure 3. Different connection situations of sa_h. (**a**) Multiple inputs and multiple outputs; (**b**) Multiple inputs and single output; (**c**) Single input and multiple outputs.

If sa_h connects the other nodes, which are in selective relationships and parallel relationships, intricately, it will be impractical to distinguish different relationships by values of inw_h and $outw_h$. As shown in Figure 4a, when $w^S_{hi} = w^S_{hj} = w^S_{hk} = w^S_{hg} = x/2$, relationships among sa_i, sa_j, sa_k, and sa_g cannot be distinguished directly. In order to solve this situation, there will be two situations to integrate in the virtual service activity nodes sa_m and sa_n, as follows:

1. $w^S_{hm} = w^S_{hn} = w^S_{mi} = w^S_{mj} = w^S_{nk} = w^S_{ng} = x/2$ means that sa_i and sa_j are in a selective relationship; sa_k and sa_g are in a selective relationship. Meanwhile, sa_i (or sa_j) and sa_k (or sa_g) are in a parallel relationship, as expressed in Figure 4b.
2. $w^S_{hm} = w^S_{hn} = x, w^S_{mi} = w^S_{mj} = w^S_{nk} = w^S_{ng} = x/2$ means that sa_i and sa_j are in a parallel relationship; sa_k and sa_g are in a parallel relationship. Meanwhile, two parallel clusters are in a selective relationship, as indicated in Figure 4c.

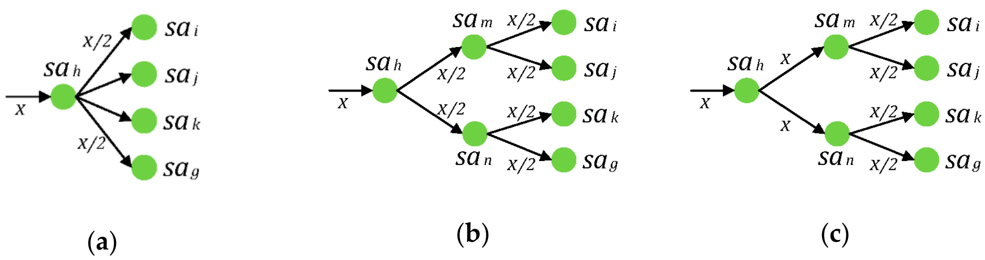


Figure 4. The logical relationship between sa_i and sa_j. (**a**) It cannot be distinguished; (**b**) It is selective relationship; (**c**) It is parallel relationship.

Assuming that the number of *sa* and *ss* in Net_S are indicated, respectively, by δ and γ, and there is only one *sa*, then the description of Net_S is consequently shown by Formula (9):

$$\begin{gathered} Net_S = \langle S, E_S \rangle; \\ S = \left\{s_0, s_1, s_2, \cdots, s_{\delta+\gamma}\right\} = \left\{se_0, sa_1, sa_2, \cdots, sa_\delta, ss_{\delta+1}, ss_{\delta+2}, \cdots, ss_{\delta+\gamma}\right\}; \\ E_S = \left\{e^S_{ij}\right\}; \\ W_S = \left\{w^S_{ij}\right\}. \end{gathered} \tag{9}$$

3.3. Resource Layer

The node pr_i or er_i in Net_R is described by Formula (10):

$$\begin{array}{c} r_i = pr_i \vee er_i = \langle r\_REC_i, c_i, r\_st_i \rangle; \\ r\_st_i = idle \vee busy; \end{array} \tag{10}$$

where r_REC_i is the set of records of services operated by r_i, c_i is the cost for r_i working in one unit of time, r_st_i denotes the status of r_i. If $r\_st_i = idle$, then r_i could accept a new assignment of the service activity; if $r\_st_i = busy$, it means r_i is in working status and will not accept any new assignments until the service activity r_i is working on is finished. If there are not any resource contents in kr_j, then $r\_REC_j = \varnothing$, $c_j = 0$, $r\_st_j = \varnothing$.

If there is a subordinate relationship between $pr_i \vee er_i$ and kr_j, which means that the resource type of $pr_i \vee er_i$ is kr_j, then $e^R_{ji} = 1$. This denotes that there is a directed edge from the father node kr_j to the child node $pr_i \vee er_i$. Otherwise, $e^R_{ji} = 0$. For any $pr_i \vee er_i$, $\sum_j e^R_{ji} = 1$.

Assuming that quantities of pr, er and kr in Net_R are indicated by α, β, and b, respectively, then the description of Net_R is shown by Formula (11):

$$\begin{array}{c} Net_R = \langle R, E_R \rangle; \\ R = \left\{ pr_1, \cdots, pr_\alpha, er_{\alpha+1}, \cdots, er_{\alpha+\beta}, kr_{\alpha+\beta+1}, \cdots, kr_{\alpha+\beta+b} \right\} = \left\{ r_1, r_2, \cdots, r_{\alpha+\beta+b} \right\}; \\ E_R = \left\{ e^R_{ji} \right\}, (i = 1, 2, \cdots, \alpha+\beta; j = \alpha+\beta+1, \cdots, \alpha+\beta+b). \end{array} \tag{11}$$

3.4. Inter-Layer Edges

The service content is determined by the working status of products, while service activities are executed on products by required resources. In the PSS multilayer network, some inter-layer edges exist between Net_S and Net_P, the rest exist between Net_S and Net_R.

3.4.1. Inter-Layer Edges Between Net_S and Net_P

For p_i and sa_j, if $p_i \in O_j$, then $e^{SP}_{ji} = 1$, which denotes the directed inter-layer edge from sa_j to p_i. This means that p_i is one of the service objects of sa_j. Otherwise, $e^{SP}_{ji} = 0$. Meanwhile, $e^{SP} = 1$ only makes sense between p and sa; for ss or se, $e^{SP} = 0$.

Therefore, the set of inter-layer edges between Net_S and Net_P is as described by Formula (12):

$$E_{P,S} = \left\{ e^{SP}_{ji} \right\}. \tag{12}$$

3.4.2. Inter-Layer Edges Between Net_S and Net_R

Inter-layer edges between Net_S and Net_R include directed inter-layer edges from service nodes to resource nodes and directed inter-layer edges from resource nodes to service nodes. Their descriptions are depicted below:

- The execution of service activities demands the resources be in corresponding specific kinds and quantities. If sa_j demands the resources whose type is kr_i, then $e^{SR}_{ji} = 1$, which denotes a directed inter-layer edge from sa_j to kr_i and the quantity demand is represented by integer weight w^{SR}_{ji}. Otherwise, $e^{SR}_{ji} = 0$, $w^{SR}_{ji} = 0$. Meanwhile, $e^{SR} = 1$ only makes sense between kr and sa; for other nodes, $e^{SR} = 0$;
- If $e^{SR}_{jh} = 1$, $e^R_{hi} = 1$, and $r\_st_i = idle$, it means that $pr_i \vee er_i$ is assignable to sa_j, then $e^{RS}_{ij} = 1$, which denotes a directed inter-layer edge from $pr_i \vee er_i$ to sa_j. Otherwise, $e^{RS}_{ij} = 0$. Meanwhile, $e^{RS} = 1$ only makes sense between $pr \vee er$ and sa; for other nodes, $e^{RS} = 0$. The reliability of sa_j being completed in time T_j by $pr_i \vee er_i$ is denoted by weight w^{RS}_{ij} in [0,1].

Every pr_i has a different knowledge level and proficiency level about different *sa*. Assuming PA_i denotes the number of times that sa_j has been completed in the time T_j by pr_i, PB_i denotes the total number of times that pr_i has participated in the execution of sa_j, then w_{ij}^{RS} between pr_i and sa_j is represented by PA_i/PB_i. Every er_i has a constant function in a different *sa*. Assuming EA_i denotes the number of times that any *sa* has been completed in time by er_i and EB_i denotes the total number of times that er_i has participated in the execution of any *sa*, then w_{ij}^{RS} between er_i and sa_j is represented by EA_i/EB_i.

The set of inter-layer edges between Net_S and Net_R, and the set of weights on these inter-layer edges are described by Formula (13):

$$\begin{aligned} E_{R,S} &= \left\{ e_{ij}^{RS}, e_{ji}^{SR} \right\}; \\ W_{R,S} &= \left\{ w_{ij}^{RS}, w_{ji}^{SR} \right\}. \end{aligned} \tag{13}$$

Therefore, E_I is as given by Formula (14):

$$E_I = \left\langle E_{P,S}, E_{R,S} \right\rangle. \tag{14}$$

4. PSS Configuration Optimization

In most practical cases, many services are usually triggered at the same time. Consequently, the configuration optimization should be designed by considering multi-services. After any *ss* is triggered, service activities selection and resource allocation should be accomplished under given constraints to obtain the PSS configuration under constraint conditions.

