Assembly System Configuration Design for Reconfigurability under Uncertain Production Evolution

Zhengqian Jiang, Hui Wang[1], ASME Member
Department of Industrial and Manufacturing Engineering
Florida A&M University-Florida State University College of Engineering
2525 Pottsdamer St., Tallahassee, FL 32310
[1] Contact Author: hwang10@fsu.edu

Maxim A. Dulebenets, Junayed Pasha
Department of Civil and Environmental Engineering
Florida A&M University-Florida State University College of Engineering
2525 Pottsdamer St., Tallahassee, FL 32310

ABSTRACT

Assembly system configuration determines the topological arrangement of stations with defined logical material flow among them. The design of assembly system configuration involves (1) subassembly planning that defines subassembly tasks and between-task material flows and (2) workload balancing that determines the task-station assignments. The assembly system configuration should be flexibly changed and updated to cope with product design evolution and updating. However, the uncertainty in future product evolution poses significant challenges to the assembly system configuration design since the higher cost can be incurred if the assembly line suitable for future products is very different from that for the current products. The major challenges include (1) the estimation of reconfiguration cost, (2) unavailability of probability values for possible scenarios of product evolution, and (3) consideration of the impact of the subassembly planning on the task-station assignments. To address these challenges, this paper formulates a concurrent optimization problem to design the assembly system configuration by jointly determining the subassembly planning and task-station assignments considering uncertain product evolution. A new assembly-hierarchy similarity model is proposed to estimate the reconfiguration effort by comparing the commonalities among different subassembly plans of current and potential future product designs. The *assembly system configuration is chosen by maximizing both assembly-hierarchy similarity and assembly system throughput under the worst-case scenario. A case study motivated by real-world scenarios demonstrates the applicability of the proposed method including scenario analysis.*

Keywords: *assembly system configuration; optimization; subassembly planning; reconfigurability; assembly system.*

1. INTRODUCTION

Assembly system configuration or logical layout refers to the topological arrangement among the machines or assembly stations with logical material flows among them, regardless of their physical locations. Thus, it reflects a logical layout of stations/machines and is a critical step in the early stage of assembly system design. With the expedited product development for meeting a variety of market demands, manufacturers need to constantly adjust their assembly system configuration to launch new products. The assembly system configuration design should be created flexible to new product lines within a short period and strike a balance between system reconfigurability and productivity to maximize the manufacturers' long-term profit.

Benkamoun et al. [1] listed resources planning, layout configuration of stations, distribution of operations among stations, assignment of resources, and line balancing as possible sub-activities for assembly system configuration designs. Prior to these activities, subassembly modules along with the associated assembly tasks should be determined by assembly hierarchy, which concerns the hierarchical material flows among these tasks. Fig. 1 shows an example of one serially linked product design with four components and three assembly tasks. Two possible assembly hierarchies are given in Fig. 2, where

Hierarchy 1 represents a sequential way of assembling AB, C, and D incrementally, while Hierarchy 2 represents a parallel way of assembling AB and CD as two subassemblies, and then combining them by assembly task 2 to complete the product A-B-C-D. In summary, the assembly system configuration involves two decisions including: (1) the process planning to determine subassembly modules, the associated tasks, and material flows among these tasks and (2) assignment of these subassembly tasks to different stations.

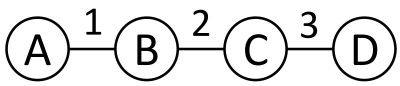


Fig. 1.A serially linked product design

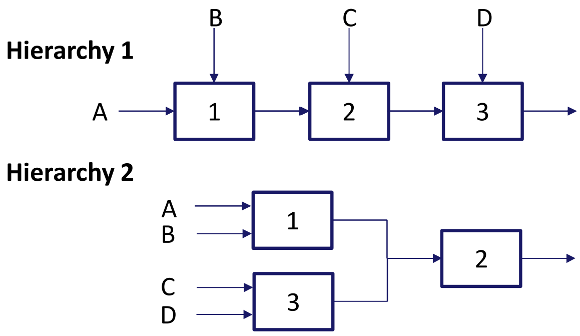


Fig. 2. Two possible assembly hierarchies for the product design in Fig. 1

The problem of exploring all possible assembly hierarchies is mostly equivalent to subassembly identification, which determines the subassembly module at each level of assembly hierarchy.

In addition, the assembly hierarchy also determines the assembly tasks to create the subassembly modules. Different methods have been used to generate candidate subassembly modules. For example, Homem de Mello and Sanderson [2] and Baldwin et al. [3] used a "cut-set" method to enumerate all the subassemblies based on the assembly liaison diagram. The advantage of adopting the assembly liaison diagram is the convenience of implementation on computers. Jiang [4, 5] also employed a liaison graph method to realize the automated assembly hierarchy generation algorithm. Massive research has been conducted on subassembly identification as reviewed by Wang and Liu [6]. These research efforts mostly dealt with one generation of the products and did not consider the improvement of assembly system reconfigurability for future product evolution.

Koren et al. [7] studied the reconfigurable manufacturing system (RMS), which adopts modular components and interfaces to cost-effectively create a variety of manufacturing system configurations in response to dynamic market demands. The reconfigurability of an RMS can be expressed as the ability to adjust the capacity and functionality of production by rearranging/changing system components or modules. It includes several inter-related characteristics, such as customization, convertibility, scalability, modularity, integrability, diagnosability, module mobility as well as automatability [8, 9]. The RMS research focused on different aspects of the reconfigurability as reviewed in [9-11]. The reconfiguration problem has been treated as either a scalability or convertibility problem [11-14]. The scalability planning considers the changes in production volumes [15, 16], and the convertibility planning dealt with the changes in product functionality [17]. Benkamoun et al. [1] focused on the assembly system configuration design to deal with product variety and frequent market changes.

The uncertainty conditions have attracted attentions for different manufacturing systems[18-20]. The uncertainty of production performance and reconfiguration in RMS also significantly affects assembly system designs. Several aspects of the uncertainty have been considered in the existing research with a primary focus on stochastic leading time [21] and demand uncertainty [22]. The evolvement of assembly systems is another important aspect of the RMS research [23]. Several evaluation criteria have been developed to help select the most appropriate configuration without the detailed information on future system evolvements such as throughput, investment cost, floor space, and system quality [8, 24, 25]. A number of methods were developed to determine the reconfiguration planning by optimizing the reconfiguration time and cost [26, 27].

When the product evolution is considered, product family design and platform-based product development have been massively studied over the last decade [28, 29]. A product family refers to a set of similar products that are derived from a common platform with specific features to meet certain requirements [30]. There are different metrics for product family design including modularity, commonality, variety, cost and other platform-related metrics [28]. In order to evaluate the similarity among different product designs in a family, multiple methods have been proposed for the identification of commonality [31, 32]. Thevenot and Simpson [33] compared various commonality indices for assessing product families. These studies addressed product evolution from the perspective of product designs. The co-evolution of product family design and manufacturing system has been studied in [34, 35]. Limited research incorporated the consideration of product evolution into assembly system configuration design. When dealing with the RMS, prior research dealt with the estimation of future reconfigurations based on a vision of new product designs, followed by the design of the assembly system configuration for the convenient or low-cost transition between product generations [36, 37]. Life cycle cost models were proposed to estimate the reconfiguration cost between different product generations and manufacturing cost within a product generation [14, 17, 38]. Yuan et al. [39] proposed a multi-objective optimization scheduling model to minimize the cost of assembly line reconstruction for a reconfigurable assembly system. Ko and Hu [40, 41] considered product evolution and task recurrence for the manufacturing system design for line balancing. These methods required the assembly sequence or assembly hierarchy to be specified.

