The International Journal of Advanced Manufacturing Technology (2019) 105:3815–3828
https://doi.org/10.1007/s00170-019-03802-2

ORIGINAL ARTICLE

Managing complexity of assembly with modularity: a cost and benefit analysis

Shraga Shoval[1] · Mahmoud Efatmaneshnik[2]

Received: 1 August 2018 / Accepted: 17 April 2019 / Published online: 21 May 2019

Abstract

Industry 4.0 is characterized by a modular structure of the production process that consists of cyber-physical systems. These cyber-physical systems provide interoperability, information transparency, and decentralization of decisions. The modular structure, according to Industry 4.0 principle, creates intelligent networks of machines, work pieces, and systems that can predict failures, self-organize themselves, and react to unexpected events. In this paper, we consider the complexity of assembly processes and propose modular structures for assembly processes based on probabilistic formulation. Despite the reliability and precisions that the use of cyber-physical systems such as robotics and automation in assembly processes have introduced, and because of the increasing complexity, there is a need for probabilistic process characterization models for smart assembly planning purposes. First, a new framework for assembly complexity measurement based on processes' probabilistic and Markovian characters is suggested. Then, two effects of modularization, namely stabilization of components by boundary creation and application modular interfaces, are analyzed. For each case, a probabilistic formulation for assembly formation and analysis is presented. The effect of task sequencing and component modularization on assembly time and cost is considered simultaneously by the Bayesian formulation of the assembly problem. Several heuristics are derived from simulation examples, and the modularization cost is studied through utilization of design structure matrix.

Keywords Modularity · Sequencing · Assembly · System complexity · Probabilistic modeling

1 Introduction

Cyber-physical system (CPS) consists of physical assets and computational capabilities that are integrated and interconnected. The availability and affordability of field devices, information systems, and computer networks drive industries to implement new methodologies, resulting in more intelligent, resilient, and self-adaptable systems. These types of systems, known as Industry 4.0 systems, offer significant economic potentials by acquiring accurate and reliable data from the production plants, converting this data into useful information, analyzing the information, optimizing operational decision, and applying the corrective and preventative decisions. The challenge as well as the key to the success of Industry 4.0 are highly modular structures for multivendor interoperability facilitation [1]. In Industry 4.0, the traditional production hierarchy is replaced by a decentralized self-organization scheme [2]. Components and procedures with local control intelligence communicate to other components and procedures through the system's network to self-organize themselves within the production network. In this way, production lines become flexible and modular, allowing an easy plug-and-play integration or replacement of entities (components and/or procedures) [3]. The increased demand for personalized products, reduced life cycle time, and higher competitive markets results in more complexity of the organizational structures and procedures, including the assembly processes. However, the complexity introduced by Industry 4.0 technologies demands new approaches to manage modular design and design for assembly techniques.

Industry 4.0 is heralding an era of personalized production and assembly [4], which demands and is motivated by

✉ Mahmoud Efatmaneshnik
m.efatmaneshnik@adfa.edu.au

Shraga Shoval
shraga@ariel.ac.il

[1] Department of Industrial Engineering and Management, Ariel University, 47000 Ariel, Israel

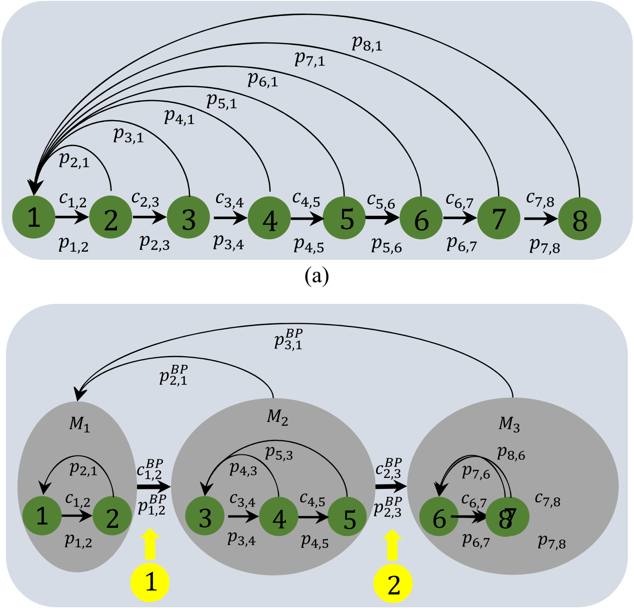
[2] School of Engineering and Information Technology, UNSW Canberra at ADFA, Campbell, ACT, Australia

integration of online/real-time sequence optimization and optimal assembly formation tools. This information becomes readily available to the assembly workers, for example though augmented reality headsets [4], and to assembly machines for evolvable readjustment [5]. The optimization problem applies to both assembly time, through balancing problem [6, 7], and assembly process cost. In this paper, we introduce a probabilistic formulation for assembly optimization problem and explore the sensitivities of an assembly process to the constraints imposed by modular design. This method can be used with online optimization of assembly processes through simultaneous sequencing and module formation.

1.1 Problem definition

In the assembly literature, the cost or the number of defects is typically taken as the measure of assembly process efficiency. In this paper, we turn our attention to the cost of the assembly process (or the time of the assembly process) as a result of imperfect process efficiencies. The paper investigates the effect of modularity in the assembly sequence planning (ASP) for heterogeneous complex assembly processes where each operation (task) has a unique probability for success and a unique cost as shown Fig. 1a. Here, $c_{i,\ i+1}$ and $p_{i,\ i+1}$ are the cost and the probability for a successful completion of assembly task between elements i and $i+1$, respectively. We analyze the effect of buffering the assembly process by introducing break points as shown in Fig. 1b, given the heterogeneous probabilities and costs, and the mutual dependencies between all tasks, and in view of a principal assumption about the complexity of assembly, a failure in any task requires restarting the assembly process from the first task. The goal of these break points is to interrupt the mutual dependencies in case of failures such that a failure of a specific assembly task affects previous operations only to the last break point. These break points can take place as operational buffers (e.g., mechanical couplers, electronic diodes), as assembly stabilizers (e.g., additional fixtures and connectors), or as modules and subassemblies that can be executed independently. According to this structure, all operations between two break points construct a cluster, and any pair of clusters is connected at a break point. Each break point is defined by two values ($c^{BP}_{M_i,M_j}$ and $p^{BP}_{M_i,M_j}$), representing the cost and probability of assembling the two modules (i and j), respectively. Although each break point entails additional cost to the entire assembly process, we anticipate that the expected process cost and duration are improved by both optimal modularization and optimal sequencing of tasks within each module. Also, Fig. 2 shows a scenario where three workstations assemble the three modules and a forth workstation assembles the modules together. The total assembly time in this case is the sum of the longest module assembly time (for this example, workstation 2) and the module integration time in station 4.

In this work, we consider task sequencing (optimization) and (optimal) modularization simultaneously. Figure 3 shows three modular architectures of the same system: in Fig. 3a, b, the sequences are identical and the modularization is different, while in Fig. 3c, both the sequence and the modularization are altered. We define a modular structure by a modularization



(b)

Fig. 1 **a** Serial assembly sequence of eight components with seven fully mutually dependent assembly tasks, each with a different cost and reliability. **b** Two break points replace two initial assembly tasks to interrupt the mutual dependencies between the tasks