Let us assume that the number of triggered *ss* is σ up to the point of time t. If ss_φ is triggered and is represented by u_φ following Formula (15), then the situation of all *ss* is represented by trigger matrix $\left[u_{\delta+1}, u_{\delta+2}, \cdots, u_{\delta+\gamma}\right]$ with $\sigma = \sum_{\varphi=\delta+1}^{\delta+\gamma} u_\varphi$:

$$u_\varphi = \begin{cases} 1, & ss_\varphi \text{ is triggered} \\ 0, & otherwise \end{cases}. \tag{15}$$

As a result, service activities selection is regarded as a selection of the paths with multi *ss* as start points in Net_S. It is necessary to consider the executing time, cost, and reliability of *sa* in the selective relationship. The allocation of resources is viewed as resource allocation for multitasks (service activities), in which the cost and reliability of different r_i are considered. Therefore, the services solution is optimized with the goal of executing time minimization, cost minimization, and reliability maximization.

4.1. Importance Degree of Service Activity

When a certain type of resource is required by several service activities simultaneously, then those resources with higher reliability should be allocated to more important service activities. Product parts are the objects of service activities; hence, the importance degree of product parts exerts a decisive influence on the importance degrees of service activities. Consequently, the importance degree of product parts is taken as the base of calculation for the importance degree of service activities, which provides a theoretical reference for the resource allocation.

Today, network-based importance measurement is thoroughly developed and widely applied. Various measure methods can be summarized as:

(1) Importance calculation methods based on neighbor relationships between nodes, such as degree centrality or semi-local centrality [59];

(2) Path-based importance calculation methods, such as closeness centrality and betweenness centrality;

(3) Importance calculation methods based on eigenvectors, such as search algorithm LeaderRank [60] and SRank [61] based on algorithm PageRank [62];

(4) Importance calculation methods based on node removal and contraction, in which the importance degree of nodes is measured by the variation in network attributes after node removal or contraction [63].

4.1.1. Importance Degree of Product Parts

Importance calculation methods based on eigenvector are applied in this paper. In the concept of PageRank and LeaderRank, the importance of a page in a network is determined by the number and the quality of other pages pointing to it. However, the importance of p_i in Net_P is ascertained by the number and quality of other nodes it points to. This makes PageRank and LeaderRank unsuitable for calculating the importance degree of product parts. Referring to the weighted LeaderRank [64], PartRank is proposed to calculate the importance degree of product parts. Details are illustrated as follows.

A ground node p_g, which connects with every part node by a bidirectional edge, is introduced. Then $e^P_{gi} = 1$ and $e^P_{ig} = 1$, $\forall i \neq g$. Based on the concept of weighted LeaderRank, nodes with higher input intensity embody a higher probability of being visited by p_g. Inversely, p_i with a higher output intensity is more likely to visit p_g in Net_P. w^P_{ig} is consequently described by formula (16), where θ is a free parameter, n is total number of nodes in Net_P (excluding p_g). w^P_{gi} is described as Formula (17):

$$w^P_{ig} = \left(\sum\nolimits_{h=1}^{n+1} e^P_{ih} \cdot w^P_{ih}\right)^{\theta}; \tag{16}$$

$$w^P_{gi} = \frac{\sum_{h=1}^{n} e^P_{hi} \cdot w^P_{hi}}{\sum_{h=1}^{n} e^P_{hi}}. \tag{17}$$

The importance degrees are calculated iteratively until they reach a steady state. Assuming that calculation reaches a steady state at step k, then the importance degree of p_g will be allotted to every p_i. The importance degree of p_i is denoted as PR_i, calculated as Formula (18),

$$\begin{cases} PR_g(0) = 0 \\ PR_i(0) = 1, \forall i \neq g \\ PR_i(t) = \sum_{j=1}^{n+1} \frac{e^P_{ij} \cdot w^P_{ij}}{k^{in}_j} \cdot PR_j(t-1), t = 1, 2, \cdots, k \\ PR_i = PR_i(k) + \frac{w^P_{ig}}{k^{in}_g} \cdot PR_g(k) \end{cases}, \tag{18}$$

where $k^{in}_j = \sum_{h=1}^{n+1}\left(e^P_{hj} \cdot w^P_{hj}\right)$ is the input intensity of p_i in Net_P.

4.1.2. Importance Degree of Service Activities

Knowing that the service object of sa is p and if sa is operated on p with a high importance degree, a designer should pay more attention to that sa. CNI_j denotes the cross layer importance degree of sa_j, which is related to the product parts on which sa_j operates. The value of CNI_j is determined by p_i with the highest PR_i in the set of service objects of sa_j. CNI_j is consequently calculated by Formula (19):

$$CNI_j = \max_{1 \leq i \leq n}\left(e^{SP}_{ji} \cdot PR_i\right). \tag{19}$$

After being normalized, the importance degree of sa_j is denoted as $D_j = CNI_j/\max$, where max is the maximum of all CNI_j in Net_S. Therein, the virtual service activity node does not have any service object, so the importance degree of every virtual service activity node is consequently 0.

4.2. Mathematical Model

Define the binary variable y_j to indicate whether sa_j is selected, and define the binary variable ax_{ji} to indicate whether to allocate r_i to sa_j, by the Formulas (20) and (21), respectively:

$$y_j = \begin{cases} 1, & sa_j \text{ is selected} \\ 0, & otherwise \end{cases}, \tag{20}$$

$$ax_{ji} = \begin{cases} 1, & allocate\ r_i\ to\ sa_j \\ 0, & otherwise \end{cases}. \tag{21}$$

The mathematical model of PSS configuration optimization is constructed by Formulas (A1)–(A15) in Appendix A. The objective functions are given by Formulas (A1)–(A3) and constraints are given by the Formulas (A4)–(A15).

The objective function of executing time minimization is described by Formula (A1). The executing times of service activities are assumed to be fixed parameters, and they are not impacted by resource allocation. T_S denotes the sum of executing times for all selected sa.

The objective function of cost minimization is described by Formula (A2). The executing cost of sa_j includes the fixed cost C_j and the work costs of all resources allocated to sa_j, which are related to the executing times of sa_j. C_S denotes the sum of executing costs of all selected sa.

The objective function of reliability maximization is described by Formula (A3). The executing reliability of sa_j is determined by w_{ij}^{RS} of all r_i that are allocated to sa_j. Therefore, a logic function z_{ji}, which depends on x_{ji} as the formula (A12) is defined. Then, $exre_j$, denoting the executing reliability of sa_j, is described as $exre_j = \prod_{i=1}^{\alpha+\beta} z_{ji}$. The target of reliability is always to maximize the executing reliability of the whole solution, which is related to the $exre_j$ of all selected sa_j. However, if a certain type of resource is required by several service activities with overlapping service times, the allocation result of that type tends to make a sa_j with a higher importance degree for reaching a higher $exre_j$. Therefore, $AVRE_S$ is defined as the average executing reliability of all selected sa_j, which is shown by Formula (22):

$$AVRE_S = \frac{\sum_{j=1}^{\delta}\left(y_j \cdot D_j \cdot exre_j\right)}{\sum_{j=1}^{\delta} y_j \cdot D_j}. \tag{22}$$

The $exre_j$ of a sa_j with higher D_j has a greater impact on the value of $AVRE_S$.

When $y_j = 1$ and r_i is assigned to sa_j, allocating r_i to sa_j might be feasible. Therefore, x_{ji} is defined by the formula (A13). Here, $x_{ji} = 1$ means r_i is allocated to sa_j successfully, $x_{ji} = 0$ means r_i is not allocated to sa_j. Constraints for y_j and ax_{ji} are shown by Formulas (A14) and (A15). Constraints for the three objective functions are developed as indicated below:

1. All selected service activities must be part of the construction for the triggered service flows. They are represented by Formula (A4);
2. Every service flow starts with the triggered ss_φ, and the out-degree of ss_φ is 1. This is expressed by Formula (A5);
3. The number of triggered ss is σ. All triggered service flows end at se_0. Consequently, the in-degree of se_0 is σ. This is shown by Formula (A6);
4. If all selected service activities constitute complete service flows, there will not be any selective relationship between selected sa_i and sa_j. As a result, the sum of weights on edges from triggered ss_φ and selected sa_j to sa_i will equal to the sum of weights on edges from sa_i to se_0 and selected sa_j. This is shown by Formula (A7);
5. For every triggered ss_φ, $start\_t_\varphi = t$. Every triggered flow must be completed within the service time range $[start\_t_i, ddl\_t_i]$ requested by a customer, as illustrated in Formula (A8);

6. The resources allocated to the selected sa_j must satisfy the constraints as the resource type and the quantity demands of sa_j. This constraint is denoted by Formula (A9);
7. Every r_i can only participate in a maximum of one service activity at the same time. For all service activities that r_i is allocated to, their service time intervals cannot overlap. Defining ol_{jh} indicates whether there is any overlap between the service time intervals of sa_j and sa_h, as shown in Formula (A11). That constraint is described by Formula (A10).

4.3. Obtaining the Optimal Solutions

To obtain the optimal solutions by the trigger matrix $\left[u_{\delta+1}, u_{\delta+2}, \cdots, u_{\delta+\gamma}\right]$ and the proposed mathematical model, the solving procedures are as follows.