Through the literature review of the assembly system design and reconfiguration planning, the following **research challenges and gaps** have been identified:

- *Characterizing the impact of product evolution uncertainty on the assembly system configuration*. Product evolution exhibits uncertainty due to different product line designs, i.e., there are a number of choices for the designs of the next generation product line. The assembly system configuration should be designed to reduce the potential reconfiguration cost even if the product lines change/upgrade with uncertainty. State-of-the-art methods used probability values to characterize the uncertainty of different evolution scenarios. However, in the early stage of assembly system designs, there might be a lack of sufficient data to model the product evolution probability and any change in the probability can affect the system design outputs.
- *Understanding the impact of subassembly planning/re-planning and the material flow on assembly system reconfiguration*. Multiple reconfiguration

cost models have been proposed for assembly system designs [14, 17, 37, 40]. However, the existing research focused on the task-station reassignment without considering the material flows among different assembly tasks that are affected by the selected subassembly modules. Consideration of such subassembly planning would also help arrange the conveyors and machines in one station to reduce the potential reconfiguration cost.

- *Jointly optimizing subassembly re-planning and system reconfiguration strategy under product evolution*. When considering the uncertain product evolution, most research dealt with the two problems separately. Nevertheless, the two problems are closely related due to the impact of subassembly planning/re-planning on assembly system reconfiguration. As such, the decision on one problem could significantly influence the potential solutions for the other problem.
- *Developing an efficient solution algorithm for assembly system configuration*. Mathematical formulations for the concurrent design problem of subassembly re-planning and system reconfiguration usually have non-deterministic polynomial-time hard (NP-hard) complexity. It has been shown that when dealing with large-size problems, metaheuristic algorithms are more computationally efficient [17, 42]. There is a lack of research that customizes the metaheuristic algorithm for the concurrent optimization in the assembly system configuration design under uncertain product evolution.

This paper develops a method to optimize the assembly system configuration that jointly concerns assembly system reconfiguration and subassembly re-planning considering uncertain product evolution. The optimization is based on a model of assembly hierarchy similarity for estimating potential system reconfiguration effort by comparing the similarity between different assembly hierarchies. Unlike prior research, this model does not require the assumption of any probability values for the uncertainty scenarios, nor detailed values for the reconfiguration cost. The paper also develops an efficient Evolutionary Algorithm (EA) with the chromosome representation and operators that are customized to deal with the joint optimization problem.

The remainder of this paper is organized as follows. Section 2 formulates a concurrent design problem of the subassembly planning and task-station assignments for improving assembly system configuration design based on the development of a similarity model for assembly hierarchies. Section 3 provides a detailed description of an efficient EA customized to the formulated problem. Applicability of the proposed method is demonstrated by a case study with a scenario analysis in Section 4. Section 5 summarizes the findings from this study.

2. PROBLEM STATEMENT AND FORMULATION

This section presents a mathematical formulation of assembly system configuration design under production evolution. Section 2.1 first discusses the impact of uncertain product evolution on the subassembly planning problem and summarizes two major challenges. Sections 2.2 and 2.3 propose a similarity model and formulate a joint optimization model to deal with the challenges.

2.1 Consideration of product evolution with uncertainty into subassembly planning

The state-of-the-art research utilized the probability to characterize the uncertain product evolution. The traditional approaches encounter the following challenges for implementation.

- *Availability of evolution probability*. On many occasions, the probability data might be not conveniently available in the early stage of product development due to a lack of historical data. It may not be convenient to estimate the probabilistic scenarios for different product evolution paths.
- *Estimation of the reconfiguration effort*. Different methods/models have been proposed to deal with the challenge, such as the task recurrence [21] and life cycle cost [13]. There is a lack of research estimating the reconfiguration effort or reconfigurability ***when the assembly hierarchies should be changed*** to deal with the product evolution. The task recurrence and life-cycle cost models considered the assignments of tasks/resources to the stations. However, the hierarchical material flows among the tasks in one station also influence the subassembly planning, potentially increasing the reconfiguration cost. Specifically, when such subassembly planning changes, the subassemblies dealt with by each station/machine could be different. As shown in Fig. 2, task 2 in the first subassembly planning generates subassembly ABC by combining AB and C, whereas task 2 in the second subassembly planning generates subassembly ABCD by combining different modules, i.e., AB and CD. Although the task 2 in both subassembly plans deals with the same assembly liaison (B-

C), it might be necessary for task 2 to make adjustments and conduct labor training when the subassembly plan changes from one to another.

2.2 Assembly hierarchy-based similarity model

Any change in the assembly system configuration can alter the assembly hierarchy (e.g., Fig. 2) including subassembly modules and the associated assembly tasks, the assembly sequences, and the hierarchical material flows among the tasks. The higher similarity in the assembly hierarchy between the products can allow for more convenient system configuration change when the product evolves, reducing the reconfiguration efforts. Thus, the similarity of assembly hierarchy can be employed to compare assembly sequences and subassembly planning between the product designs at different evolutions.

The assembly hierarchy similarity in this research is developed based on a BMIM (bypassing moves and idle machines) similarity coefficient (Goyal et al. [43]), aiming to estimate the reconfiguration effort. Different product designs in one product family should have common components. The assembly hierarchy similarity is calculated by comparing the material flows of these common components. Consider the examples, presented in Figs. 1 and 2, and examine a sequence, by which each component flows through multiple assembly tasks. In Hierarchy 1, component B flows through assembly tasks 1 to 3 sequentially. Thus, the material flow of component B in Hierarchy 1 is 1→2→3. In Hierarchy 2, component B does not flow through assembly task 3, and the material flow is 1→2. Once the information of material flow is known, the subassembly modules for each assembly task are determined. Tasks 1, 2, and 3 in Hierarchy 1 perform assemblies of A+B, AB+C, ABC+D, respectively. In Hierarchy 2, these tasks assemble A+B, C+D, AB+CD, respectively. As such, only the machines/fixtures/labors for Task 1 can be directly reusable when the assembly hierarchy evolves with the product design. For Tasks 2 and 3, the subassembly modules are changed, and the machines/labors may require adjustments or skill training, potentially increasing the reconfiguration cost.

The material flow similarity of the component e between assembly hierarchies u and v is defined as

$$SC_{u,v,e} = \frac{2 \cdot NOTL_{u,v,e}}{2 \cdot NOTL_{u,v,e} + 3 \cdot NOBT_{u,v,e} + NOI_{\ u,v,e}} \ \ \forall u \in A, v \in A, e \in \Omega, \tag{1}$$

where the adopted notations are explained in Table 1. A longest common material flow ($LCMF$) between two material flows is the longest common ordered subsequence in both material flows with the same tasks and precedence relationship. It can also be defined as follows: Given two material flows, X and Y, the goal of finding $LCMF$ is to produce the longest common subsequence such that the longest sequence of task numbers that appear left-to-right (but not necessarily in a contiguous block) in both material flows. The $LCMF$ function is defined as follow:

Let two material flows be defined as $X = (x_1, x_2 \cdots x_m)$ and $Y = (y_1, y_2 \cdots y_n)$, where x_1 and y_1 represent the task numbers in the material flows. Based on X and Y, also define a set of material flows $X_1, X_2 \cdots X_{\mathrm{m}}$, where $X_i = (x_1, x_2 \cdots x_i) \subseteq X, i < m$, and $Y_1, Y_2 \cdots Y_{\mathrm{n}}$, where $Y_j = (y_1, y_2 \cdots y_j) \subseteq Y, j < n$. Let $LCMF(X_i, Y_j)$ represent the set of $LCMF$ of X_i and Y_j. This set of $LCMF$ is given by:

$$LCMF(X_i, Y_j) = \begin{cases} \emptyset & if\ i = 0\ or\ j = 0 \\ \{LCMF(X_{i-1}, Y_{j-1}), x_i\} & if\ x_i = y_j \\ longest\left(LCMF(X_i, Y_{j-1}), LCMF(X_{i-1}, Y_j)\right) & if\ x_i \neq y_j \end{cases} \quad (2)$$