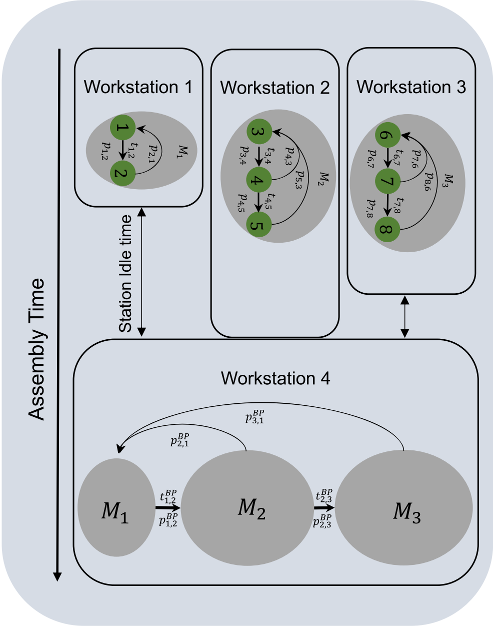


Fig. 2 A hypothetical assembly time scenario for modularization in Fig. 1b

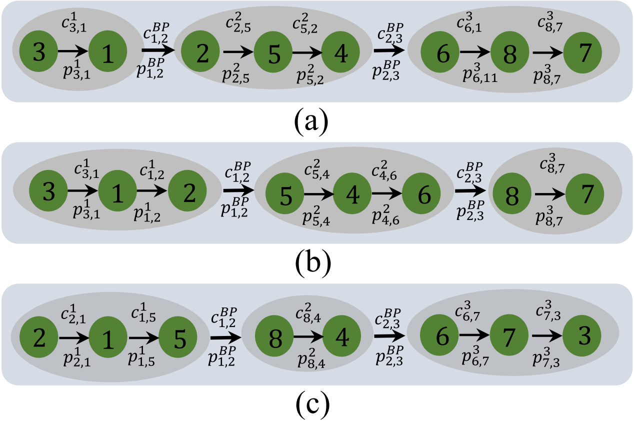


Fig. 3 **a**–**c** Different modular structures with different sequences of the same system resulting in different expected costs

vector (MV) that characterizes a modular architecture by the number of modules in the entire process and by the number of elements within each module for a given sequence of the tasks (H_k). For example, the MV in Fig. 3a is [2, 3] (two elements in M_1, three elements in M_2, and three elements in M_3) for the sequence [3 → 1 → 2 → 5 → 4 → 6 → 8 → 7], the MV in Fig. 3b is [2, 3] for the same sequence, and the MV in Fig. 3c is [2, 3] for the sequence [2 → 1 → 5 → 8 → 4 → 6 → 7 → 3]. The aim here is to find optimum modularization and sequencing for minimum time (optimally balanced line [6]) and cost.

More specifically, there are several questions regarding the implementation of the break points that the paper tries to answer:

1. How many points (modules) the optimal cost and time values give?
2. Where to insert them?
3. Should there be a change of assembly sequence within each module? Is there an optimal assembly sequencing for the new modular structure with respect to both assembly cost and time?
4. What sort of trade-offs exists between assembly time and cost for each modularization and sequence?

We study the above problems in a very general form by considering the following assumptions:

1. There are enough working stations to accommodate any modularization.
2. There are no precedence constraints for assembly tasks. This is not a limiting assumption, and this actually increases the computational complexity of the optimization process relative to when precedence constraints are present.
3. The probability for successfully conducting an assembly task and its cost are constant, regardless of the number of attempts and repeats. In reality, repeating a task might increase the probability of its success because of learning.

In Section 2, a literature review and background to the assembly complexity, and some initial developments to the assembly model, are provided. We show how task sequencing affects complex assemblies with heterogeneous success probabilities. We then examine, in Section 3, the effects of employing break points in a given assembly sequence. In Section 4, we examine the simultaneous effect of employing break points and changes in the assembly sequencing. The paper concludes that, for cost optimality, task sequencing must be performed subsequent to task decomposition; otherwise, the cost-saving effects of task sequencing can be eradicated by inappropriate decomposition.

2 Background

Traditionally, an assembly line is a sequential manufacturing setup in which components are assembled in several workstations (local or external) which are connected by a transportation network [8]. The transportation network can be rigid (e.g., conveyors) or flexible (e.g., autonomous guided vehicles (AGVs)). Each station in the line performs one or several assembly operations using humans, machines, tools, or robots. The output of the assembly line is a complete unit that is assembled according to the product structure or “bill of materials” (BOM) that lists the components, sub-assemblies and parts, and their relationships. The assembly process has a major impact on the product cost and quality. As assembly is often the last process in the manufacturing chain, detection and correction of failures in this stage is an important factor for the product quality. Design for assembly (DFA), originally proposed by Boothroyd [9], is a product design methodology that aims at reducing costs and the probability for assembly failures. Reducing the number of components (and therefore, the number of assembly tasks), adding grasping and orientation tools, and considering assembly directionality assist in reducing assembly cost and increasing the reliability. Numerous enterprises use the DFA methodology and report large savings [10–12]. Many models, technologies, and methodologies have been developed since Boothroyd’s introduction of the DFA for improving the assembly process and reducing the costs [13–17].

Statistical quality control (SQC) is a common tool in industry for process monitoring and improvement, which is based on sampling products from the manufacturing line, observing variations, and applying corrections [18]. The Motorola 6σ (Six Sigma) method [19] is another common tool used for process improvement. It identifies and removes the causes for failures during the manufacturing process using a clear sequence of actions with specific value targets. SQC and 6σ methods in assembly processes assume all defects are the results of normal distribution of variations that can be observed and corrected. However, many failures during the assembly process can be defined only in terms of probabilistic occurrence due to different types of errors [20] such as missed assembly operation, processing errors, tools and machines' setting-up errors, missing parts, wrong parts, and tools and jigs' errors. Hinckley [21] presents the type of defects, the source for these defects, and metrics for assessing the relative importance of each source. He introduces the concept of assembly complexity factor and states that "complexity is the least understood source of defects in assembly processes because of the difficulty of defining relative measures of complexity".

The complexity of the assembly process can be expressed by quantitative measures (e.g., the number of components and the required assembly operations) and by qualitative measures (e.g., the level of components and assembly difficulty). While quantitative measures are explicit and unambiguous, the qualitative measures are subjective and multitudinous. The expected time for completion of a specific assembly operation is often used as an approximation for the assembly complexity. Hinckley [21] formulates complexity by the probability that an assembly process is successful according to

$$P_Y = \prod_{i=1}^{n} \left\{ C_k \left(t_i - t_0\right)^k \right\} \left(1 - d_i\right) \tag{1}$$

where the notations are as follows:

$P_Y$the probability that the entire assembly process is successful

$C_k$the level of the quality control of the assembly operations ($C_k > 0$)

$t_i$the expected assembly time of assembly operation i

$t_0$the expected assembly time of the benchmark assembly operation

$k$the sensitivity of assembly complexity to defects ($k > 1$)

$n$the number of assembly operations

$d_i$the probability that the ith operation is defected ($0 \le d_i \le 1$)

The expected times in Eq. (1) can be determined from direct measurements or by using one of the DFA methodologies. The probability for successful assembly as given by Eq. (1) expresses the effect of the relevant factors on the assembly complexity and provides a tool for comparing design alternatives. However, this complexity definition does not take into consideration the effect of the assembly operations task sequencing on P_Y. It also does not help in the identification and formation of the assembly batches.

Here, we propose a new definition for assembly complexity that is based on a network of assembly operations and their level of interdependencies that determine the amount of rework. According to this definition, assembly processes are classified by the effect that a failure of one assembly task may have on pervious completed tasks. In simple assemblies, a failure of one assembly task requires rework of that task alone and does not necessitate reworks of prior tasks. A failure of a task in a complex assembly process, on the other hand, requires either scrapping the semi-finished assembly or disassembling some of the components and reassembling them. According to this definition, the assembly complexity is determined according to the *mutual dependencies* between the assembly tasks. Given the availability and affordability of a wide range of sensors and quality assurance techniques, each task in the assembly process can be accurately monitored and registered independently. Analyzing the outcomes of the quality assurance of each task, the assembly process can be adjusted by either repeating the failed tasks or scraping the entire process and restarting it from the first task.

Figure 4 illustrates these dependencies as a state diagram in which the states represent the assembly tasks and the edges represent the transitions between these tasks. Each transition has a probability ($p_{i,\,j}$) associated with it. The probability $p_{i,\,i+1}$ reflects the probability for a successful completion of task i and continuing to the next task in the sequence ($i + 1$). $p_{i,\,i}$ is the probability of repeating the same task after a failure, and $p_{i,\,j}$ $\forall j < i$ is the probability of returning to task j after a failure in task i. In this case, we define a complexity index ($k_i = i - j$) that indicates the number of steps the assembly process needs to repeat in case of a failure in task i. For example, in Fig. 1, tasks 1 and 2 are simple as a failure requires repetition of the failed tasks only. However, tasks 3 and 4 are complex as a failure in one of them requires repetition of previously completed tasks. Notice that a task may have several complexity indices (e.g., task 4) as there may be probabilities for returning to different previous tasks according to the specific type of failure. According to this definition, the complexity of the entire assembly sequence (C_A) is given by

$$C_A = \sum_{i=1}^{n-1} k_i(\max) \tag{2}$$

where k_i(max) is the largest complexity index of task i.