Step 1: Obtaining SAS_i as the service activities solution set for ss_i, the number of solutions is denoted as n_i. If $u_i = 1$, obtain $SAS_i = \left\{SA_1^i, \cdots, SA_{x_i}^i, \cdots, SA_{n_i}^i\right\}$ by Depth-First Traversal (DFT), wherein SA_x^i is the set of all service activities selected in the x_i time traversal; if $u_i = 0$, ss_i is not activated, then $n_i = 1$, $SAS_i = \left\{SA_1^i\right\}$, wherein $SA_1^i = \varnothing$. Net_S is different from ordinary complex networks, and as a consequence, the traditional DFT is unsuitable for Net_S. Depth-First Traversal based on service layer (SL-DFT) is designed and detailed in Section 4.3.1.

Step 2: Solutions of service activities for each triggered ss_i are arranged by permutation and combination. Obtaining a solutions set of service activities as $SAS_{all} = \left\{SA_1^{all}, SA_2^{all}, \cdots, SA_{n_{all}}^{all}\right\}$ for system, $n_{all} = \prod_v n_v$, $SA_x^{all} = \left\{SA_{x_{\delta+1}}^{\delta+1}, SA_{x_{\delta+2}}^{\delta+2}, \cdots, SA_{x_{\delta+\gamma}}^{\delta+\gamma}\right\}$.

Step 3: Using the Pareto-improved multiple-objective discrete cuckoo search algorithm (Pareto-IMODCS) [23] to obtain the set of optimal solutions of resource allocation for each SA_i^{all} as $RAS_i = \left\{RA_1^i, \cdots, RA_{m_i}^i\right\}$, the number of solutions is denoted as m_i. Pareto optimality and Pareto-IMODCS are detailed in Sections 4.3.2 and 4.3.3, respectively.

Step 4: Merging all RAS_i into one as $RAS_{all} = \left\{RA_1^{\delta+1}, \cdots, RA_{m_{\delta+1}}^{\delta+1}, \cdots\cdots, RA_1^{\delta+\gamma}, \cdots, RA_{m_{\delta+\gamma}}^{\delta+\gamma}\right\}$, filter RAS_{all} to obtain the set of non-dominated solutions as RAS_{pareto} based on Pareto optimality. RAS_{pareto} is the final set of optimal solutions.

4.3.1. Obtaining Solutions of Service Activities by SL-DFT

Obtaining SAS_v by SL-DFT involves searching all paths from ss_v to se_0 in Net_S. Assuming the current visited node is s_x:

- If $\sum_{i=0}^{\delta} e_{xi}^S = 1$ and $e_{xy}^S = 1$, then add node s_y into the path and continue searching from s_y;
- If $\sum_{i=0}^{\delta} e_{xi}^S > 1$ and $outw_x > inw_x$, nodes satisfying $e_{xi}^S = 1$ are in a selective relationship, then select one of these to visit. Assuming that the selected node is s_y, continue to search from s_y until all paths from s_y to se_0 are obtained, then backtrack to s_x, selecting an unvisited node from nodes satisfying $e_{xi}^S = 1$ to visit. This new search forms a new path, so nodes visited in the search from s_y can still be visited. Repeat this step until all nodes satisfying $e_{xi}^S = 1$ are visited;
- If $\sum_{i=0}^{\delta} e_{xi}^S > 1$ and $outw_x = inw_x$, it means nodes satisfying $e_{xi}^S = 1$ are in a parallel relationship. Then, select one to visit and continue searching until the visited node s_z satisfies $\sum_{i=0}^{\delta} e_{iz}^S > 1$ and $inw_z = outw_z$, which means s_z is the confluence node of these parallel paths. Then, backtracking to s_x, select an unvisited node from nodes satisfying $e_{xi}^S = 1$ to visit. Repeat this step until nodes satisfying $e_{xi}^S = 1$ are all visited. Add all nodes visited in this step into the path and continue to search from s_z.

After all paths from s_x to se_0 are obtained, if s_x is a service trigger node, then end the search. Otherwise, backtrack to the node before s_x. Based on the above search, service activity compositions obtained by SL-DFT satisfy the constraints as given by Formulas (A4)–(A7).

4.3.2. Pareto Optimality

In the optimizing process, the solution with smaller executing time, smaller executing cost, and greater executing reliability is considered better. However, objectives in one solution might conflict with each other in reality. Traditional linear weighting methods are unsuitable for solving such multi-objective optimization, which could be solved based on Pareto optimality [65].

If x_i and x_j satisfy that $T_S(x_i) \leq T_S(x_j)$, $C_S(x_i) \leq C_S(x_j)$ and $AVRE_S(x_i) \geq AVRE_S(x_j)$, and at least one of these inequalities is a strict inequality, then x_i dominates x_j. If x_m is not dominated by any x, then x_m is a non-dominated solution, which is the Pareto optimal solution.

4.3.3. Obtaining Optimal Solutions of Resource Allocation by Pareto-IMODCS

Let us assume that the number of sa in SA^{all} is v regardless of ss, se, and virtual service activity nodes. The types and quantities of resources that every sa demands can be determined based on the multilayer network model. Then, each possible resource allocation solution is represented by a cuckoo nest: the dimension $L(xK_xN_{xK_x})$ of nest represents the N_{xK_x}th resource instance for the K_xth resource type demand of the xth sa in set SA^{all}; various instances in each dimension are all represented by decimal integers. The subscript number of the service activity node corresponding to the xth sa is denoted as $nsa(xK_x)$. The subscript number of the resource type node corresponding to the K_xth resource type is denoted as $nr(xK_x)$. $nsa(xK_x)$ and $nr(xK_x)$ facilitate in decoding the nest into the corresponding resource allocation solutions, which satisfy the constraint defined by Formula (A9).

The original Pareto-IMODCS algorithm is detailed in reference [23]. If multiple sa in SA_i^{all} demand same type of resources in the same time interval, the solution obtained by the original Pareto-IMODCS algorithm might allocate the same resource instance to multiple sa simultaneously. This does not comply with the constraint described by Formula (A10). To avoid this, a "nest pre-processing" step is added to Pareto-IMODCS. The specific procedures of the improved Pareto-IMODCS algorithm to obtain Pareto optimal solutions for each SA_i^{all} are shown in Figure 5.

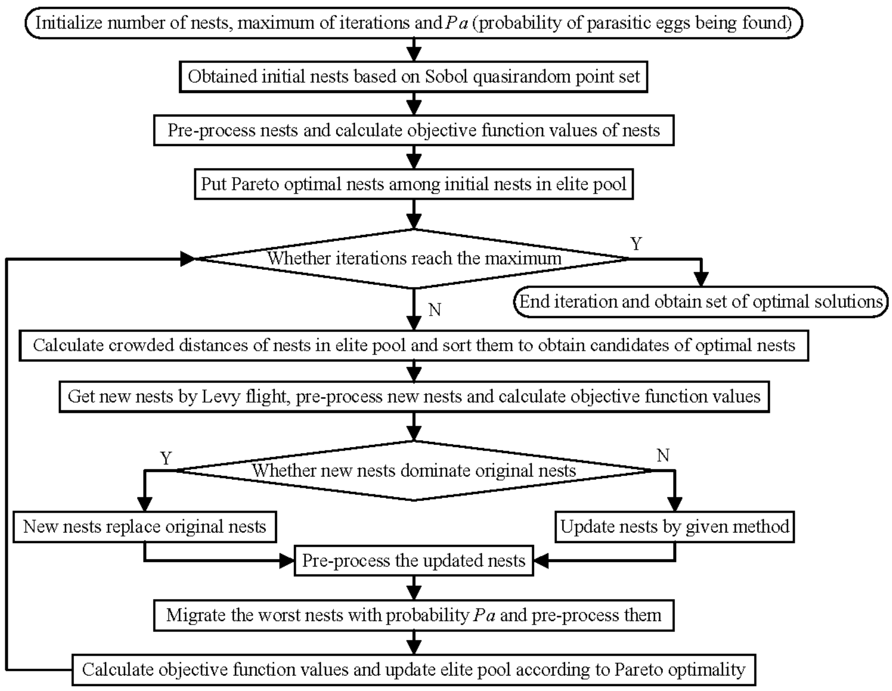


Figure 5. Procedures of the improved Pareto-improved multiple-objective discrete cuckoo search algorithm (Pareto-IMODCS) algorithm.

The procedure "nest pre-processing" is specified as: for any two dimensions as $L(hij)$ and $L(lmn)$, when $nr(hi) = nr(lm)$ and $L(hij) = L(lmn) = x$, it means that the same resource instance is allocated to multiple dimensions. If $nsa(hi) = nsa(lm)$ or $ol_{nsa(hi)\ nsa(lm)} = 1$, it denotes that service activities related to $L(hij)$ and $L(lmn)$ are the same one or their working time intervals overlap, then $L(hij) = x$ and $L(lmn) = x + 1$.