A shortest joint material flow ($SJMF$) is an arrangement obtained from the $LCMF$, which is the shortest possible length of a sequence that accommodates all the tasks of both flows following their precedence constraints. Given two material flows X and Y, a sequence U is a joint material flow of X and Y if some tasks in U can be removed to produce X or Y. An $SJMF$ is a joint material flow with a minimal number of tasks. For example, consider two material flows for a certain component (Tasks 1→2→4→9→5) and (Tasks 1→2→5→9→8). The $LCMF$ of the two flows follows Tasks 1→2→9. The $SJMF$ can be generated from the obtained $LCMF$ by inserting the non-$LCMF$ tasks while preserving the task orders in each material flow. It should be noted that the $SJMF$ is not unique since different $SJMF$s with the same length may be obtained by altering the positions of the inserted tasks in both material flows based on the same $LCMF$. For example, the $SJMF$ of the two flows can be such as Tasks 1→2→4→5→9→5→8, or Tasks 1→2→5→4→9→8→5, etc. The concepts $LCMF$ and $SJMF$ in this paper are derived from the two commonly used concepts in computer science "Longest common substring (LCS)" [44] and "Shortest common supersequence (SCS)" [45].

Table 1. Notations for the proposed similarity model

A	A set of assembly hierarchies
u, v	Two different assembly hierarchies $u \in A$ and $v \in A$
Ω	$\Omega = \{\mathrm{A, B, C, D, \dots}\}$ common components in hierarchy u and hierarchy v
e	Common component e, $e \in \Omega$

Θ	$\Theta = \{1, 2, 3, 4 \ldots\}$ common tasks in hierarchy u and hierarchy v
t	Common assembly task t, $t \in \Theta$
$LCMF_{u,v,e}$	The longest common material flow of component e between hierarchies u and v
$SJMF_{u,v,e}$	The shortest joint material flow of component e between hierarchies u and v
$NOTL_{u,v,e}$	Number of the assembly tasks in $LCMF_{u,v,e}$
$NOBT_{u,v,e}$	Number of bypassing tasks of component e calculated by $LCMF_{u,v,e}$ and $SJMF_{u,v,e}$
$NOIT_{u,v,e}$	Number of end-idle tasks of component e calculated by $LCMF_{u,v,e}$ and $SJMF_{u,v,e}$
$SA_{t,u}$	The subassembly module created by task t of hierarchy u
SS_t	equals 1 if $SA_{t,u} = SA_{t,v}$, and equals 0 otherwise.

Eq. (1) conducts a comparison between the material flow of common components for the current and future product designs that are related to assembly system changes for reconfiguration when a task becomes idle, or an idle task becomes activated. An idle task is defined as the one that is no longer used (idle) when the assembly hierarchy changes. If the idle tasks are located at two ends of the material flow, they are defined as the end-idle tasks. If they are located in the middle of the material flow, they are defined as bypassing tasks. Fig. 3 compares the $SJMF$ of material flow 1 (i.e., 1→2→4→9→5) and flow 2 (i.e., 1→2→5→9→8) that an assembly component goes through. One of the $SJMF$s of material flows 1 and 2 is 1→2→4→9→5→9→8. Assuming the material flow 1 is adopted, tasks 9 and 8 at the end of the $SJMF$ are not activated at current stage. For flow 1, they are defined as the end-idle tasks enclosed by circles. When material flow 1 changes to flow 2, the middle tasks 4 and 9 will be deactivated, and they are the bypassing tasks enclosed by squares. Based on the concepts of idle and bypassing machines, the $SJMF$

that gives the minimum number of both bypassing machines and idle machines should be selected, resulting in less effort for reconfiguration.

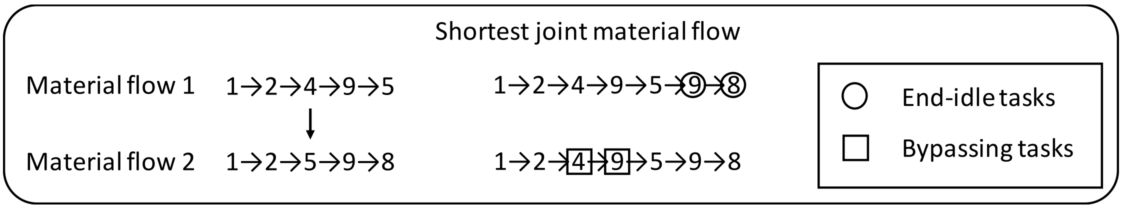


Fig. 3. Example of bypassing tasks and end-idle tasks

The multiplication coefficients in front of $NOTL_{u,v,e}$, $NOBT_{u,v,e}$, and $NOIT_{u,v,e}$ in Eq. (1) are determined by the number of material transfer routes (e.g., conveyor) that should be modified for future product designs. These coefficients can be changed when different conveyor systems or configurations are selected. In this paper, it is assumed that each task is associated with at least one transfer route for material input and one for output. And if the task changes, both transfer routes need to be modified for reconfiguration. As shown in Fig. 4, for the bypass tasks 4 and 9, one more route is needed to connect tasks 2 and 5, indicating that four transfer routes (Please refer to the dashed arrows under “Bypassing task”) should be modified for reconfiguration. For the end-idle tasks 9 and 8, two transfer routes (the dashed arrows under “End-idle task”) should be modified. The value of $SC_{u,v,e}$ should fall within [0,1].

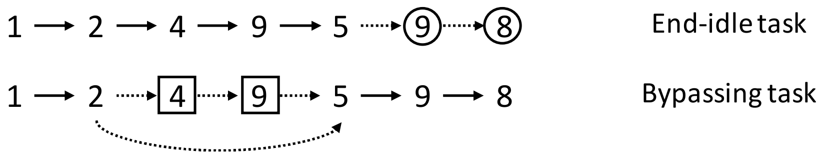


Fig. 4. Illustration of potential reconfiguration effort in Fig. 3

When there are no bypass tasks and end-idle tasks, the similarity between the two hierarchies is 1. Thus, the *material flow similarity* between two assembly hierarchies is given as:

$$SC_{u,v} = \sum_{i\in\Omega} SC_{u,v,i} \cdot \omega_i \; \forall u \in A, v \in A, \tag{3}$$

where ω_i is the weight value of each common component in space Ω. For example, the weight value could be proportional to the mass weight or the volume of the component, and $\sum \omega_i = 1$.

The *subassembly planning similarity* is defined as

$$SS_{u,v} = \frac{\sum_{t\in\Theta} SS_t}{\dim(\Theta)} \; \forall u \in A, v \in A, \tag{4}$$

which falls within the range [0,1]. $\dim(\Theta)$ is the total number of common tasks in both hierarchies u and v. Eq. (4) reflects the percentage of the tasks that generate the same subassemblies between two assembly hierarchies.

The rationale of the similarity model is that the higher similarity between two assembly hierarchies for two products usually indicates similar subassembly planning and material flow, and thus less reconfiguration effort/cost. The higher subassembly planning similarity ($SS_{u,v}$) means lower reconfiguration effort in *machine adjustment and less investment in labor skill training*; while the higher material flow similarity ($SC_{u,v}$) usually indicates less change on *materials transfer among stations and raw materials feeding*.

An example of the similarity model

Figure 5 shows an example of the product evolution from design (1) to design (2), where the nodes represent the components and the numbers above the lines connecting nodes represent assembly tasks. The assembly hierarchy involved in design (1) is given in

Fig. 6.a, and there are two candidate assembly hierarchies for product design (2) as shown in Figs. 6.b and 6.c, respectively. The subassembly planning (subassembly modules generated by different assembly tasks) and material flows for all the common components are given in Tables 2 and 3.