Although most assembly processes are a mixture of simple and complex tasks, in this paper, we consider the worst-case scenario in which all tasks are mutually dependent and are therefore sensitive to failures in the subsequent steps, and each task has the highest possible assembly complexity index such that a failure in one task results in repeating all previous tasks. In particular, we consider the cost associated with the

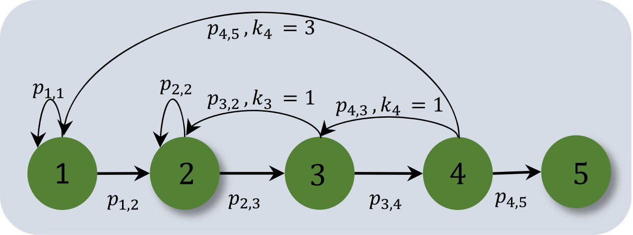


Fig. 4 State diagram of an assembly process

assembly process and the expected assembly time, given the probability for success of each assembly operation.

In the study of Efatmaneshnik and Ryan [22], we introduced a general formulation for tackling complex assemblies by modularization of the entire assembly process into subassemblies for homogeneous operations that are characterized by identical costs and probabilities for success of all assembly operations. In the study of Shoval et al. [23], we considered the effect of assembly sequencing on the expected assembly cost for simple and complex heterogeneous assembly processes. For simple heterogeneous serial assembly processes, we used the traveling salesperson problem (TSP) solver for determining the optimal assembly sequence that provides a lower bound for the assembly process. For complex heterogeneous assembly processes, we presented heuristics that claim that sequencing the more complex operations at the beginning of the process decreases the expected assembly cost. We also showed that when the probabilities for success of the assembly operations are similar, the more expensive operations should be deployed as late as possible (subject to the operational constraints) in order to reduce the total expected assembly cost.

A summary of the results presented in the study of Shoval et al. [23] that is relevant to the modularization of the assembly process is presented here for clarity. As mentioned, we consider a complex assembly system where tasks are fully mutually dependent, and therefore, a failure in the current assembly task requires rework of all previous tasks. Efatmaneshnik and Ryan [22] found that the expected cost of n assembly tasks for a *homogeneous* system (identical task reliabilities and costs) is

$$\hat{C}_n = \sum_{i=1}^{n} \frac{C}{P^i} = \frac{C}{1-P}\left(\frac{1}{P^n}-1\right) \tag{3}$$

where P and C are the probability for a successful completion and the cost of all the homogeneous assembly tasks, respectively. An important assumption in the model presented in the study of Efatmaneshnik and Ryan [22] is that the probability for success and the cost are constant, regardless of the number of attempts. That is, the probability and the cost of the first attempt are identical to all additional attempts due to the failures. Note that similar formulation can be driven for assembly time, in which case the cost is replaced by the homogenous time of completing the individual tasks. Because of this similarity, we pay closer attention to the assembly time in Section 3, where the assembly time of modular structures is considered. Since all probabilities and costs for all tasks are homogeneous, the expected cost of the entire process, given by Eq. (3) is not affected by the assembly sequence.

Next, we examine systems with heterogeneous probabilities and costs. Consider an assembly system that consists of $n + 1$ element that is assembled in n serial heterogeneous tasks, where the costs and reliability values of task i (connecting elements i and $i + 1$) are $c_{i,\ i+1}$ and $p_{i,\ i+1}$, respectively ($i = 1,..., n$). Assuming a system with no precedence constraints, then $H(n \times n!)$ represents all possible sequences for the n tasks' system, where each row is a specific assembly sequence. Let H_k ($k = 1,\ldots, n!$) be the kth row in H that represents a specific sequence, which consists of n tasks, each with a cost ($C_{H_k(i)}$) and probability for success ($P_{H_k(i)}$) ($i = 1,\ldots, n$). Notice that $C_{H_k(i)} = c_{H_k(i),H_k(i+1)}$ and $P_{H_k(i)} = p_{H_k(i),H_k(i+1)}$. The expected cost of the entire system is given by

$$\hat{\mathrm{C}}(H_k) = \sum_{i=1}^{n-1} \frac{c_{H_k(i),H_k(i+1)}}{\prod_{j=i}^{n} p_{H_k(i),H_k(i+1)}} \tag{4}$$

For proof of Eq. (4), see the study of Shoval et al. [23]. The complexity of determining the optimal assembly sequence using an exhaustive search algorithm is $O(n^2 \times n!)$. This problem is NP-complete and therefore intractable for assembly sizes greater than 20 tasks. However, given the precedence constraints, the number of possible valid sequences is lower, and often the problem can be solved using conventional optimization techniques (e.g., linear programming, graph optimization). Two useful heuristic based on Eq. (4) are that of Shoval et al. [23]:

1. Given equal costs for all assembly tasks, when less reliable tasks (lower probabilities of success) are executed earlier in the process, the expected cost of the whole assembly would be lower.
2. Given all assembly tasks are mutually dependent and have the same reliabilities for success, the minimum expected cost of a complex assembly process is given by a sequence that sorts the costs in ascending order.

3 Modularity in assembly planning

In this section, we examine the effect of modular boundaries on assembly cost and time. In the next section, we examine the effect of modular interfaces. Here, both effects are studied in combination with task sequencing. A complex assembly system, in which tasks are mutually dependent, was initially investigated by Simon [24]. Each time the process is disturbed, it has to be repeated from the first task. According to Simon [24], the solution to this complexity is the division of the process into several subassemblies such that a failure in one task affects only the tasks within that subassembly. Dividing assembly processes into subassemblies (modules) may require additional resources (machines, tools, space, etc.), and possibly new product architectures, as the connectivity between the components may need to be altered. These features should be considered in terms of their entire contribution to the system.

Two principal purposes can be envisaged for modularity in assembly [25]:

1. Reduce assembly cost

a. To stabilize the number of components in a module by creating boundaries for subassemblies that protect them from being harmed in the assembly rework process
b. Reduce the assembly complexity level by creating interfaces (standard or otherwise) between modules (subassemblies). These interfaces reduce the complexity of the process. By process complexity, we mean the measure introduced earlier in Eq. (2). Interfaces facilitate this by easing the plug-in process that have the ability to decouple the processes, thus removing the dependency between failure and success of subassembly processes. This process will be explained in Section 4

2. Reduce assembly time due to parallelism (further explained in this section)

Let us assume that a serial assembly process that consists of $n + 1$ elements and n tasks is redesigned such that m break points are added to the original assembly sequence, creating an $m + 1$ subassembly. In addition, the assembly sequence is replanned, subject to the assembly sequence constraints. The redesigned system is similar to the one shown in Fig. 2, but now, the assembly costs and probabilities for successful completion within each subassembly are marked by $c^{M_k}_{i,i+1}$ and $p^{M_k}_{i,i+1}$, respectively, representing costs and probabilities of assembling elements i and $i + 1$ within module M_k. These terms consider possible changes in the assembly sequences, as well as special features related to the modular structure. Furthermore, the modular structure includes different types of costs and probabilities: $c^{BP}_{M_i,M_{i+1}}$ is the cost of assembling modules i and $i + 1$, and similarly, $p^{BP}_{M_i,M_{i+1}}$ is the probability for success in assembling modules i and $i + 1$. Notice that assembling two components in a separate module may require additional fixtures, tools, or machines and also may change the probability for successful completion of that task. Therefore, in Fig. 4, $c^{BP}_{M_i,M_{i+1}} \neq c^{M_1}_{2,3} \neq c^{M_2}_{2,3}$ and, similarly, $p^{BP}_{M_i,M_{i+1}} \neq p^{M_1}_{2,3} \neq p^{M_2}_{2,3}$ as assembling modules M_1 and M_2 are different from assembling elements 2 and 3. In most cases, we anticipate that $c^{BP}_{M_i,M_{i+1}} \geq c^{M_i}_{j,j+1}$ and $p^{BP}_{M_i,M_{i+1}} \leq p^{M_i}_{j,j+1}$ as connecting two modules is typically more complex than connecting two elements.

Based on the previous notations, the total expected assembly cost in a modular system is given by

$$\hat{C}^M = \sum_{i=1}^{m-1} \frac{c^{BP}_{M_i,M_{i+1}}}{\prod_{j=i}^{m-1} p^{BP}_{M_j,M_{j+1}}} + \sum_{i=1}^{m} \sum_{j=1}^{n_i-1} \frac{c^{M_i}_{j,j+1}}{\prod_{k=j}^{n_i-1} p^{M_i}_{k,k+1}} \qquad (5)$$

where m is the number of modules, and $c^{BP}_{M_i,M_{i+1}}$ and $p^{BP}_{M_i,M_{i+1}}$ are the cost and reliability of the ith break point between the ith and the $(i + 1)$th modules, respectively (assembling modules M_i and M_{i+1}). In Eq. (5), $c^{M_i}_{j,j+1}$ and $p^{M_i}_{j,j+1}$ are respectively the cost and reliability of the jth assembly task inside module i, and n_i is the number of elements in the ith module. The first part in Eq. (5) is the expected cost of connecting all the modules, and the second part is the expected cost of assembling the elements within each module.