5. Case Study

In this section, a case study for the PSS of an automobile enterprise is provided to validate the proposed multilayer network-based PSS model and the fine-grained PSS configuration optimization method. A product was delivered to a customer with a contract to implement eight selected services (running system maintenance, steering system maintenance, braking system repair, etc.). These services could be decomposed as service flows consisting of service activities. For example, running system maintenance was decomposed as the maintenance of axles, maintenance of suspensions, maintenance of shock absorbers, maintenance of wheels, and maintenance of frames. As indicated, the PSS of the automobile enterprise was modeled as a multilayer network, which included 193 product nodes, 89 service nodes (1 *se*, 80 *sa*, and 8 *ss*), and 140 resource nodes (70 *pr*, 50 *er*, and 20 *kr*).

When services are triggered, a set of PSS configuration solutions should be provided to the customer for making a final decision. The customer's desired performances for the final solution were the minimal executing time and cost and maximal executing reliability. The designer had a PSS configuration optimization task at time *t*, which contained the configuration of services as running system maintenance (ss_{81}) and steering system maintenance (ss_{82}). The network status of the case at time *t* is modeled in Figure 6, in which the sparse matrixes of edges denoting corresponding relationships among nodes in multilayer network are detailed in Figure 7.

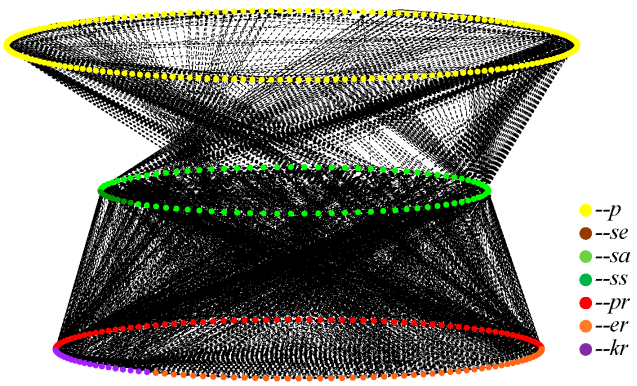
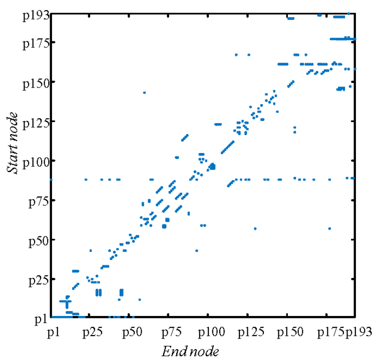
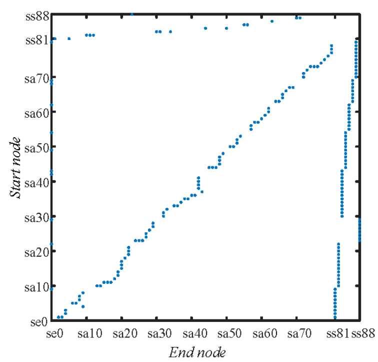


Figure 6. Network status at time *t*.



(**a**)



(**b**)

Figure 7. *Cont.*

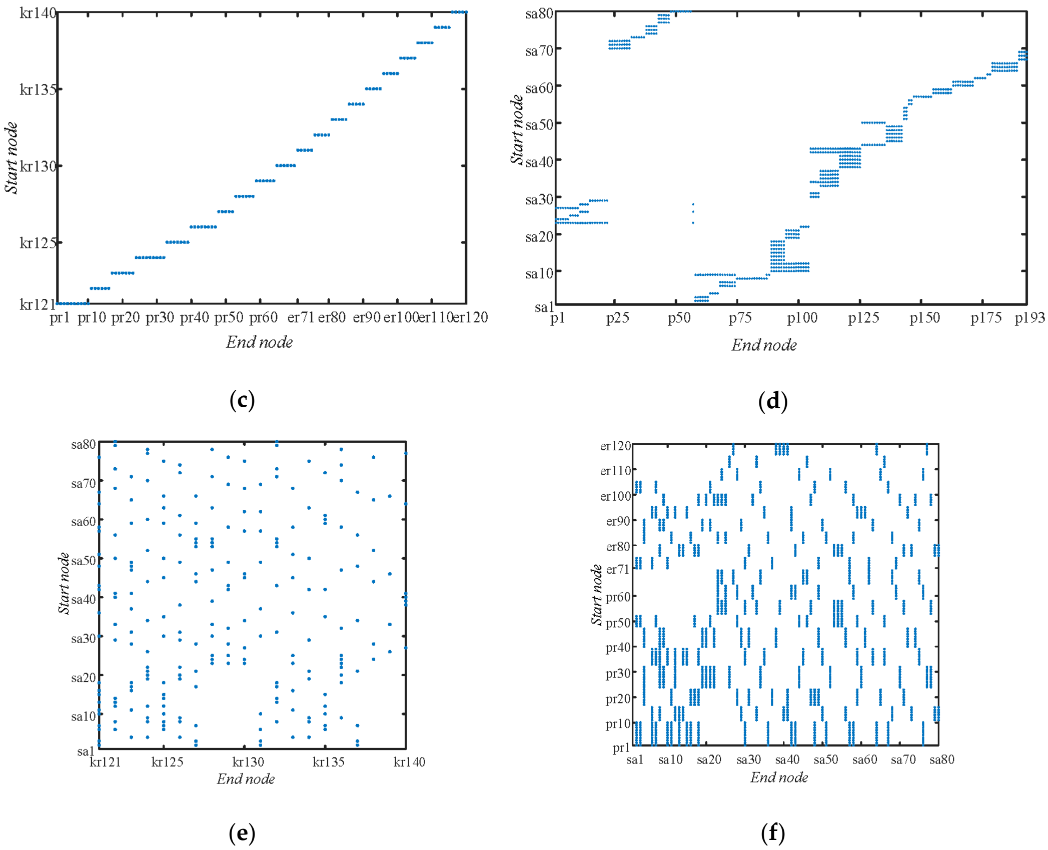


Figure 7. The sparse matrixes of edges at time t. (**a**) Sparse matrixes of edges in Net_P; (**b**) sparse matrixes of edges in Net_S; (**c**) sparse matrixes of edges in Net_R; (**d**) sparse matrixes of inter-layer edges from Net_S to Net_P; (**e**) sparse matrixes of inter-layer edges from Net_S to Net_R; (**f**) sparse matrixes of inter-layer edges from Net_R to Net_S.

According to the PSS configuration optimization task, the trigger matrix at time t was [1 1 0 0 0 0 0 0]. Structures of triggered service flows in service layer are detailed in Figure 8. The executing time, cost, and reliability of related sa_j are detailed in Table 1. It is evident that sa_1 and sa_5 were virtual service activity nodes. The cost for all r_i working in one unit of time is shown in Figure 9.

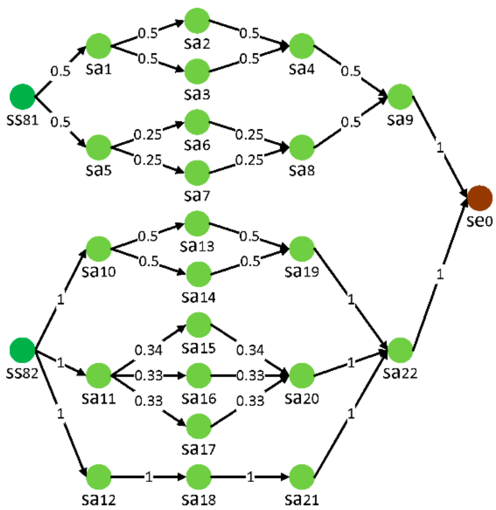


Figure 8. Triggered service flows in the service layer at time t.

Table 1. The executing time, cost, and reliability of related service activities.

sa_i	T_i (h)	C_i (10^3 yuan)	RE_i	sa_i	T_i (h)	C_i (10^3 yuan)	RE_i
1	0	0	1.0000	13	2.6	3.10	0.9897
2	4.5	1.93	0.9721	14	1.3	1.61	0.9816
3	1.9	4.24	0.9984	15	2.4	2.17	0.9949
4	3.1	2.57	0.9811	16	1.6	2.31	0.9793
5	0	0	1.0000	17	2.0	1.98	0.9996
6	2.8	3.69	0.9846	18	3.1	3.73	0.9912
7	3.5	2.21	0.9930	19	1.8	4.19	0.9838
8	2.4	0.98	0.9840	20	2.8	1.91	0.9704
9	2.3	2.47	0.9949	21	1.3	3.56	0.9880
10	3.1	0.99	0.9974	22	2.3	2.87	0.9779
11	1.4	2.31	0.9859	…	…	…	…
12	2.9	1.81	0.9881				

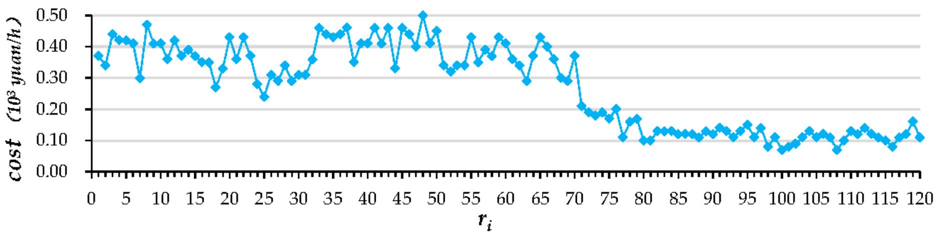


Figure 9. Cost for all resources.