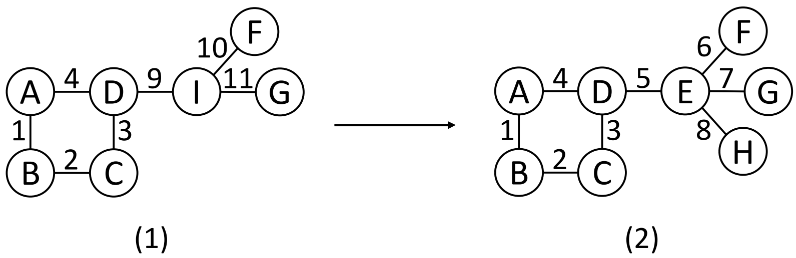


Fig. 5. A simple product evolution from design (1) to (2)

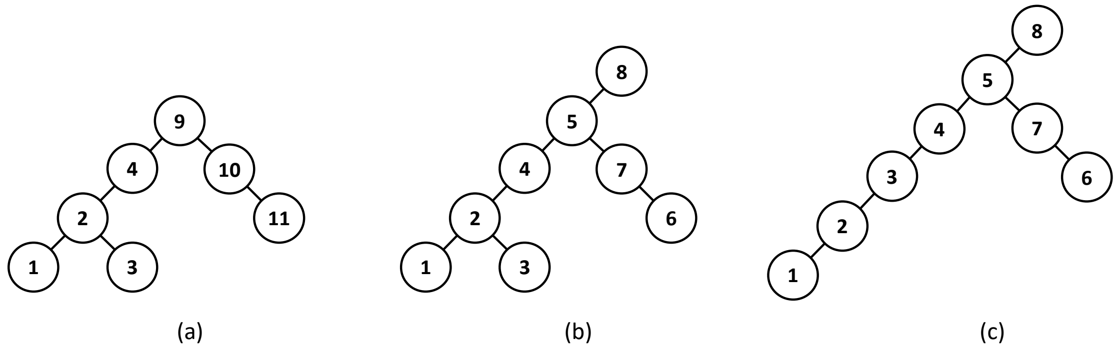


Fig. 6. Assembly hierarchies for product design (1) and (2) in Fig. 5

By comparing Hierarchy (a) for design (1) and Hierarchy (b) for design (2), it can be found that the subassembly modules created by the common tasks 1, 2, 3, and 4 are the same, meaning $SS_{u,v} = 1$, and therefore the machine adjustments and labor training for these tasks are not needed. For the material flow, the similarity $SC_{u,v}$ between Hierarchies (a) and (b) is 36%, which is higher than that (25%) between Hierarchies (a) and (c). Thus, the reconfiguration from (a) to (b) requires less effort compared with the reconfiguration from (a) to (c).

Table 2. Subassembly modules created by different planning in Fig. 6

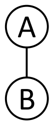
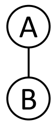
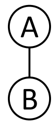
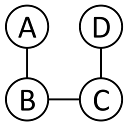
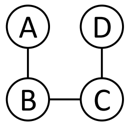
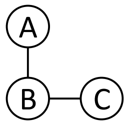
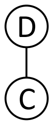
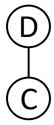
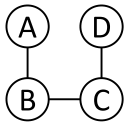
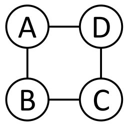
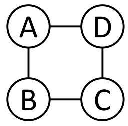
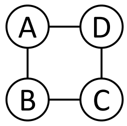
Common tasks	Hierarchy (a) for design (1)	Hierarchy (b) for design (2)	Hierarchy (c) for design (2)
1			
2			
3			
4			

Table 3. Material flow of common components among assembly tasks in Fig. 6

Common Components	Hierarchy (a) for design (1)	Hierarchy (b) for design (2)	Hierarchy (c) for design (2)
A	1→2→4→9	1→2→4→5→8	1→2→3→4→5→8
B	1→2→4→9	1→2→4→5→8	1→2→3→4→5→8
C	3→2→4→9	3→2→4→5→8	2→3→4→5→8
D	3→2→4→9	3→2→4→5→8	3→4→5→8
F	10→9	6→7→5→8	6→7→5→8
G	11→10→9	7→5→8	7→5→8

2.3 Assembly System Configuration for Reconfigurability

Based on the proposed similarity model, a concurrent optimization model can be formulated as follows.

Index sets of assembly tasks, stations, and product generation:

$T = \{1, \dots, n\}$ Set of all assembly tasks

$S = \{1, \dots, m\}$ Set of stations

$G = \{0, \dots, k\}$	Set of different product designs in the current and future generations. The value 0 stands for the current design and $1, \dots, k$ are the candidate designs for the future generation.

Decision variables for assembly system configuration and assembly hierarchy:

Task-station Assignment:

$$x_{t,s,g} = \begin{cases} 1, & \text{if task } t \text{ is assigned to station } s \text{ for candidate design } g \\ 0, & \text{otherwise} \end{cases}$$

The number of machines at station s for candidate design g:

$y_{s,g} \in N$ (Positive integer)

Task precedence:

$$w_{t_1,t_2,g} = \begin{cases} 1, & \text{if task } t_1 \text{ precedes } t_2 \text{ in generation } g \\ 0, & \text{otherwise} \end{cases}, t_1, t_2 \in T, s \in S, g \in G \,.$$

These decision variables are developed to determine (1) the hierarchical relationships of material flows among different assembly tasks and (2) the assignment of these tasks to a certain quantity of stations for the current and future generations, respectively. Different from prior research on line balancing, the task precedence is unknown and should be optimized to determine the assembly task hierarchy.

Parameters: The following parameters are related to the assembly process setup and practical constraints, and they should be predetermined.

$P_t, \;\; t \in T$	Processing time of assembly task *t*
Q	The length of production period for the new product design
$c_g, \;\; g \in G$	Assembly system's cycle time for each product design
$M_s, \;\; s \in S$	The maximum number of machines at each station

$z^{\mathrm{A}}_{t_1,t_2}$, $t_1, t_2 \in T$	Equals 1 if t_1 and t_2 must be assigned to the same station, and equals 0 otherwise
$z^{\mathrm{D}}_{t_1,t_2}$, $t_1, t_2 \in T$	Equals 1 if t_1 and t_2 cannot be assigned to the same station, and equals 0 otherwise
$w^{0}_{t_1,t_2}$, $t_1, t_2 \in T$	Known precedence required by practical constraints. They are the same for all product generations. Equals 1 if t_1 must precede t_2, and equals 0 otherwise
$f_{t,g}$, $t \in T, g \in G$	Equals 1 if t is applied in g, and equals 0 otherwise
c^{0}_{g}	Upper boundary for production cycle time
$L_\alpha, L_\beta, L_\gamma$	Estimated labor time needed for task-station reassignment (L_α), transfer route modification (L_β), and machine/labor adjustment (L_γ)

Objective function: The objective is to maximize the overall profit of the product designs, considering potential product evolutions with uncertainty considering the worst-case scenario. Since sometimes it is impractical to estimate the real dollar value of the profit, equivalent measurement indices are introduced in this paper. Specifically, we consider two index parts that are positively and negatively related to the profit, respectively. For the positively related part, we propose an index (production period/cycle time), which increases as the production increases. For the negatively related part, we introduce the indices that reflect three different aspects of reconfiguration cost ($L_\alpha, L_\beta, L_\gamma$). The increased values of these indices reflect higher reconfiguration cost and thus lower profit.