Now, assume $t^{M_i}_{j,j+1}$ is the time associated with the task of joining j and $j + 1$ components in the ith module, and $t^{BP}_{M_i,M_{i+1}}$ is the time associated with assembling modules M_i and M_{i+1}. In a modular setting, the assembly of modules can be performed in parallel. To fully utilize the parallelism, at least m working stations are required. Then, the expected minimal total assembly time is

$$\hat{T}^M = \sum_{i=1}^{m-1} \frac{t^{BP}_{M_i,M_{i+1}}}{\prod_{j=i}^{m-1} p^{BP}_{M_j,M_{j+1}}} + \max_{i=1\ldots m} \sum_{j=1}^{n_i-1} \frac{t^{M_i}_{j,j+1}}{\prod_{k=j}^{n_i-1} p^{M_i}_{k,k+1}} \qquad (6)$$

where $t^{BP}_{M_i,M_{i+1}}$ is the assembly time of modules M_i and M_{i+1}. Again, here, we assume there are enough working stations to accommodate any modularization.

In general, the number of possible partitions of a set that consists of n elements into m subsets is given by the *Stirling number of the second kind*, also known as *Stirling partition number* (SPN)

$$S(n,m) = \frac{1}{m!} \sum_{i=0}^{m} (-1)^i \frac{m!}{i!(m-i)!} (m-i)^n \qquad (7)$$

The SPN considers the number of different partitions but not the order of the elements within each subset. Therefore, the number of permutations of dividing n elements into clusters is

$$N = \sum_{m=1}^{n} S(n,m) \prod_{i=1}^{m} k_i! = S(n,m) \times n! \qquad (8)$$

where k_i is the number of elements in the ith module.

The combined modularization and sequencing problem is NP-hard and therefore can be solved only for simplified cases. Simplification can be done by limiting the total number of elements or by introducing sequential constraints. For example, maintaining a constant sequence reduces the number of possible partitions to 2^{n-1} for the n elements' system. To illustrate this argument, consider a system that consists of eight elements that are assembled in seven sequential tasks. Assume the probabilities for success completion of the assembly tasks are given by the matrix in Fig. 5. The probability of joining every pair of components *i* and *j* is the element of the *i*th column and *j*th row of the matrix. Given this, the probabilities for the sequence 1 → 2 → 3 → 4 → 5 → 6 → 7 → 8 are in the ascending order 0.91, 0.92, 0.93, 0.94, 0.95, 0.96, 0.97, respectively, where task 1 is the assembly of elements 1 and 2 and task 7 is the assembly of elements 7 and 8. For the sequence 8 → 7 → 6 → 5 → 4 → 3 → 2 → 1, the probabilities are in the descending order 0.97, 0.96, 0.95,…, 0.91. For clarification, assume identical costs and times of 1 ($c_{i,j} = t_{i,j} = 1\ \forall\ i, j$) and $c_{i,j}^{M_k} = c_{i,j}$, $t_{i,j}^{M_k} = t_{i,j}$, and $p_{i,j}^{M_k} = p_{i,j}$ such that modularization does not affect the internal costs, times, and reliabilities within a module. Also, assume that $c_{M_i,M_{i+1}i}^{BP} = 1$ (the cost of assembling any two modules is 1, and that $p_{M_i,M_{i+1}}^{BP} = p_{m,n}$ where m is the last element of module i and n is the first element of module i + 1). The final assumption is that the modules are mutually independent such that a failure in assembling two modules does not affect previously assembled modules. There are $2^{8-1} = 128$ possible modular structures for the given sequence, ranging from a single module structure ([8]) to a monolithic assembly structure ([1]). For example, the eight elements can be divided into two modules in seven different configurations ([1, 7], [2, 6], [3, 5], [4], [3, 5], [2, 6], [1, 7]), into three subassemblies in 21 different ways, into four subassemblies in 35 different ways, into five modules in 35 ways, and so on. All 128 MVs for an eight-component system are presented in Table 1 in the Appendix.

We can analyze the best modularization for particular sequences. For example, consider two extreme sequences: ascending and descending probability sequences. In the ascending sequence elements are assembled in the order of 1 → 2 → 3 → 4 → 5 → 6 → 7 → 8, whereas in the descending sequence, the elements are assembled in the order of 8 → 7 → 6 → 5 → 4 → 3 → 2 → 1. The minimum expected cost and time for ascending order occurs at MV = [2, 3], and that for descending order occurs at MV = [2, 3]. However, these two configurations are not optimum when all other sequences are considered, and despite their symmetry, they do not lead to similar assembly costs and times.

P↲	1	2	3	4	5	6	7	8
1	1	0.91	0.91	0.91	0.91	0.91	0.91	0.91
2	0.91	2	0.92	0.92	0.92	0.92	0.92	0.92
3	0.91	0.92	3	0.93	0.93	0.93	0.93	0.93
4	0.91	0.92	0.93	4	0.94	0.94	0.94	0.94
5	0.91	0.92	0.93	0.94	5	0.95	0.95	0.95
6	0.91	0.92	0.93	0.94	0.95	6	0.96	0.96
7	0.91	0.92	0.93	0.94	0.95	0.96	7	0.97
8	0.91	0.92	0.93	0.94	0.95	0.96	0.97	8

Fig. 5 The assembly success probability matrix used for the design of optimum assembly

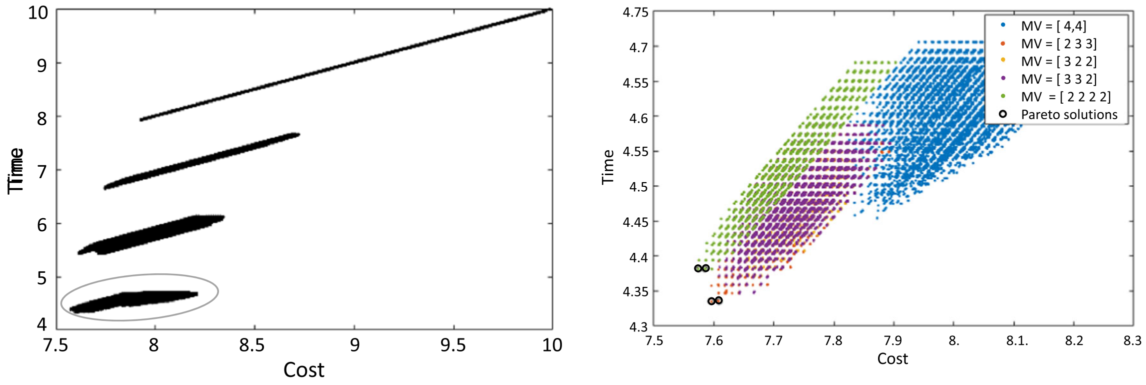
Figure 6 shows the expected cost (Fig. 6a) versus the expected time (Fig. 6b) for all possible scenarios which consist of 128 MVs multiplied by 40,320 sequences, a total of 516,0960 points. Since the minimization of cost and time possibly requires different sequences, we thus suggest using the Pareto optimization to find the non-dominated solutions (sequences and MVs) for the minimum expected cost and time problem. Figure 6b shows the specified region in Fig. 6a, which is a Pareto region, and also the Pareto solutions (global Pareto solutions) within the region. Five MVs constitute all the points in this region: [4], [2, 3], [2, 3], [2, 3], and [2]. The Pareto point characteristics are as follows:

- $\hat{C}^M = 7.58$ and $\hat{T}^M = 4.38$ for MV = [4] and sequences 2 → 4 → 3 → 5 → 6 → 8 → 7 → 1 and 2 → 4 → 3 → 5 → 6 → 7 → 8 → 1.
- $\hat{C}^M = 7.57$ and $\hat{T}^M = 4.38$ for MV = [4] and sequences 2 → 3 → 4 → 5 → 6 → 8 → 7 → 1 and 2 → 3 → 4 → 5 → 6 → 7 → 8 → 1.
- $\hat{C}^M = 7.61$ and $\hat{T}^M = 4.33$ for MV = [2, 3] and sequences 1 → 3 → 2 → 6 → 8 → 7 → 5 → 4 and 1 → 3 → 2 → 6 → 7 → 8 → 5 → 4.
- $\hat{C}^M = 7.59$ and $\hat{T}^M = 4.33$ for MV = [4] and sequences 1 → 2 → 3 → 6 → 8 → 7 → 5 → 4 and 1 → 2 → 3 → 6 → 7 → 8 → 5 → 4.