In the above-mentioned network status and solution constraints, all paths from triggered ss_{81} to se_0 and all paths from triggered ss_{82} to se_0 were searched by SL-DFT. Then, one from both sets of paths was selected to form service activity compositions. Pareto optimal solutions of resource allocation for every service activity composition were obtained by using the mentioned improved Pareto-IMODCS algorithm. Relevant parameters of the algorithm were as follows: the number of initial nests was 200, the probability of cuckoo eggs being found was 0.25, the maximum number of iterations for a nest was 3000, and the number of times of an iteration run was 10. Then, solutions of all compositions were filtered according to Pareto optimality. As a result, 14 non-dominated solutions were obtained, which were recorded as set A (Table 2).

Table 2. Solutions in set A.

No.	1	2	3	13	14
sa_i (r_j)	1 2(6,10,52,73,102) 4(18,25,42,85,90) 5 6(4,15,35,72,93) 7(7,39,51,94,103) 8(16,28,38,43,77,88) 9(28,47,90,98) 10(2,38,74,95) 13(2,15,78) 14(12,37,79) 19(26,29,44,89) 22(29,43,44,99)	1 3(2,10,51,75,105) 4(19,25,42,85,90) 5 6(9,15,39,73,94) 7(7,36,52,95,102) 8(14,31,36,47,80,87) 9(31,47,90,98) 10(1,38,72,92) 13(2,16,78) 14(15,38,77) 19(28,29,40,88) 22(27,44,46,100)	1 3(2,9,52,75,105) 4(21,25,42,85,90) 5 6(4,16,39,73,95) 7(6,36,51,93,102) 8(14,31,37,46,79,86) 9(28,46,90,98) 10(7,38,72,94) 13(7,16,77) 14(13,38,78) 19(26,29,44,87) 22(29,44,47,100)	1 3(5,8,51,72,102) 4(21,25,42,85,90) 5 6(10,16,35,74,95) 7(7,36,52,93,103) 8(16,29,38,44,79,86) 9(28,47,90,98) 12(13,29,38,94) 18(2,18,19,37,39,80,100) 21(31,52,89,102) 22(29,42,45,99)	1 3(1,10,51,73,102) 4(19,29,42,83,90) 5 6(2,14,35,75,94) 7(7,36,52,95,103) 8(16,31,34,44,80,88) 9(31,47,90,98) 12(13,25,37,91) 18(2,18,21,38,39,77,100) 21(32,52,88,103) 22(29,41,44,100)
T_S	29.7	27.1	27.1	25.6	25.6
C_S	63.08	61.55	61.77	63.13	62.72
$AVRE_S$	0.9760	0.9724	0.9746	0.9719	0.9699

5.1. Comparison with Method For Selecting Service Activities Before Resource Allocation

To validate the effectiveness of combining service activities selection and resource allocation, the same case was solved with another method as a contrast. In summary, the details were as follows:

Firstly, all paths from triggered ss to se_0 were searched by SL-DFT and all service activity compositions were obtained. Next, all compositions were filtered by objective functions as executing time, cost, and reliability to obtain Pareto optimal service activity compositions. Then, Pareto optimal solutions of resource allocation for each composition were acquired using the improved Pareto-IMODCS with same parameters as last time. Finally, all solutions were filtered according to Pareto optimality. As a result, six non-dominated solutions were recorded as set B (Table 3).

Table 3. Solutions in set B.

No.	1	2	5	6
sa_i (r_j)	1 3(1,10,51,73,102) 4(19,29,42,83,90) 5 6(2,14,35,75,94) 7(7,36,52,95,103) 8(16,31,34,44,80,88) 9(31,47,90,98) 12(13,25,37,91) 18(2,18,21,38,39,77,100) 21(32,52,88,103) 22(29,41,44,100)	1 3(5,8,51,72,102) 4(21,25,42,85,90) 5 6(10,16,35,74,95) 7(7,36,52,93,103) 8(16,29,38,44,79,86) 9(28,47,90,98) 12(13,29,38,94) 18(2,18,19,37,39,80,100) 21(31,52,89,102) 22(29,42,45,99)	1 2(6,7,52,73,102) 4(19,25,42,83,90) 5 6(8,16,35,75,95) 7(10,39,51,92,103) 8(16,28,39,43,78,86) 9(31,47,90,98) 12(12,28,36,93) 18(4,18,23,37,38,80,98) 21(32,52,89,103) 22(29,43,45,100)	1 2(6,7,52,75,103) 4(21,25,42,83,90) 5 6(10,16,35,72,95) 7(4,37,51,93,105) 8(16,28,39,44,78,86) 9(28,46,90,98) 12(15,29,38,91) 18(9,18,23,36,38,80,98) 21(29,51,88,102) 22(32,45,47,100)
T_S	25.6	25.6	28.2	28.2
C_S	62.72	63.13	65.64	64.95
$AVRE_S$	0.9699	0.9719	0.9732	0.9721

By analyzing solutions in Tables 2 and 3, it can be found that set A and set B share four common solutions. Nevertheless, set A also contains other solutions in which service activity compositions are different from solutions in set B. The other two solutions in set B (solutions B5 and B6) are dominated by solution A3. Obviously, combining the service activities selection and resource allocation effectively enriches the diversity of the optimal solution set by retaining more non-dominated solutions. The last ones might have been hindered by using the method where service activities selection finished individually before resource allocation.

5.2. Comparison with the Method Using Different Objective Functions

By calculating the importance degrees of service activities related to triggered ss_{81} and ss_{82}, it was found that the importance degree of sa_9 is significantly higher than other service activities. Reverting to the actual setting of the case, sa_9 was frame maintenance in the running system maintenance. The frame is the base of the whole automobile assembly, which makes frame maintenance more important compared to other activities in the maintenance of the running system and the steering system.

To validate the maximization of $AVRE_S$, an objective function which improves the executing reliability of service activities with higher importance degrees, the same case was solved in another way, as follows.

The value 1 was given to the weights of all $exre_j$ in the objective function of executing reliability, regardless of the importance degrees of the service activities. The altered objective function of the executing reliability is shown in Formula (23), where $EXRE_S$ denotes the overall executing reliability of all selected activities. As a result, 13 Pareto optimal solutions were obtained, which were recorded as set C.

$$\max EXRE_S = \sum_{j=1}^{\delta} y_j \cdot exre_j. \tag{23}$$

The executing reliabilities of sa_9 ($exre_9$) in different solutions with set A and set C are shown in Figure 10. The values of $exre_9$ in set A are all higher than those in set C. The values of $exre_9$ in solution A3 and C11 are the highest in set A and set C, respectively. Service activity compositions in solution A3 and C11 are the same. Comparing and analyzing these two solutions shows why maximizing $AVRE_S$, as one of the objective functions, could facilitate more beneficial solutions, in which service activities with higher importance degrees have higher executing reliabilities.

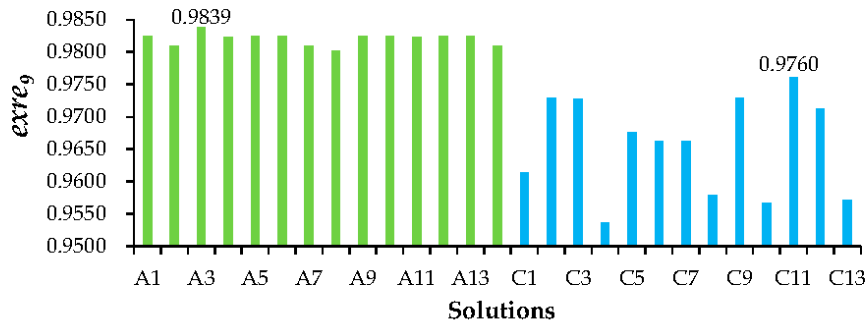


Figure 10. $exre_9$ in different solutions in set A and set C.

The Gantt chart of selected service activity nodes in solution A3 and C11 is depicted in Figure 11. From the perspective of the service time interval, sa_9 overlaps with sa_{19} and sa_{22}, sa_{19} overlaps with sa_8, sa_8 overlaps with other nodes. Situations of service time interval overlap between service activities are intricate. Consequently, resource allocation of one activity might influence another one and indirectly impact other activities.

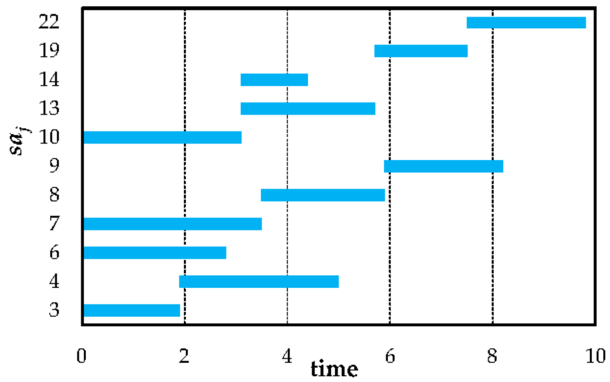


Figure 11. Gantt chart of selected service activity nodes.