The worst-case consideration ***does not*** aim to replace the existing formulation in [37] based on a given probability of evolution path. Sometimes, the value of the probability of evolution path is impractical to obtain. Thus, it is critical to find a design and reconfiguration planning robust to the worst scenario. The mathematical representation of the objective is

$$\max\min\left(\frac{Q}{\mathrm{c}_g}-\frac{L_\alpha}{Q}R\left(x_{t,\mathrm{s},g},x_{t,\mathrm{s},0}\right)-\frac{L_\beta}{Q}SC_{u,v}-\frac{L_\gamma}{Q}SS_{u,v}\right) \tag{5}$$

where $\mathrm{c}_g=\max(\sum_{t\in\mathrm{T}}P_t x_{t,s,g}f_{t,g})$, the term $\sum_{t\in\mathrm{T}}P_t x_{t,s,g}f_{t,g}$ calculates the cycle time of each station for different product designs, and max function finds the bottleneck station (that has the longest cycle time among all stations) for the production system corresponding to each product design. Q/c_g reflects the impacts of the productivity during a certain production period Q on the decisions, $R(x)$ is the percentage of the tasks that need task-station reassignments as determined by decision variables $x_{t,\mathrm{s},g}$, and $SC_{u,v}$ and $SS_{u,v}$ are the similarity model proposed in Eqs. (3)-(4), where u and v can be obtained from decision variables $z_{t_1,t_2,g}$. The L/Q's with three subscripts $(L_\alpha,L_\beta,L_\gamma)$ in the objective function indicate the influence of the three types of reconfiguration cost on productivity. A large production period Q can make the value of L/Q small, indicating that the reconfiguration cost has a weak impact on the overall profit and vice versa. This formulation can change the emphasis on task-station reassignments, transfer route modification, or machine/labor adjustment in the objective function by setting different values of $L_\alpha,L_\beta,L_\gamma$. The min objective is to identify the product design leading to the lowest profit, and the $maxmin$ objective aims to maximize the worst-case profit, which refers to the scenario when the future product design leading to the lowest profit is selected (i.e., the worst decision is made). The proposed framework can be implemented complementarily with the probability-based formulation, depending on the availability of the probability data.

Constraints: Common constraints that reflect practical considerations and assumptions are as follows, i.e.,

$$\sum_{s\in S} x_{t,s,g} = 1 \ \forall t \in T, g \in G \tag{6}$$

$$z^{\mathrm{A}}_{t_1,t_2} - 1 \le x_{t_1,s,g} - x_{t_2,s,g} \le 1 - z^{\mathrm{A}}_{t_1,t_2} \ \ \forall t_1, t_2 \in T, t_1 \ne t_2, s \in S, g \in G \tag{7}$$

$$x_{t_1,s,g} + x_{t_2,s,g} \le 2 - z^{\mathrm{D}}_{t_1,t_2} \ \ \forall t_1, t_2 \in T,\ t_1 \ne t_2, s \in S, g \in G \tag{8}$$

$$w^0_{t_1,t_2} \le w_{t_1,t_2,g} \ \ \forall t_1, t_2 \in T,\ t_1 \ne t_2, g \in G \tag{9}$$

$$y_{s,g} \le M_s \ \ \forall s \in \mathrm{S}, g \in \mathrm{G} \tag{10}$$

Constraint set (6) follows a convention that one task is assigned to only one station (that can include multiple identical machines) for one product generation (if applicable). Constraint sets (7) and (8) are the zoning constraints, according to which some tasks must be assigned to the same station, whereas certain tasks cannot be assigned to the same station. Constraint set (9) is the precedence constraint. Constraint set (10) defines the maximum number of machines in each station.

Remark: A revision of the formulation. If the product designs in a certain evolution plan are less similar compared with that in other candidate evolution plans, this evolution plan will dominate the decisions by significantly increasing the reconfiguration cost in the objective function since the less similar plan can lead to the high reconfiguration effort. The selected configuration can be too conservative and may negatively affect the potential profits if other product evolutions are selected in the future. To deal with such a drawback, we can remove the productivity in the objective function and only minimize the reconfiguration cost. Then an additional constraint can be added to impose an upper bound to the cycle time so that the productivity meets the demand. Such mathematical reformulation will effectively prevent generating too conservative solutions with high reconfiguration cost, i.e.,

$$\min \sum_{g\in G} \left(\frac{L_\alpha}{Q} R\left(x_{t,s,g}, x_{t,s,0}\right) + \frac{L_\beta}{Q} SC_{u,v} + \frac{L_\gamma}{Q} SS_{u,v}\right), \tag{11}$$

$$\sum_{t\in T} P_t x_{t,s,g} f_{t,g} \leq c_g^0 \cdot y_{s,g} \quad \forall s \in S, g \in G. \tag{12}$$

3. SOLUTION PROCEDURE FOR ASSEMBLY SYSTEM CONFIGURATION

This paper customizes the EA to the formulated assembly system configuration problem, which usually involves a large search space for solutions, to obtain good-quality solutions within reasonable computational time.

3.1 Encoding for representing the assembly system configuration

Solutions to the assembly system design problem can be represented by integer chromosomes. Every chromosome consists of two parts including task sequence and task-station assignments. For the task sequence, the value of a gene (i.e., allele) marks a task labeled by an integer number. The location of a gene (i.e., locus) represents its position in the sequence. A task-to-station indicator represents the task-station assignment. For example, Fig. 7 shows six genes in the task sequence part (Left) for six tasks. Arrows can be inserted into seven positions (shown as black dots) along this gene sequence to separate tasks and assign the separated tasks to stations. A task-to-station indicator specifies the positions where the arrows should be inserted. In this example, the task-to-station indicator (Right) has four genes, assigning these tasks to three stations (as indicated by four arrows). If there are empty genes left in the chromosome, zeroes will be used as their alleles. This chromosome representation can be applied to each product evolution separately in the EA algorithm.

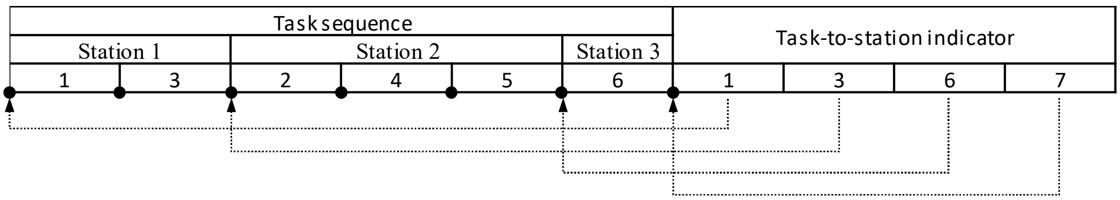


Fig. 7. An example chromosome representation for 6-task and 3-station problem

3.2 Initial generation of assembly system configurations

To improve the search efficiency, this paper develops a heuristic method that generates the initial population for the EA algorithm in three stages. First, a joint-liaison graph of the product designs (including the current and future designs) is constructed to generate a joint assembly hierarchy including the information on the subassembly planning and the task precedence relationships for all product designs. In the second stage, a heuristic approach is used to assign the assembly tasks to the earliest *available* station successively. The term “available station” refers to the station that can still be assigned with a new task so that the cycle time does not exceed the upper-bound of cycle time for all product designs at each station. In the third stage, the initial solutions to the assembly system designs for current and future products are obtained from the result, generated in the second stage, by following the same task-station assignments and the task sequences.

For example, assume that there is one product design of the current generation, and there are two future evolution scenarios. The current product design involves assembly tasks 1, 2, 4, and 6. The two future evolution scenarios involve tasks 1, 2, 3, 5 and tasks 1, 2, 3, 4, 5, 6, respectively. Given the joint-precedence relationship among these tasks for all product designs in Fig. 8, one candidate task-station assignment generated by the earliest-available station method is given in Table 4. Constrained by the task-station assignments and assembly sequence in Table 4 for all product designs, one initial solution for the EA algorithm is generated as shown in Table 5.