From Fig. 6b, an important observation can be made: that the MVs in the Pareto region have particular characteristics that also lead to a useful heuristic for modular assembly formation. First, the size of each module in these MVs is lower or equal to $n/2$ (= 4). Second, these MVs are the most balanced ones, meaning they have minimum standard deviation (SD) among the categories of similar sized MVs

$$\begin{array}{r} C1: m \le \frac{n}{2} \\ C2: \min\limits_{\mathrm{MV}_m} \mathrm{SD}[\mathrm{MV}] \end{array} \tag{9}$$

Note that the two constraints need to be observed together and that this heuristic does not advise on a particular sequence. The standard deviation of MVs can be found in Table 1 in the



(a) (b)

Fig. 6 Panel **a** shows the plot of assembly time versus the cost for each configuration determined by a MV and a sequence. Panel **b** shows the close up look into the region circled out in **a** and the MV numbers that form that region. The Pareto solutions are also shown

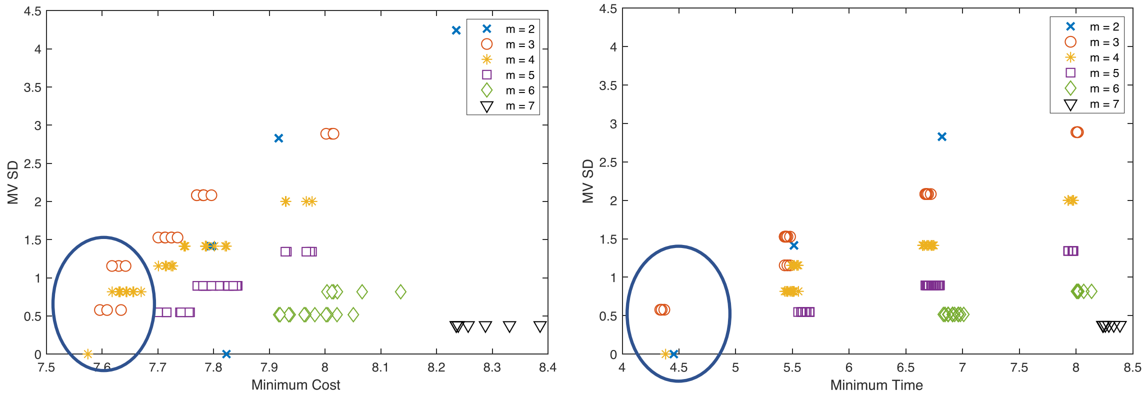
Appendix. These two rules can be examined further in Fig. 7a, b where the minimum expected cost and minimum time for each MV (over all sequences) are respectively plotted against the standard deviation of corresponding MVs. The figure classifies different solutions according to the size of the MVs (m). To verify this heuristic, a further example with different P and C matrices is shown in Fig. 12 in the Appendix.

4 Modularization costs

Let us now loosen some of the assumptions used for simplification in the previous section by considering the effect of modularization and modular interface creation on the expected cost. In simple mechanical systems, the additional connection costs of the modules are due mainly to supplementary manufacturing cost of the mechanical components. A simple illustration of this is brought in Fig. 8. However, in more complex systems, the connection costs between modules, as well as the probabilities for success, depend on the relations between the elements of the modules as determined by the design structure matrices (DSMs) or by connectivity graphs. The connection costs and probabilities between modules are determined by the internal costs of connecting the elements within the module and by "bottleneck" or "external" connections, which represent interactions and dependencies between elements that belong to different modules [26, 27]. Figure 9 shows the DSMs of two products [28]: PW4098 Jet Engine (Fig. 9a), and Kodak single-use camera (Fig. 9b). Although the two systems consist of a nearly similar number of modules (9 for the jet engine and 8 for the camera), there are significant differences between their modular structures, as well as differences between modules within the same product. For example, the "waterproof" module of the camera has a single external interface with the "shutter" module (element 4 is the "rubber cover" and element 28 is the "shutter release"), while module 8 of the jet engine has more than ten external connections with each of the other modules in the engine.

In general, the intermodular connection costs and probabilities are given by

$$\begin{aligned} c^{BP}_{M_k,M_{k+1}} &= \mathcal{F}(E(n_i),E(n_{i+1})) \\ p^{BP}_{M_k,M_{k+1}} &= \mathcal{M}(E(n_i),E(n_{i+1})) \end{aligned} \qquad (10)$$



(a) (b)

Fig. 7 The minimum expected assembly cost (**a**) and time (**b**) versus the standard deviation of their MVs. The encircled regions show that validity of the two-step heuristic

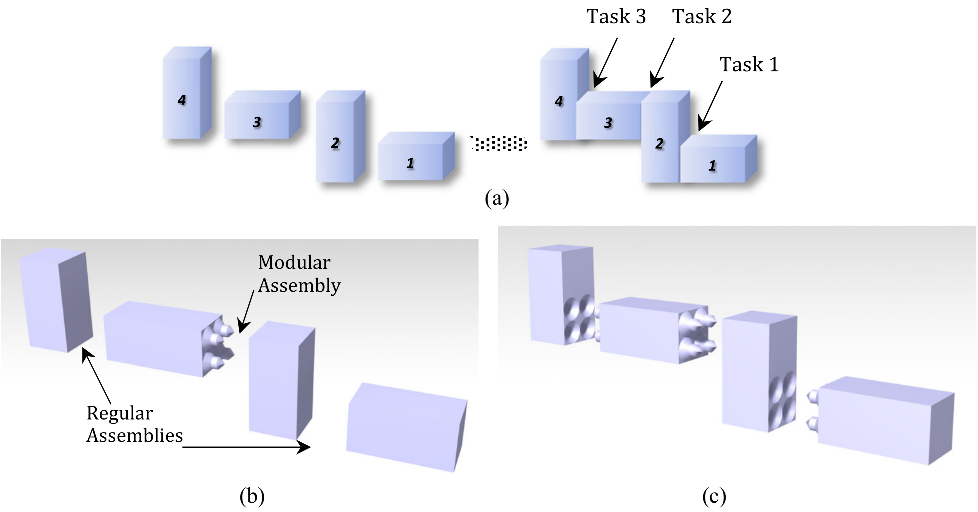


Fig. 8 Assembling four blocks using simple mutually dependent connection (**a**), modified modular assembly interfaces and mutually dependent connections (**b**), and a fully modular assembly configuration with modular interfaces (**c**)

where $E(n_i)$ and $E(n_{i+1})$ are the subsets of elements in modules k and $k+1$, respectively. Similarly, the costs and probabilities of the internal connections between elements within a module are determined by the costs and probabilities of the internal connections given by

$$\begin{aligned} c_{i,j}^{M_k} &= \mathcal{H}(E(n_k)) \\ p_{i,j}^{M_k} &= \mathcal{L}(E(n_k)) \end{aligned} \tag{11}$$

where $E(n_k)$ is a subset of elements in module k. As mentioned in Section 3, $c_{i,j}^{M_k}$ and $p_{i,j}^{k}$ may be different from $c_{i,\,j}$ and $p_{i,\,j}$ due to special features within the module. The goal is to determine an optimal modular structure that minimizes the total expected assembly cost and time as given by Eqs. (5) and (6).

A simple example to demonstrate the above follows. Consider the binary DSM of a system that consists of the eight elements shown in Fig. 10a. The optimal clustering of this system results in three subassemblies without any external (bottleneck) connections (three independent subsystems), and therefore, the only costs are related to the internal connections of the elements within each subassembly (Fig. 10b). Furthermore, a failure in any assembly task can affect only the tasks in the same cluster. Assume the nominal connection costs and probabilities between the elements as $c_{i,\,j} = t_{i,\,j} = 1$ and $p_{i,\,j}$ are those shown in Fig. 5, with no assembly sequence constraints. If we assume that the assembly cost between any two modules is determined by $c_{M_k,M_{k+1}}^{BP} = t_{M_k,M_{k+1}}^{BP} = \left(\sum c_{i,j}^{b}\right)$, where $c_{i,j}^{b}$ is the cost of a bottleneck connection between elements i and j (let us refer to this by case 1), the optimal modularization and sequence is that depicted in Fig. 10c.