The types and quantities of resources demanded by sa_8, sa_9, sa_{19}, and sa_{22} are shown in Table 4. The actual situation of resource allocation about the kr related to sa_9 in solution A3 and C11 is illustrated in Table 5. w_{ij}^{RS} on the edges that connect sa_j and r_i (all pr_i and er_i that are related to kr_{124}, kr_{126}, kr_{134}, and kr_{136}) are detailed in Figures 12 and 13.

Table 4. Resource quantity demands of *sa*

	kr_{122}	kr_{124}	kr_{125}	kr_{126}	kr_{132}	kr_{134}	kr_{136}
sa_9	0	1	0	1	0	1	1
sa_8	1	1	1	1	1	1	0
sa_{19}	0	2	0	1	0	1	0
sa_{22}	0	1	0	2	0	0	1

Table 5. r_i that are allocated to sa_8, sa_9, sa_{19}, and sa_{22}.

	sa_8			sa_9				sa_{19}				sa_{22}			
	kr_{124}	kr_{126}	kr_{134}	kr_{124}	kr_{126}	kr_{134}	kr_{136}	kr_{124}		kr_{126}	kr_{134}	kr_{124}	kr_{126}		kr_{136}
A3	pr_{31}	pr_{46}	er_{86}	pr_{28}	pr_{46}	er_{90}	er_{98}	pr_{26}	pr_{29}	pr_{44}	er_{87}	pr_{29}	pr_{44}	pr_{47}	er_{100}
C11	pr_{30}	pr_{43}	er_{90}	pr_{31}	pr_{47}	er_{90}	er_{97}	pr_{27}	pr_{32}	pr_{44}	er_{86}	pr_{25}	pr_{44}	pr_{46}	er_{98}

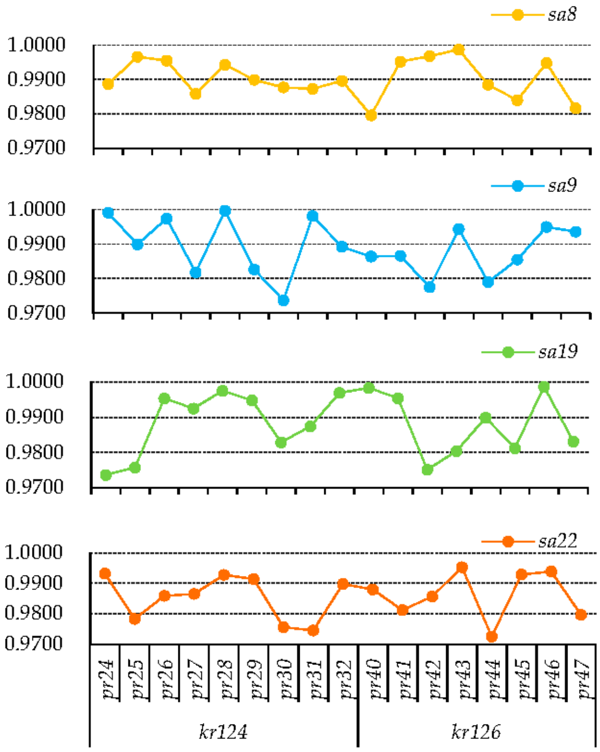


Figure 12. w_{ij}^{RS} on edges that connect pr_i to sa_8, sa_9, sa_{19}, and sa_{22}.

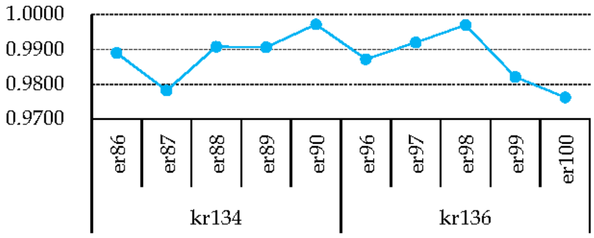


Figure 13. w_{ij}^{RS} on edges that connect er_i to all sa.

The decision process in multi-objective optimization is complicated. In solution C11, r_i with the highest w_{i9}^{RS} is not allocated to sa_9 in the process of overall executing reliability maximization. In solution A3, pr_{28}, pr_{46}, er_{90}, and er_{98} are all allocated to sa_9. w_{i9}^{RS} of these four resources is the highest among the same type of resources. In turn, the $exre_9$ in solution A3 is higher than solution C11. For the remaining three sa, for whom the importance degrees are not conspicuously higher than each other, the resource allocation is not only based on w_{ij}^{RS}, but also on the cost (as shown in Figure 9). By including the importance degree of sa_j with the objective function of executing reliability as weight, it enables more important sa_j to have higher executing reliabilities, by allocating r_i with the highest w_{ij}^{RS} to this sa_j. If the importance degree of sa_j is significantly higher, the impact on the overall executing reliability and executing cost will be tolerable.

6. Discussion

As an imperative part of PSS design, PSS configuration is usually studied in a predefined level of module granularity [21,22]. The service modules in existing studies of PSS [23,34] are essentially whole service flows, which can be decomposed into service activities. In the studies of service activity composition [42,45], the impact of resource allocation on service activity is not considered during the service activities selection. This paper studied the PSS configuration in a finer granularity level to inspire the final configuration's optimization. By combining service activities selection and resource allocation, the proposed approach enriched the diversity of the optimal solution set by retaining more non-dominated solutions. These solutions might be eliminated in advance within the process of service activities selection, when service activities selection and resource allocation are divided into two independent processes in the PSS configuration optimization. Furthermore, the above-mentioned existing studies highlight the multi-objective optimization of the overall solution. The proposed approach improves the executing reliabilities of service activities with a higher importance degree, by allocating resources with higher reliabilities to them. In terms of modeling, a multilayer network model can provide a better description of relationships among three types of elements (product parts, service activities, and resources), compared to an ontology-based model [43], structure tree model [44], and Petri net model [45,46]. The mathematical characteristics of multilayer networks are useful for analyzing the importance degree of product parts and service activities. The sparse matrixes of edges in multilayer networks are applicable as the input of the proposed optimization algorithm.

This paper focused on product-oriented PSS. Adding services to traditional products is used in product-oriented PSS as an approach to reach sustainability [66]. The ultimate purpose of the proposed approach is to achieve PSS configuration optimization. Even if the purpose of sustainability is not directly mentioned, the PSS configuration optimization is still related to achieving sustainability. Optimization of the executing time means a reduction in energy consumption and decreased work time of workers and equipment. The reduced resource utilization can lead to the concept of sustainability in PSS [5]. The optimization of executing costs is an economic benefit. The optimization of executing reliability can also be regarded as a guarantee of the optimization of executing time and cost. Furthermore, one goal of the proposed multi-objective optimization is to improve customer satisfaction. Economic perspective and customer satisfaction are also important measurement indicators of sustainability [9].

7. Conclusions

PSS enables manufacturing companies to provide customers with solutions by integrating preferable product modules and service modules. How to optimize PSS configuration in a fine-grained perspective when modules are confirmed represents an important issue for PSS management. A PSS multilayer network-based method combining service activities selection and resource allocation was developed to obtain optimal PSS solutions. Product parts, services, and resources were mapped as different layers in a PSS multilayer network. In particular, the mapping rules from logical relationships among service activities to edges among nodes were developed to model service layers appropriately. Service activities selection and resource allocation were combined to evaluate the objective functions of solutions systematically and precisely. The importance degree of a service activity was proposed to ascertain the priority of service activities in resource allocation. The advantages of the proposed method are as follows:

1. The impact of resources on the cost and reliability of service activities is thoroughly considered, which can affect the PSS configuration optimization equally with service activities. A combination of service activities selection and resource allocation can make the calculation of optimizing indexes closer to the actual implementation;

2. The PSS multilayer network model, which accommodates heterogeneous nodes and relationships without distortion, can be transformed into matrices to present information. It helps designers understand the construction of the PSS and build optimization functions easily;
3. Cross layer importance degree can sort the service activities based on the importance of the product parts and the objects of service activities.

Although the approach optimizes PSS configuration in a more fine-grained perspective, it also has some limitations. The combination of service activities selection and resource allocation leads to more complexity within the calculation. The optimization is based on a given product in a situation where customers have explicit targets on product module selection. Further work would consider improving the algorithm to shorten the calculation's response time. Besides, the multilayer network-based modeling method might be extended to the design of use-oriented PSS and result-oriented PSS to satisfy demands for different types of PSS.

Author Contributions: Conceptualization, Z.Z. and D.X.; methodology, Z.Z. and D.X.; software, D.X.; validation, Z.Z., D.X., E.O. and H.C.; formal analysis, D.X.; investigation, D.X.; resources, Z.Z.; data curation, D.X.; writing—original draft, D.X.; writing—review and editing, Z.Z., D.X., E.O. and H.C.; supervision, Z.Z.; project administration, Z.Z.; funding acquisition, Z.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China, grant number 51205242, and the Shanghai Science and Technology Innovation Action Plan, grant number 16111106402.