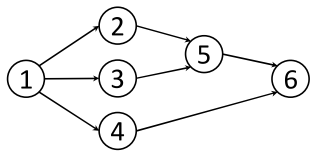


Fig. 8. Joint-precedence diagram for one product design and two future evolutions

Table 4. One candidate task-station assignment based on Fig. 8

Station 1	Station 2	Station 3
Task 1, 3	Task 2, 4, 5	Task 6

Table 5. Example of initial solutions based on Fig. 8

Product design	Task sequence						Task-to-station indicator			
Current	1	2	4	6	0	0	1	2	4	5
Evolution 1	1	3	2	5	0	0	1	3	5	5
Evolution 2	1	3	2	4	5	6	1	3	6	7

3.3 Fitness function and selection operators

The fitness function for the EA is the objective function Eq. (5). It consists of four parts including the productivity, reconfiguration effort due to task-station assignment/reassignment, reconfiguration effort due to the change of subassembly planning, and reconfiguration effort due to the change of material flow.

There are two types of selection operators, which are generally applied in canonical EA algorithms including (1) parent selection operator, which allows for identifying the parent chromosomes from the population that participate in the EA operations and (2) offspring selection operator, which allows for identifying the offspring chromosomes that survive in a given EA generation and will become candidate parents in the next EA generation. A stochastic universal sampling mechanism [31] will be used by the parent selection operator that provides higher chances for the chromosomes with larger fitness values to be selected as parents. As for the offspring selection, this study relies on a generational offspring selection scheme, according to which all the offspring chromosomes generated as a result of the EA operations (to be discussed in Section 3.4) can survive and become the candidate parents in the next generation. The generational offspring selection scheme has been widely used in Genetic Programming, proposed by Koza [46], and canonical Genetic Algorithms, developed by Holland [47].

3.4 EA operations

This section illustrates the two operators that are used in this study, i.e., crossover and mutation.

3.5.1 Crossover

Selection of an appropriate crossover operator depends on the chromosome representation, which was adopted for encoding the solutions to the problem of interest. The partially mapped, order, and cycle crossover operators have been generally deployed for the integer chromosome representation [47]. However, the aforementioned crossover operators may cause a complex infeasibility, which is associated with the violation of the task precedence constraints. Repairing such infeasibility may significantly increase the computational complexity of the developed EA algorithm. Therefore, a customized crossover operator was developed in order to prevent generation of infeasible individuals throughout the evolution of the algorithm. First, the crossover operator randomly selects two parents from the population. The probability of a given parent to participate in the crossover operation is determined by the crossover probability parameter of the EA algorithm. Second, it generates two random cut points and directly copies the genes, which are outside the two cut points, from the first parent to its offspring. Third, it copies the genes between the two cut points again from the first parent, following the sequence of those genes from the second parent. The crossover operation is illustrated in Fig. 9. Genes outside the cut points for Offspring 1 are directly copied from Parent 1 (1-4 in the front and 5 at the end). The task numbers between the cut points are (6-2-3) in Parent 1, and they are rearranged to (6-3-2) in Offspring 1 following their sequences in Parent 2. Thus, the final representation of Offspring 1 becomes (1-4-6-3-2-5). The second offspring chromosome is generated from the two parents in a similar way, except that the two parents are swapped. It should be noted that the crossover operator is applied to the task sequence part of the chromosome in this algorithm, while the task-to-station indicator part remains unchanged. By deploying this crossover operation, the precedence relationship among all the tasks can be satisfied, i.e., no infeasible solution that violates the precedence can be generated. Prevention of the infeasible solutions is critical since they may negatively affect the performance of the EA algorithm and will require the development of a repairing operator to make the solution feasible, thus increasing the computational complexity.

Cut Point Cut Point

	Task sequence						Task to station indicator			
Parent 1	1	4	6	2	3	5	1	2	5	7
Parent 2	4	6	5	3	2	1	1	3	6	7
Offspring 1	1	4	6	3	2	5	1	2	5	7
Offspring 2	4	6	2	3	5	1	1	3	6	7

Fig. 9. An illustration of the proposed crossover operator (change the task sequence).

3.5.2 Mutation

Based on preliminary computational experiments, it was found that application of the mutation operator to the chromosome portion representing the task sequence would create a significant number of infeasible mutated offspring chromosomes. The infeasibility of the mutated offspring chromosomes stems from the violation of the task precedence constraints. In order to prevent generation of infeasible individuals throughout the evolution of the algorithm, the mutation operator was not applied to the chromosome portion that represents the task sequence. However, the task sequences are being altered within the developed EA algorithm by applying the crossover operator, thereby maintaining the diversity of the population throughout the evolution of the algorithm. The floating-point mutation operator will be used to change task-station assignments. The mutation operator is applied in the task-to-station indicator part of the chromosome, while the task sequence part remains unchanged. When the task-station assignments are modified, it is necessary to change the arrow-inserting positions (black dots in Fig. 7) that separate the tasks. The arrow-inserting positions as controlled by the task-to-station indicator can be changed randomly, except that the first and the last arrows do not move. The probability that a given gene of the chromosome undergoes a mutation operation is determined by the mutation probability parameter of the EA algorithm. After the mutation operation, the algorithm evaluates the new chromosome and decides whether new machines should be added to the stations depending on the fitness value of the mutated offspring chromosome. An example is shown in Fig. 10 for a chromosome with one row. For a product design with several product evolutions, i.e., the chromosome with several rows, it is necessary to apply the mutation operator to each row of the offspring (generated by the crossover operation) separately.

	Task sequence						Task to station indicator			
Before mutation	1	4	6	2	3	5	1	2	5	7
After mutation	1	4	6	2	3	5	1	3	6	7

Fig. 10. An illustration of the mutation operator (changing the task-station indicator)

4. CASE STUDY

This section presents a case study motivated by *real-world* scenarios to show how the preference of productivity vs. reconfigurability affects assembly system configuration design under uncertain product evolution.

Figure 11 shows a product design and three possible future designs along with the corresponding joint liaison graph. The design of assembly system configuration should consider beyond the current product design by incorporating uncertain production evolutions to improve system reconfigurability. However, the probabilities of future products are unknown and detailed data to estimate the system reconfiguration cost are not given. There are thirteen different assembly tasks and ten components in total. The processing times (min) of all assembly tasks are presented in Table 6. By considering the three possible product evolutions, the problem is to determine the appropriate subassembly planning and task-station assignments that can potentially reduce the future reconfiguration effort while ensuring the productivity requirements.

Table 6. Task processing times in Case 1

Task	1	2	3	4	5	6	7	8	9	10	11	12	13
Time (min)	0.67	1.13	1.52	0.89	1.37	1.01	1.57	1.39	1.90	0.37	1.33	0.88	1.05

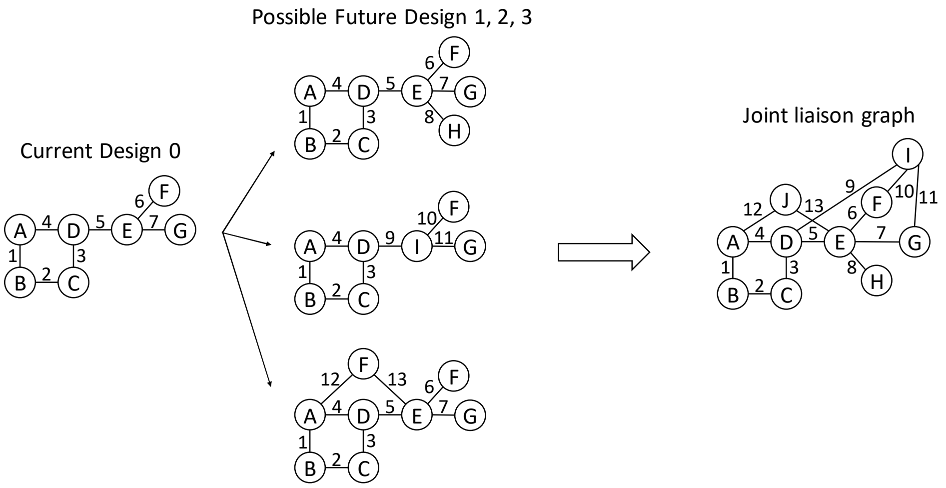


Fig. 11. Existing product design with three future product evolutions

The precedence relationship is given in Fig. 12. Task 5 cannot be processed until tasks 1, 2, 3, and 4 are finished. Meanwhile, task 9 can be operated after tasks 1, 2, and 4 are completed. Tasks 12 and 13 must be completed after task 5.