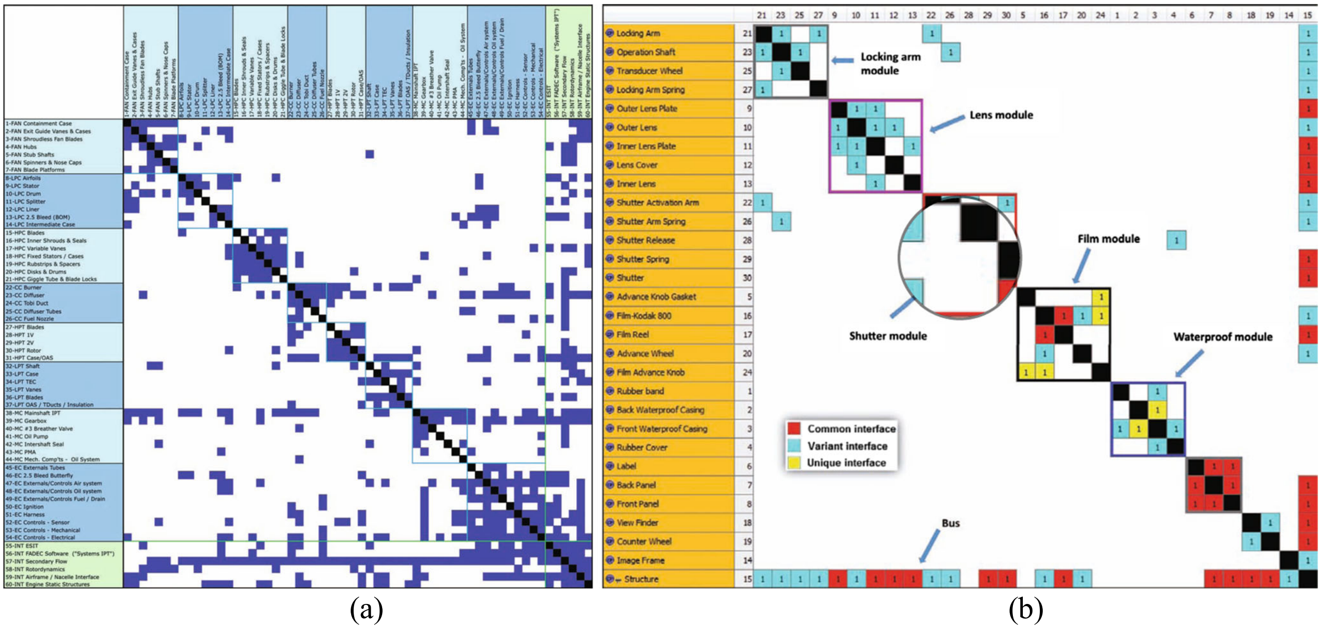
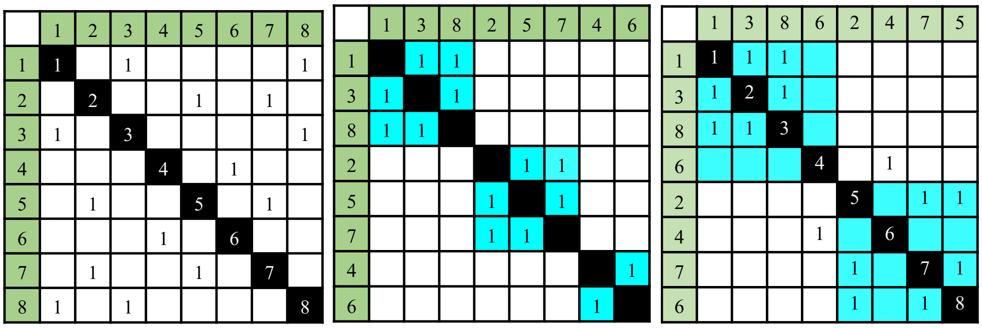


Fig. 9 Design structure matrices (DSMs) for the PW4098 Jet Engine system (**a**) and the Kodak single-use camera (**b**) [23]

Fig. 10 DSM for the eight elements' system ($c = t = 9.4250$) (**a**) and the optimal clustering of three subassemblies ($c = 8.7321$, $t = 5.4715$) (**b**) and optimal clustering for minimum assembly cost and time (**c**)



	1	2	3	4	5	6	7	8
1	1		1					1
2		2			1		1	
3	1		3					1
4				4		1		
5		1			5		1	
6				1		6		
7		1			1		7	
8	1		1					8

(a)

	1	3	8	2	5	7	4	6
1		1	1					
3	1		1					
8	1	1						
2					1	1		
5				1		1		
7				1	1			
4								1
6							1	

(b)

	1	3	8	6	2	4	7	5
1	1	1	1					
3	1	2	1					
8	1	1	3					
6				4		1		
2					5		1	1
4				1		6		
7					1		7	1
6					1		1	8

(c)

However, if we raise the assumed cost by $c^{BP}_{M_k,M_{k+1}} = t^{BP}_{M_k,M_{k+1}} = \left(\sum c^{b}_{i,j}\right)(n_k + n_{k+1})$, where n_k and n_{k+1} are the number of elements in modules k and $k + 1$, respectively (and let us refer to this scenario as case 2), the optimal modularization is no modularization (Fig. 10a). Figure 11a, b shows the general landscape for the cost and time of different modularizations under cases 1 and 2, respectively. For case 1, the following sequences lead to the minimum cost and time of 7.8692 and 4.4795, respectively, when used with MV = [4]:

1→3→8→6→2→4→7→5
1→3→8→6→2→4→5→7
1→3→6→8→2→4→7→5
1→3→6→8→2→4→5→7

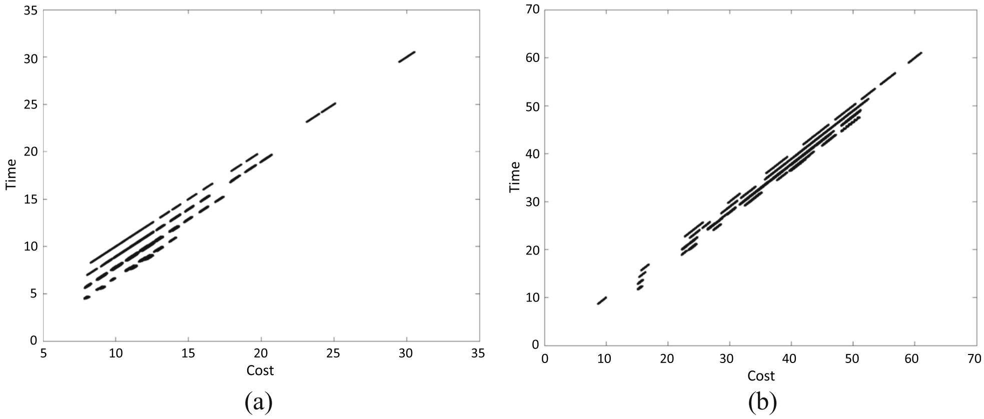
All four sequences lead to similar probabilities (0.9100, 0.9300, 0.9600, 0.9200, 0.9200, 0.9400, 0.9500). This example shows the importance of interface cost modeling for sequencing and modularization problem.