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

Appendix A

$$\min T_S = \sum_{j=1}^{\delta} y_j \cdot T_j \tag{A1}$$

$$\min C_S = \sum_{j=1}^{\delta} y_j \cdot \left(C_j + \sum_{i=1}^{\alpha+\beta} x_{ji} \cdot c_i \cdot T_j \right) \tag{A2}$$

$$\max AVRE_S = \frac{1}{\sum_{j=1}^{\delta} y_j \cdot D_j} \cdot \sum_{j=1}^{\delta} \left(y_j \cdot D_j \cdot \prod_{i=1}^{\alpha+\beta} z_{ji} \right) \tag{A3}$$

$$y_j \cdot \left(\sum_{\varphi=\delta+1}^{\delta+\gamma} e_{j\varphi}^S \cdot u_\varphi \right) = y_j, j = 1, 2, \cdots, \delta \tag{A4}$$

$$\sum_{j=1}^{\delta} y_j \cdot e_{\varphi j}^S \cdot w_{\varphi j}^S = u_\varphi, \varphi = \delta + 1, \cdots, \delta + \gamma \tag{A5}$$

$$\sum_{j=1}^{\delta} y_j \cdot e_{j0}^S \cdot w_{j0}^S = \sigma \tag{A6}$$

$$\left[\left(\sum_{j=1}^{\delta} y_j \cdot e_{ij}^S \cdot w_{ij}^S \right) + e_{i0}^S \cdot w_{i0}^S \right] \cdot y_i = y_i \cdot \left[\left(\sum_{\varphi=\delta+1}^{\delta+\gamma} e_{\varphi i}^S \cdot w_{\varphi i}^S \right) + \left(\sum_{j=1}^{\delta} y_j \cdot e_{ji}^S \cdot w_{ji}^S \right) \right], i = 1, 2, \cdots, \delta \tag{A7}$$

$$y_j \cdot e_{j\varphi}^S \cdot ft_j \le u_\varphi \cdot ddl\_t_\varphi, j = 1, 2, \cdots, \delta; \varphi = \delta + 1, \cdots, \delta + \gamma \tag{A8}$$

$$\sum_{i=1}^{\alpha+\beta} e_{hi}^R \cdot x_{ji} = y_j \cdot e_{jh}^{SR} \cdot w_{jh}^{SR}, j = 1, 2, \cdots, \delta; h = \alpha + \beta + 1, \cdots, \alpha + \beta + b \tag{A9}$$

$$\sum_{i=1}^{\alpha+\beta} \sum_{j=1}^{\delta} \sum_{h=1}^{\delta} \left(x_{ji} \cdot x_{hi} \cdot ol_{jh} \right) = 0, j \ne h \tag{A10}$$

$$ol_{jh} = \begin{cases} 1, & \left(bt_j, ft_j \right) \cap \left(bt_h, ft_h \right) \ne \varnothing \\ 0, & \left(bt_j, ft_j \right) \cap \left(bt_h, ft_h \right) = \varnothing \end{cases}, j \ne h \tag{A11}$$

$$z_{ji} = \begin{cases} w_{ij}^{RS}, & x_{ji} = 1 \\ 1, & x_{ji} = 0 \end{cases} \tag{A12}$$

$$x_{ji} = ax_{ji} \cdot e_{ij}^{RS} \cdot y_j \tag{A13}$$

$$ax_{ji} \cdot \left(ax_{ji} - 1\right) = 0 \tag{A14}$$

$$y_j \cdot \left(y_j - 1\right) = 0 \tag{A15}$$

References

1. Meier, H.; Völker, O.; Funke, B. Industrial product-service systems (IPS 2). *Int. J. Adv. Manuf. Technol.* **2011**, *52*, 1175–1191. [CrossRef]
2. Lindahl, M.; Sundin, E.; Sakao, T. Environmental and economic benefits of Integrated Product Service Offerings quantified with real business cases. *J. Clean. Prod.* **2014**, *64*, 288–296. [CrossRef]
3. Mont, O.K. Clarifying the concept of product–service system. *J. Clean. Prod.* **2002**, *10*, 237–245. [CrossRef]
4. Tukker, A.; Tischner, U. Product-services as a research field: Past, present and future. Reflections from a decade of research. *J. Clean. Prod.* **2006**, *14*, 1552–1556. [CrossRef]
5. Baines, T.S.; Lightfoot, H.W.; Evans, S.; Neely, A.; Greenough, R.; Peppard, J.; Roy, R.; Shehab, E.; Braganza, A.; Tiwari, A. State-of-the-art in product-service systems. *Proc. Inst. Mech. Eng. Part B* **2007**, *221*, 1543–1552. [CrossRef]
6. Tukker, A. Product services for a resource-efficient and circular economy—A review. *J. Clean. Prod.* **2015**, *97*, 76–91. [CrossRef]
7. Rapaccini, M. Pricing strategies of service offerings in manufacturing companies: A literature review and empirical investigation. *Prod. Plan. Control.* **2015**, *26*, 1247–1263. [CrossRef]
8. Tukker, A. Eight types of product–service system: Eight ways to sustainability? Experiences from SusProNet. *Bus. Strategy Environ.* **2004**, *13*, 246–260. [CrossRef]
9. Lee, S.; Geum, Y.; Lee, H.; Park, Y. Dynamic and multidimensional measurement of product-service system (PSS) sustainability: A triple bottom line (TBL)-based system dynamics approach. *J. Clean. Prod.* **2012**, *32*, 173–182. [CrossRef]
10. Sakao, T. Engineering PSS (Product/Service Systems) toward sustainability: Review of research. In *Handbook of Sustainable Engineering*; Kauffman, J., Lee, K., Eds.; Springer: Dordrecht, The Netherlands, 2013; pp. 597–613.
11. Pascual, R.; Siña, M.; Santelices, G.; Román, M.; López Droguett, E. Optimal channel coordination in use-based product-service system contracts. *Int. J. Prod. Res.* **2017**, *55*, 6946–6956. [CrossRef]
12. Benedettini, O.; Neely, A.; Swink, M. Why do servitized firms fail? A risk-based explanation. *Int. J. Oper. Prod. Manag.* **2015**, *35*, 946–979. [CrossRef]
13. Gebauer, H.; Fleisch, E. An investigation of the relationship between behavioral processes, motivation, investments in the service business and service revenue. *Ind. Mark. Manag.* **2007**, *36*, 337–348. [CrossRef]
14. Neely, A. Exploring the financial consequences of the servitization of manufacturing. *Oper. Manag. Res.* **2008**, *1*, 103–118. [CrossRef]
15. Cavalieri, S.; Pezzotta, G. Product–Service Systems Engineering: State of the art and research challenges. *Comput. Ind.* **2012**, *63*, 278–288. [CrossRef]
16. Matschewsky, J.; Kambanou, M.L.; Sakao, T. Designing and providing integrated product-service systems–challenges, opportunities and solutions resulting from prescriptive approaches in two industrial companies. *Int. J. Prod. Res.* **2018**, *56*, 2150–2168. [CrossRef]
17. Hussain, R.; Lockett, H.; Vasantha, G.V.A. A framework to inform PSS Conceptual Design by using system-in-use data. *Comput. Ind.* **2012**, *63*, 319–327. [CrossRef]
18. Song, W.; Sakao, T. A customization-oriented framework for design of sustainable product/service system. *J. Clean. Prod.* **2017**, *140*, 1672–1685. [CrossRef]
19. Mourtzis, D.; Fotia, S.; Vlachou, E.; Koutoupes, A. A Lean PSS design and evaluation framework supported by KPI monitoring and context sensitivity tools. *Int. J. Adv. Manuf. Technol.* **2018**, *94*, 1623–1637. [CrossRef]
20. Marilungo, E.; Peruzzini, M.; Germani, M. Review of Product-Service System Design Methods. In *Product Lifecycle Management in the Era of Internet of Things, Proceedings of the IFIP International Conference on Product Lifecycle Management, Doha, Qatar, 19–21 October 2015*; Springer: Cham, Switzerland, 2016; pp. 271–279.
21. Geng, X.; Xu, S.; Ye, C. Module configuration approach for product service system design driven by customer requirements. *Comput. Integr. Manuf. Syst.* **2016**, *22*, 55–61.