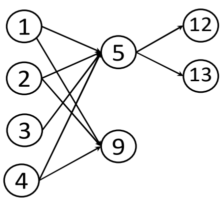


Fig. 12. Joint precedence diagram

Not all the tasks must have precedence relationships. In this example, tasks 6, 7, and 8 are to attach the part to component E. There are no constraints on the precedence among these tasks. They can either be operated before or after task 5. The same condition applies to the assembly tasks 10 and 11. With such weak constraints in precedence relationship, the search space for the system design problem can be huge when the number of assembly tasks increases. By employing an algorithm to enumerate all assembly hierarchies in [2], there are 1,791,648 assembly hierarchies generated for the joint-liaison in Fig. 13 based on the precedence relationship in Fig. 14. The potential search space for all the decisions is even larger. The advantage of using the EA is to obtain feasible solutions of good quality within reasonable computational time.

The objective function considers four aspects including productivity, task recurrence, material flow similarity, and sub-assembly similarity to characterize the reconfiguration effort. Manufacturers may have a different preference considering productivity vs. system reconfigurability. This case study discusses the assembly system configuration design under three scenarios that place different preference on these aspects in the objective function. Scenario 1 considers the long-period (Q) of production so that the impact of reconfiguration cost is less significant. Scenario 2 places more emphasis on the reduction of task-station reassignments that cause high reconfiguration effort, and scenario 3 considers the short-term production given the labor time for task-station reassignment, material transfer route modification, and machine/labor adjustment of the reconfiguration. Table 7 summarizes the values of L/Q for the three scenarios.

Table 7. Impact of reconfiguration on productivity L/Q for three scenarios

Impact of reconfiguration	Scenario 1	Scenario 2	Scenario 3
L_α/Q	0.01	100	0.2
L_β/Q	0.01	1	0.6
L_γ/Q	0.01	1	0.5

The following values were set for the EA parameters after running a parameter tuning analysis based on the methodology described in [48]. The population size is 20, the crossover rate is 90%, and the mutation rate is 20%. The algorithm will be terminated after 500 generations. Since the developed EA is stochastic, multiple replications are required to estimate the average objective and computational time values. In this study, a total of 100 of EA replications are performed for each scenario.

Scenario 1:

In this scenario, the productivity is far more significant than reconfigurability because (1) the impact of the reconfiguration due to task-station reassignments or the change of subassembly planning on the productivity or decision is relatively low and (2) the manufacturer requires short cycle time and high productivity for long-term production. The outcome from EA is given in Fig. 13. Three different future reconfiguration plans are provided. The average of the cycle times determined by the bottleneck stations that are corresponding to the three reconfiguration plans is 2.6 min, indicating the potential throughput. The results show that the task-station assignments frequently change, indicating high reconfiguration cost, while the assembly system has relatively lower cycle time and higher productivity.

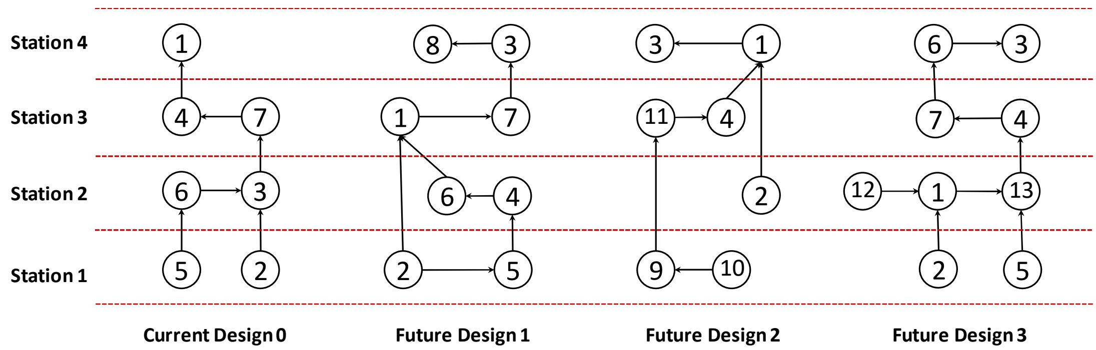


Fig. 13. Assembly system reconfiguration design under scenario 1

Scenario 2:

The reconfiguration cost due to task-station reassignments becomes very significant for the assembly system under this scenario, which is similar to Ko et al. [40]. When the consideration of task recurrence dominates the objective, the results suggested by the developed EA algorithm are given in Fig. 14.

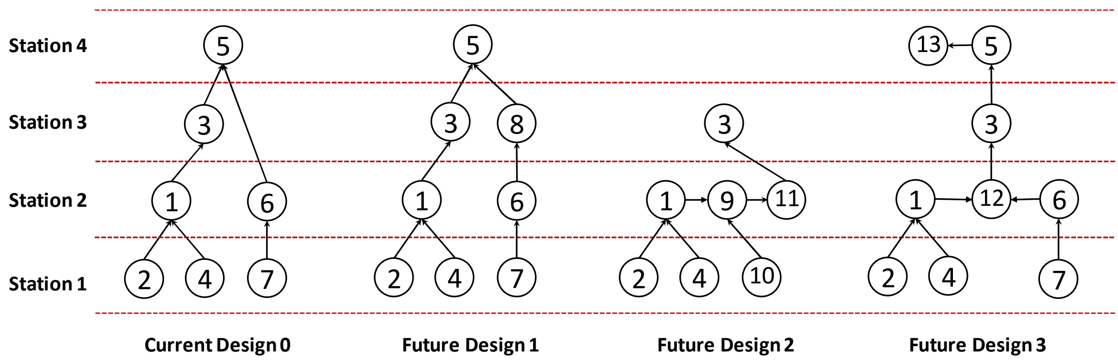


Fig. 14. Assembly system reconfiguration planning for scenario 2

Similar to [40], the results show that all the common tasks of different product designs remain in the same station to avoid the reconfiguration cost. However, there are two fundamental differences from [40] *under scenario 2* including

- The evolution probability of different product designs was assumed in [40] whereas the proposed method does not require this information. The solution considering the worst-case scenario in this paper can be more conservative than [40]; however, it provides a design guideline for assembly system configuration when the probability data are lacking. When more information is available to characterize the uncertainty, the probability-based method can be implemented to reach an aggressive decision with controlled risks for high reconfiguration cost.

- The reconfiguration planning in [40] was given without detailed hierarchical relationships of the assembly tasks in the stations. For example, in station 2, there can be multiple assembly sequences/hierarchies among tasks 1, 9, and 11 given future design 2, leading to different subassembly planning. As shown in Fig. 14, the proposed algorithm can determine the assembly hierarchies among these tasks which supply detailed information on how to arrange these tasks on each station to maximize the subassembly similarity, in addition to the task-station assignments. The task-station assignments can be influenced by the subassembly planning, and therefore the results obtained from the proposed concurrent optimization can be different from the traditional line balancing methods.

In this scenario, each task is assigned to the same station for all current and future product designs, thus reducing the reconfiguration effort. However, the decisions dominated by the reconfigurability consideration may lower the productivity. The result in Fig. 14 shows that the average of the cycle times determined by the bottleneck stations is 3.8 min, which means the productivity is 46% lower than Scenario 1.