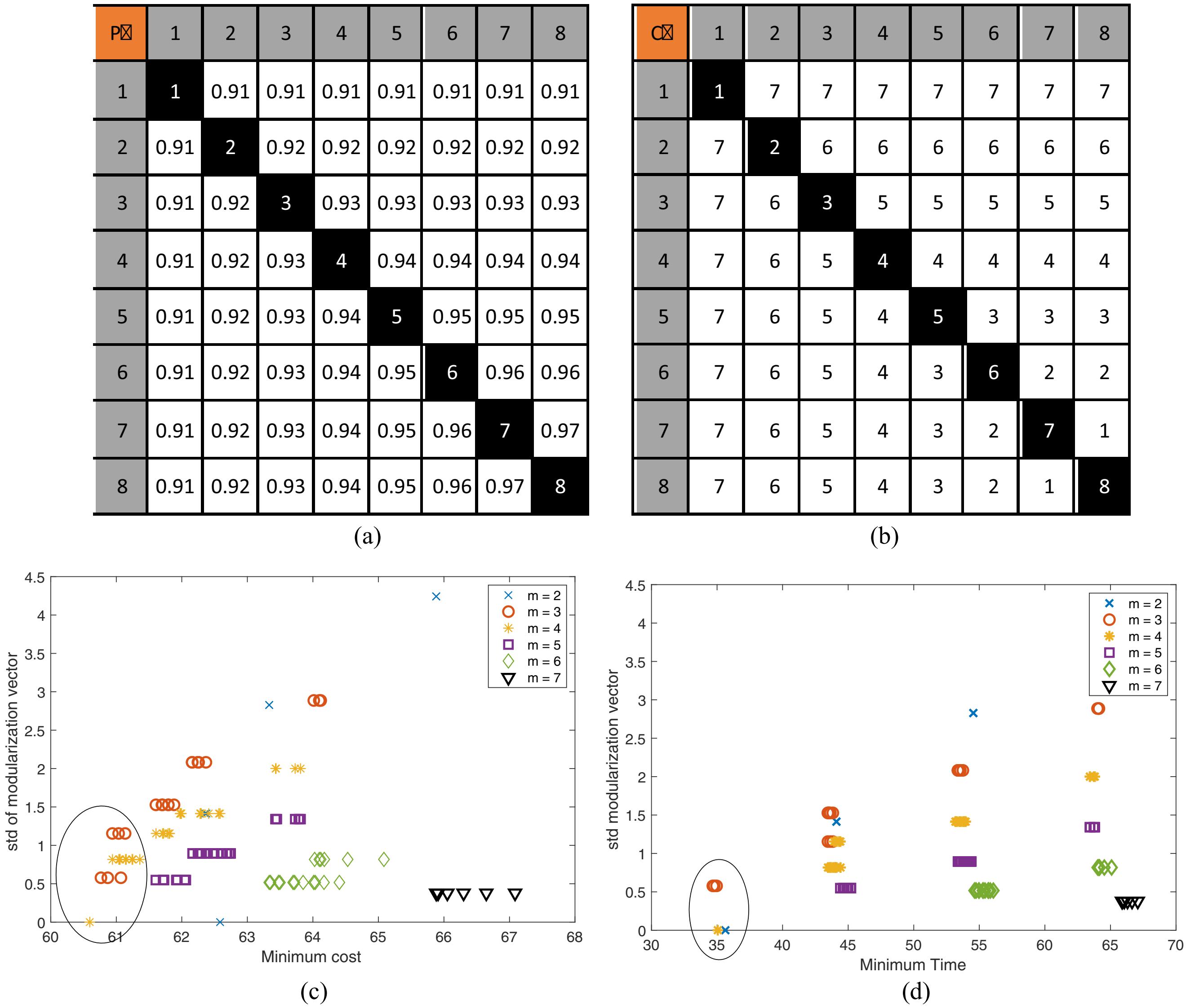
5 Conclusions

Modularity is an important feature in any complex system. It is particularly important in cyber-physical systems (CPSs), where many physical and computational resources are integrated and interconnected. Modularity is one of the major principles in the Industry 4.0 structure, as systems are becoming more complex, and the traditional production hierarchy is replaced by alternative dynamic structures.

This paper considered the design of complex assembly processes, given that each assembly task has cost and reliability values associated with it. The reliability value denotes the probability of the assembly task to be completed successfully in a single attempt. Given the availability and affordability of a wide range of sensors and quality assurance techniques, each task in the assembly process can be monitored, and if required, the process is adjusted according to the updated information. In particular, the paper focuses on the assembly process in which the tasks are mutually dependent such that a failure of a specific assembly task is backpropagated to the previous tasks. In such a case, the assembly must restart from previous tasks and, in the extreme case, even from the first tasks. Our previous work focused on the effect of task sequencing in assembly processes with probabilistic success rate on the expected process cost and duration. This paper illustrates the importance of considering task modularization prior to task sequencing in complex assemblies in order to reduce the effect of the mutual failure dependency. Simulation results indicate that a balanced modularization, in terms of the number of assembly tasks within each module, as well as sequencing the tasks in a descending order of the probability for failure of each task, improves the process performance. The cost savings that might result from this simple heuristic are

Fig. 11 The assembly time versus cost for case 1 (**a**) and for case 2 (**b**)





P⊠	1	2	3	4	5	6	7	8
1	1	0.91	0.91	0.91	0.91	0.91	0.91	0.91
2	0.91	2	0.92	0.92	0.92	0.92	0.92	0.92
3	0.91	0.92	3	0.93	0.93	0.93	0.93	0.93
4	0.91	0.92	0.93	4	0.94	0.94	0.94	0.94
5	0.91	0.92	0.93	0.94	5	0.95	0.95	0.95
6	0.91	0.92	0.93	0.94	0.95	6	0.96	0.96
7	0.91	0.92	0.93	0.94	0.95	0.96	7	0.97
8	0.91	0.92	0.93	0.94	0.95	0.96	0.97	8

(a)

C⊠	1	2	3	4	5	6	7	8
1	1	7	7	7	7	7	7	7
2	7	2	6	6	6	6	6	6
3	7	6	3	5	5	5	5	5
4	7	6	5	4	4	4	4	4
5	7	6	5	4	5	3	3	3
6	7	6	5	4	3	6	2	2
7	7	6	5	4	3	2	7	1
8	7	6	5	4	3	2	1	8

(b)

(c)

(d)

Fig. 12 The minimum cost (**c**) and minimum time (**d**) (over all sequences) for 128 MVs for the *P*, *C*, and *T* (=*C*) matrices shown in **a** and **b**

calculable and depend on the nature of the assembly system and the assembled product. For simple assemblies, the presented heuristic does not hold, as the only factors affecting their sequencing are the assembly liaison matrices and precedence relations [29]. However, as the assembly process becomes more complex, the advantages of modular assembly become more apparent. The paper unveils several hidden complexities even for presumably simple assembly processes.

The implementation of the proposed method requires considerations at both the product design and the assembly process planning stages. At the product design stage, the interaction between the different components, mainly in terms of the product's liaison graph and the assembly sequence, must be considered. In particular, a failure in the assembly process of a particular component may have an effect on other components and subassemblies. In the assembly process stage, the planners should consider the effect of failures on the logistics of the assembly process, in terms of the cost of each assembly operation, as well as the required resources in the assembly operations (e.g., manpower, special machinery and tools, transportation, assembly scheduling).

In practice, designers and planners can use standard risk management tools such as probabilistic risk assessment (PRA), event tree analysis (ETA), and fault tree analysis (FTA) to address the uncertainties with metrics, parameterization, and prioritization. The PRA tools, also known as likelihood consequence or probability impact tools, use the magnitude and the likelihood of potential failures in order to express the expected loss. The PRA tools can provide the designers and planners with insights regarding the sensitive components and tasks that can initiate significant failures, the detriments of these failures, and their probabilities or frequencies. The ETA and FTA tools can guide designers and planners in optimizing the use of resources, assist in planning quality assurance and monitoring procedures, and assist in the development of diagnostic manuals and processes.

One limitation to this work is that it uses fairly simple examples to illustrate the utility of the presented model. The number of components in these examples was limited to no more than eight components. This is because we could not efficiently run the model in MATLAB with a greater number of components, primarily due to the large number of sequences that needs to be considered. MATLAB could not run the code for examples with more than 18 components, which is relatively small compared to the size of most complex assemblies. There are two possible ways to address this issue. One is to use precedence graphs as shown in the study of Shoval et al. [23] to limit the number of possible sequences. The second possible solution is to use genetic search algorithms such as for effective search for optimal modularization and sequencing. We will take on the challenge of system size in future work.