22. Sun, J.; Chai, N.; Pi, G.; Zhang, Z.; Fan, B. Modularization of product service system based on functional requirement. *Procedia CIRP* **2017**, *64*, 301–305. [CrossRef]
23. Zhang, Z.; Shang, Y.; Sun, J.; Fan, B. Product services system configuration optimization based on multi-objective discrete cuckoo search algorithm. *Comput. Integr. Manuf. Syst.* **2017**, *23*, 1774–1786.
24. Shen, J.; Erkoyuncu, J.A.; Roy, R.; Wu, B. A framework for cost evaluation in product service system configuration. *Int. J. Prod. Res.* **2017**, *55*, 6120–6144. [CrossRef]
25. Zhang, Z.; Xu, D.; Ostrosi, E.; Yu, L.; Fan, B. A systematic decision-making method for evaluating design alternatives of product service system based on variable precision rough set. *J. Intell. Manuf.* **2019**, *30*, 1895–1909. [CrossRef]
26. Li, H.; Ji, Y.; Gu, X.; Qi, G.; Tang, R. Module partition process model and method of integrated service product. *Comput. Ind.* **2012**, *63*, 298–308. [CrossRef]
27. Long, H.; Wang, L.; Shen, J.; Wu, M.; Jiang, Z. Product service system configuration based on support vector machine considering customer perception. *Int. J. Prod. Res.* **2013**, *51*, 5450–5468. [CrossRef]
28. Wu, B.; Fan, S.; Yu, A.J.; Xi, L. Configuration and operation architecture for dynamic cellular manufacturing product–service system. *J. Clean. Prod.* **2016**, *131*, 716–727. [CrossRef]
29. Long, H.J.; Wang, L.Y. A rough set based approach to knowledge acquisition for product service system configuration. *Appl. Mech. Mater.* **2012**, *220–223*, 2534–2539. [CrossRef]
30. Zhang, Z.; Chai, N.; Ostrosi, E.; Shang, Y. Extraction of association rules in the schematic design of product service system based on Pareto-MODGDFA. *Comput. Ind. Eng.* **2019**, *129*, 392–403. [CrossRef]
31. Liu, Y.; Zhang, Z.F.; Yao, D.; Chu, X.N. Extraction of Product Service System Configuration Rules Based on Multi-objective DPSO Algorithm. *J. Shanghai Jiaotong Univ.* **2015**, *49*, 1123–1130.
32. Shen, J.; Wu, B.; Yu, L. Personalized configuration rules extraction in product service systems by using Local Cluster Neural Network. *Ind. Manag. Data Syst.* **2015**, *115*, 1529–1546. [CrossRef]
33. Long, H.; Wang, L.; Zhao, S.; Jiang, Z. An approach to rule extraction for product service system configuration that considers customer perception. *Int. J. Prod. Res.* **2016**, *54*, 5337–5360. [CrossRef]
34. Li, H.; Ji, Y.; Chen, L.; Jiao, R.J. Bi-level coordinated configuration optimization for product-service system modular design. *IEEE Trans. Syst. Man Cybern. Syst.* **2015**, *47*, 537–554. [CrossRef]
35. Yan, H.; Zhong, W. Modular decomposition model of complex product system and its application. *J. Beijing Univ. Aeronaut. Astronaut.* **2017**, *43*, 654–659.
36. Pezzotta, G.; Pirola, F.; Pinto, R.; Akasaka, F.; Shimomura, Y. A Service Engineering framework to design and assess an integrated product-service. *Mechatronics* **2015**, *31*, 169–179. [CrossRef]
37. Shimomura, Y.; Nemoto, Y.; Ishii, T.; Nakamura, T. A method for identifying customer orientations and requirements for product–service systems design. *Int. J. Prod. Res.* **2018**, *56*, 2585–2595. [CrossRef]
38. Szwejczewski, M.; Goffin, K.; Anagnostopoulos, Z. Product service systems, after-sales service and new product development. *Int. J. Prod. Res.* **2015**, *53*, 5334–5353. [CrossRef]
39. Fargnoli, M.; Haber, N.; Sakao, T. PSS modularisation: A customer-driven integrated approach. *Int. J. Prod. Res.* **2018**, *57*, 1–17. [CrossRef]
40. Mourtzis, D.; Fotia, S.; Boli, N.; Vlachou, E. An approach for the modelling and quantification of PSS customisation. *Int. J. Prod. Res.* **2018**, *56*, 1137–1153. [CrossRef]
41. Pezzotta, G.; Pirola, F.; Rondini, A.; Pinto, R.; Ouertani, M.-Z. Towards a methodology to engineer industrial product-service system–Evidence from power and automation industry. *CIRP J. Manuf. Sci. Technol.* **2016**, *15*, 19–32. [CrossRef]
42. Zhang, M.-W.; Zhang, B.; Liu, Y.; Na, J.; Zhu, Z.-L. Web service composition based on QoS rules. *J. Comput. Sci. Technol.* **2010**, *25*, 1143–1156. [CrossRef]
43. Zhu, H.; Gao, J.; Li, D.; Tang, D. A Web-based Product Service System for aerospace maintenance, repair and overhaul services. *Comput. Ind.* **2012**, *63*, 338–348. [CrossRef]
44. Cao, J.; Wang, J.; Zhao, H.; Sun, X. A service process optimization method based on model refinement. *J. Supercomput.* **2013**, *63*, 72–88. [CrossRef]
45. Wang, Y.; Lin, C.; Ungsunan, P.D.; Huang, X. Modeling and survivability analysis of service composition using Stochastic Petri Nets. *J. Supercomput.* **2011**, *56*, 79–105. [CrossRef]
46. Hu, Q.; Du, Y.; Yu, S. Service net algebra based on logic Petri nets. *Inf. Sci.* **2014**, *268*, 271–289. [CrossRef]
47. Johnson, J. *Hypernetworks in the Science of Complex Systems*; Imperial College Press: London, UK, 2013.

48. Johnson, J.H. Hypernetworks: Multidimensional relationships in multilevel systems. *Eur. Phys. J. Spec. Top.* **2016**, *225*, 1037–1052. [CrossRef]
49. Johnson, J.H. Hypernetworks for reconstructing the dynamics of multilevel systems. In Proceedings of the European Conference on Complex Systems, Oxford, UK, 22–29 September 2006.
50. Kivelä, M.; Arenas, A.; Barthelemy, M.; Gleeson, J.P.; Moreno, Y.; Porter, M.A. Multilayer networks. *J. Complex Netw.* **2014**, *2*, 203–271. [CrossRef]
51. De Domenico, M.; Solé-Ribalta, A.; Cozzo, E.; Kivelä, M.; Moreno, Y.; Porter, M.A.; Gómez, S.; Arenas, A. Mathematical formulation of multilayer networks. *Phys. Rev. X* **2013**, *3*, 041022. [CrossRef]
52. De Domenico, M.; Granell, C.; Porter, M.A.; Arenas, A. The physics of spreading processes in multilayer networks. *Nat. Phys.* **2016**, *12*, 901–906. [CrossRef]
53. Omodei, E.; De Domenico, M.; Arenas, A. Evaluating the impact of interdisciplinary research: A multilayer network approach. *Netw. Sci.* **2017**, *5*, 235–246. [CrossRef]
54. Pasqual, M.C.; de Weck, O.L. Multilayer network model for analysis and management of change propagation. *Res. Eng. Des.* **2012**, *23*, 305–328. [CrossRef]
55. Leng, J.; Jiang, P. Dynamic scheduling in RFID-driven discrete manufacturing system by using multi-layer network metrics as heuristic information. *J. Intell. Manuf.* **2019**, *30*, 979–994. [CrossRef]
56. Eppinger, S.D.; Browning, T.R. *Design Structure Matrix Methods and Applications*; MIT Press: Cambridge, MA, USA, 2012.
57. Tomiyama, T. Service engineering to intensify service contents in product life cycles. In Proceedings of the Second International Symposium on Environmentally Conscious Design and Inverse Manufacturing, Tokyo, Japan, 11–15 December 2001; pp. 613–618.
58. Cheng, Y.; Zhao, D.; Tao, F.; Zhang, L.; Liu, Y. Complex networks based manufacturing service and task management in cloud environment. In Proceedings of the 2015 IEEE 10th Conference on Industrial Electronics and Applications, Auckland, New Zealand, 15–17 June 2015; pp. 242–247.
59. Chen, D.; Lü, L.; Shang, M.-S.; Zhang, Y.-C.; Zhou, T. Identifying influential nodes in complex networks. *Phys. A* **2012**, *391*, 1777–1787. [CrossRef]
60. Lü, L.; Zhang, Y.-C.; Yeung, C.H.; Zhou, T. Leaders in social networks, the delicious case. *PLoS ONE* **2011**, *6*, e21202. [CrossRef]
61. Gu, Y.; Zhu, Z. Node Ranking in Complex Networks based on LeaderRank and Modes Similarity. *J. Univ. Electron. Sci. Technol. China* **2017**, *46*, 441–448.
62. Brin, S.; Page, L. The anatomy of a large-scale hypertextual web search engine. *Comput. Netw. ISDN Syst.* **1998**, *30*, 107–117. [CrossRef]
63. Restrepo, J.G.; Ott, E.; Hunt, B.R. Characterizing the dynamical importance of network nodes and links. *Phys. Rev. Lett.* **2006**, *97*, 094102. [CrossRef]
64. Li, Q.; Zhou, T.; Lü, L.; Chen, D. Identifying influential spreaders by weighted LeaderRank. *Phys. A* **2014**, *404*, 47–55. [CrossRef]
65. Dudas, C.; Ng, A.H.; Pehrsson, L.; Boström, H. Integration of data mining and multi-objective optimisation for decision support in production systems development. *Int. J. Comput. Integr. Manuf.* **2014**, *27*, 824–839. [CrossRef]
66. Manzini, E.; Vezzoli, C.; Clark, G. Product-service systems: Using an existing concept as a new approach to sustainability. *J. Des. Res.* **2001**, *1*, 27–40. [CrossRef]