Appendix

Table 1 The MVs and their SD and size (*m*) for the eight elements

No.	*m*	SD (MV)	MV
1	1	0	8
2	2	4.2	[1, 7]
3	2	2.8	[2, 6]
4	2	1.4	[3, 5]
5	2	0	[4]
6	2	1.4	[3, 5]
7	2	2.8	[2, 6]
8	2	4.2	[1, 7]
9	3	2.9	[1, 6]
10	3	2.1	[1, 2, 5]
11	3	1.5	[1, 3, 4]
12	3	1.5	[1, 3, 4]
13	3	2.1	[1, 2, 5]
14	3	2.9	[1, 6]
15	3	2.1	[1, 2, 5]
16	3	1.2	[2, 4]
17	3	0.6	[2, 3]
18	3	1.2	[2, 4]
19	3	2.1	[1, 2, 5]
20	3	1.5	[1, 3, 4]
21	3	0.6	[2, 3]
22	3	0.6	[2, 3]
23	3	1.5	[1, 3, 4]
24	3	1.5	[1, 3, 4]
25	3	1.2	[2, 4]
26	3	1.5	[1, 3, 4]
27	3	2.1	[1, 2, 5]
28	3	2.1	[1, 2, 5]
29	3	2.9	[1, 6]
30	4	2	[1, 5]
31	4	1.4	[1, 2, 4]
32	4	1.2	[1, 3]
33	4	1.4	[1, 2, 4]
34	4	2	[1, 5]
35	4	1.4	[1, 2, 4]
36	4	0.8	[1–3]
37	4	0.8	[1–3]
38	4	1.4	[1, 2, 4]
39	4	1.2	[1, 3]
40	4	0.8	[1–3]
41	4	1.2	[1, 3]
42	4	1.4	[1, 2, 4]
43	4	1.4	[1, 2, 4]
44	4	2	[1, 5]
45	4	1.4	[1, 2, 4]
46	4	0.8	[1–3]
47	4	0.8	[1–3]
48	4	1.4	[1, 2, 4]
49	4	0.8	[1–3]
50	4	0	[2]
51	4	0.8	[1–3]
52	4	0.8	[1–3]
53	4	0.8	[1–3]
54	4	1.4	[1, 2, 4]
55	4	1.2	[1, 3]
56	4	0.8	[1–3]
57	4	1.2	[1, 3]
58	4	0.8	[1–3]
59	4	0.8	[1–3]
60	4	1.2	[1, 3]
61	4	1.4	[1, 2, 4]
62	4	1.4	[1, 2, 4]
63	4	1.4	[1, 2, 4]
64	4	2	[1, 5]
65	5	1.3	[1, 4]
66	5	0.9	[1–3]
67	5	0.9	[1–3]
68	5	1.3	[1, 4]
69	5	0.9	[1–3]
70	5	0.5	[1, 2]
71	5	0.9	[1–3]
72	5	0.9	[1–3]

Table 1 (continued)

No.	*m*	SD (MV)	MV
73	5	0.9	[1–3]
74	5	1.3	[1, 4]
75	5	0.9	[1–3]
76	5	0.5	[1, 2]
77	5	0.9	[1–3]
78	5	0.5	[1, 2]
79	5	0.5	[1, 2]
80	5	0.9	[1–3]
81	5	0.9	[1–3]
82	5	0.9	[1–3]
83	5	0.9	[1–3]
84	5	1.3	[1, 4]
85	5	0.9	[1–3]
86	5	0.5	[1, 2]
87	5	0.9	[1–3]
88	5	0.5	[1, 2]
89	5	0.5	[1, 2]
90	5	0.9	[1–3]
91	5	0.5	[1, 2]
92	5	0.5	[1, 2]
93	5	0.5	[1, 2]
94	5	0.9	[1–3]
95	5	0.9	[1–3]
96	5	0.9	[1–3]
97	5	0.9	[1–3]
98	5	0.9	[1–3]
99	5	1.3	[1, 4]
100	6	0.8	[1, 3]
101	6	0.5	[1, 2]
102	6	0.8	[1, 3]
103	6	0.5	[1, 2]
104	6	0.5	[1, 2]
105	6	0.8	[1, 3]
106	6	0.5	[1, 2]
107	6	0.5	[1, 2]
108	6	0.5	[1, 2]
109	6	0.8	[1, 3]
110	6	0.5	[1, 2]
111	6	0.5	[1, 2]
112	6	0.5	[1, 2]
113	6	0.5	[1, 2]
114	6	0.8	[1, 3]
115	6	0.5	[1, 2]
116	6	0.5	[1, 2]
117	6	0.5	[1, 2]
118	6	0.5	[1, 2]
119	6	0.5	[1, 2]
120	6	0.8	[1, 3]
121	7	0.4	[1, 2]
122	7	0.4	[1, 2]
123	7	0.4	[1, 2]
124	7	0.4	[1, 2]
125	7	0.4	[1, 2]
126	7	0.4	[1, 2]
127	7	0.4	[1, 2]
128	8	0	[1]

References

1. Weyer S, Schmitt M, Ohmer M, Gorecky D (2015) Towards Industry 4.0—standardization as the crucial challenge for highly modular, multi-vendor production systems. IFAC-PapersOnLine 48(3):579–584
2. Zamfirescu CB, Pirvu B-C, Loskyll M, Zuehlke D (2014) Do not cancel my race with cyber-physical systems. IFAC-PapersOnLine 47(3):4346–4351
3. Bortolini M, Galizia FG, Mora C (2018) Reconfigurable manufacturing systems: literature review and research trend. J Manuf Syst 49:93–106
4. Bortolini M, Ferrari E, Gamberi M, Pilati F, Faccio M (2017) Assembly system design in the Industry 4.0 era: a general framework. IFAC-PapersOnLine 50(1):5700–5705. https://doi.org/10.1016/j.ifacol.2017.08.1121
5. Thramboulidis K, Kontou I, Vachtsevanou DC (2018) Towards an IoT-based framework for evolvable assembly systems. IFAC-PapersOnLine 51(11):182–187. https://doi.org/10.1016/j.ifacol.2018.08.255
6. Baudin M (2002) Lean assembly: the nuts and bolts of making assembly operations flow. CRC, New York
7. Bortolini M, Faccio M, Gamberi M, Pilati F (2017) Multi-objective assembly line balancing considering component picking and ergonomic risk. Comput Ind Eng 112:348–367. https://doi.org/10.1016/j.cie.2017.08.029
8. Boysen N, Fliedner M, Scholl A (2007) A classification of assembly line balancing problems. Eur J Oper Res 183(2):674–693
9. Boothroyd G (1987) Design for assembly—the key to design for manufacture. Int J Adv Manuf Technol 2(3):3–11
10. Fan J, Dong J Intelligent virtual assembly planning with integrated assembly model. In proceedings of IEEE International Conference on Systems, man and cybernetics, October 5-8, 2003, Washington, D.C., USA pp. 4803–4808
11. Boothroyd G, Dewhurst P, Knight WA (2010) Product design for manufacture and assembly. CRC Press, London
12. Burke GJ, Carlson JB (1989) DFA at Ford Motor Company. DFMA Insight magazine 1(4):1–10
13. Yoosufani Z, Boothroyd G (1978) Design of parts for ease of handling. Technical report, Department of Mechanical Engineering, University of Massachusetts, report no 2
14. Boothroyd G (1979) Design for manual handling and assembly. Report no. 4. Department of Mechanical Engineering, University of Massachusetts, Amherst
15. Seth B, Boothroyd G (1979) Design for manual handling. Technical report, Department of Mechanical Engineering, University of Massachusetts, Report no. 9
16. Yoosufani Z, Ruddy M, Boothroid G (1983) Effect of part symmetry on manual assembly times. J Manuf Syst 2(2):189–195
17. Corbett J, Crookall J (1986) Design for economic manufacture. CIRP Ann Manuf Technol 35(1):93–97
18. Ishikawa K (1976) Guide to quality control: industrial engineering and technology. Asian Productivity Organization, Tokyo
19. Harry MJ, Stewart R (1988) Six sigma mechanical design tolerancing. Motorola University Press, Schaumburg
20. Shimbun NK (1989) Poka-yoke: improving product quality by preventing defects. CRC Press, New York
21. Hinckley M (1993) A global conformance quality model—a new strategic tool for minimizing defects caused by variation, error, and complexity. Doctoral Thesis, Stanford University
22. Efatmaneshnik M, Ryan Mj (2015) On optimal modularity for system construction. Complexity 21(5):176–189 https://doi.org/10.1002/cplx
23. Shoval S, Efatmaneshnik M, Ryan MJ (2017) Assembly sequence planning for processes with heterogeneous reliabilities. Int J Prod Res 55(10):2806–2828
24. Simon HA (1962) The architecture of complexity. Proc Am Philos Soc 106(6):467–482
25. Efatmaneshnik M, Shoval S, Qiao L (2018) A Standard Description of the Terms Module and Modularity for Systems Engineering, IEEE Transactions on Engineering Management. https://doi.org/10.1109/TEM.2018.2878589
26. Shoval S, Qiao L, Efatmaneshnik M, Ryan M (2016) Dynamic modular architecture for product lifecycle. Procedia CIRP 48:271–276
27. Shoval S (2016) Dynamic modularization throughout system lifecycle using multilayer design structure matrices. Procedia CIRP 40:85–90. https://doi.org/10.1016/j.procir.2016.01.062
28. Eppinger SD, Browning TR (2012) Design structure matrix methods and applications. MIT Press, Cambridge
29. Boothroyd G (2005) Assembly automation and product design. CRC Press, London