Scenario 3:

Under this scenario, no one single factor dominates the objective. The results are shown in Fig. 15, where the average of the cycle times determined by the bottleneck stations is 2.87 min. While the assignments of most tasks to stations do not change for all the product designs, station assignment for tasks 1 and 3 are changed. Although this reassignment somehow increases the reconfiguration cost associated with task recurrence, the overall reconfiguration effort is reduced in this case since all the tasks for different product designs have the same subassembly planning, potentially reducing the machine adjustments and labor training. In addition, the expectation of cycle time (2.87 min) of the proposed assembly system reconfiguration planning becomes much shorter than that under scenario 2 (3.8 min).

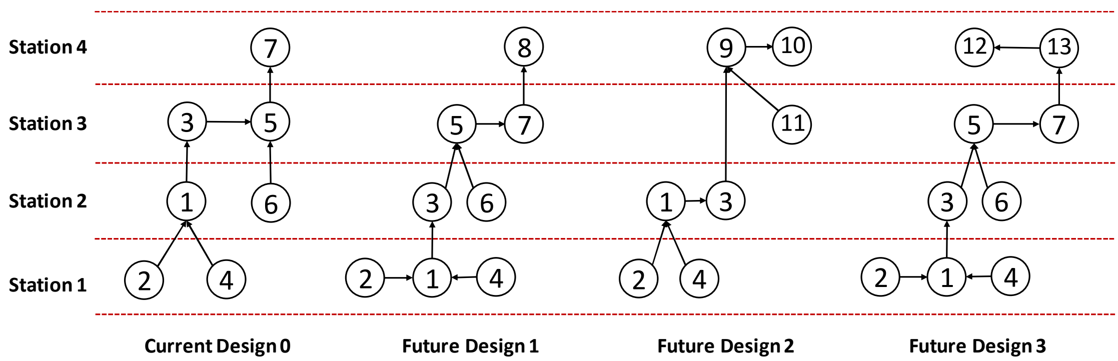


Fig. 15. Assembly system reconfiguration planning for scenario 3

Figure 16 shows the convergence of fitness function vs. EA solution generations under the three scenarios. The convergence patterns demonstrate that that developed EA algorithm was able to move efficiently along the search space and identify the promising domains (containing the solutions that have the fitness values close to the fineness values of the best solution discovered at convergence) within ≈100 generations.

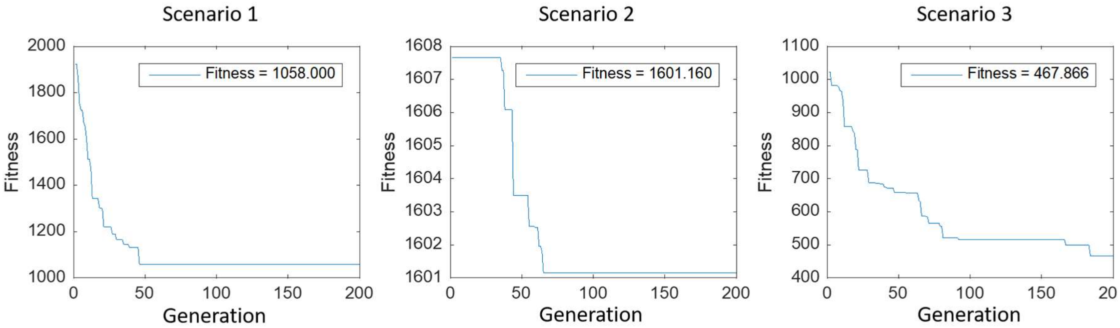


Fig. 16. Convergence pattern for 3 scenarios.

5. CONCLUSIONS

Design of the assembly system configuration jointly determines assembly process planning, task-machine assignments, and the logical material flow among stations. It is a critical step in the early stage of assembly system design when launching new product lines. The assembly system configuration should be flexible to cope with fast product development due to dynamic market demands. As such, the system configuration design should consider the generational evolution of future product designs to reduce potential reconfiguration effort. A product line usually has multiple evolution paths with uncertainty, posing a significant challenge to assembly system configuration design, especially when the probability values of evolutionary scenarios are not available.

This paper proposes a method for optimizing the assembly system configuration considering the product evolution with uncertainty. The highlights of this paper can be summarized as follows, i.e.,

- *A new similarity model for estimating reconfiguration effort*. Different from prior reconfiguration cost modeling, the proposed similarity model considers the impact of material flow and subassembly planning on potential reconfiguration effort. The method does not require detailed data on reconfiguration cost for labor training and equipment re-setup and therefore is very suitable for the early stage of product development.

- *Concurrent decision-making for subassembly planning and task-station assignments*. Based on the similarity model, a mixed integer programming model is developed for the concurrent optimization of the subassembly planning (subassembly modules and logic material flows) and task-station assignments for both current and future products by considering the reconfiguration effort due to uncertain product evolution. Compared with the prior research on assembly system reconfiguration, the concurrent decision-making further incorporates the impacts of subassembly planning on assembly system configurations into the estimation of reconfiguration effort.
- *Considering production evolution uncertainty without probability data*. The worst case analysis is adopted to deal with the situation that the probability of the evolution path cannot be provided. The model includes effective mathematical representations to characterize complex system configurations and subassembly hierarchies. The method can be implemented complementarily with the traditional probability-based approach, depending on the availability of probability data.
- *Computation-feasible algorithm customized for concurrent design problems*. This paper also develops an Evolutionary Algorithm with chromosome representations and operators customized for the joint optimization of

subassembly planning and task-station assignments in the assembly system configuration design problem. To improve solution search efficiency, the paper further develops a liaison graph model to guide the generation of the initial solutions. This research can also provide guidance on future product designs to lower the reconfiguration cost of assembly systems.

A case study on three scenarios shows that the assembly system configurations for the current and future product designs are significantly affected by the requirements of productivity vs. reconfigurability.

REFERENCES

[1] Benkamoun, N., Huyet, A. L., and Kouiss, K., "*Reconfigurable assembly system configuration design approaches for product change*," Proc. Proceedings of 2013 International Conference on Industrial Engineering and Systems Management (IESM), pp. 1-8.
[2] Homem de Mello, L. S., and Sanderson, A. C., 1990, "*AND/OR graph representation of assembly plans*," Robotics and Automation, IEEE Transactions on, 6(2), pp. 188-199. DOI: 10.1109/70.54734.
[3] Baldwin, D. F., Abell, T. E., Lui, M.-C., De Fazio, T. L., and Whitney, D. E., 1991, "*An integrated computer aid for generating and evaluating assembly sequences for mechanical products*," IEEE transactions on robotics and automation, 7(1), pp. 78-94. DOI: 10.1109/70.68072.
[4] Jiang, Z., 2015, "*Hierarchy generation for designing assembly system for product with complex liaison and sub-assembly branches*," Master, Florida State University.
[5] Jiang, Z., and Wang, H., 2016, "*Assembly Hierarchy Generation for Assembly System Design for a Product Family*," ASME 2016 11th International Manufacturing Science and Engineering Conference, p. V002T004A024. DOI: 10.1115/MSEC2016-8704.
[6] Wang, Y., and Liu, J., 2013, "*Subassembly identification for assembly sequence planning*," The International Journal of Advanced Manufacturing Technology, 68(1), pp. 781-793. DOI: 10.1007/s00170-013-4799-y.
[7] Koren, Y., Heisel, U., Jovane, F., Moriwaki, T., Pritschow, G., Ulsoy, G., and Van Brussel, H., 1999, "*Reconfigurable manufacturing systems*," CIRP Annals-Manufacturing Technology, 48(2), pp. 527-540. DOI: 10.1016/S0007-8506(07)63232-6.
[8] Koren, Y., and Shpitalni, M., 2010, "*Design of reconfigurable manufacturing systems*," Journal of Manufacturing Systems, 29(4), pp. 130-141. DOI: http://dx.doi.org/10.1016/j.jmsy.2011.01.001.

[9] Rösiö, C., and Säfsten, K., 2013, "*Reconfigurable production system design – theoretical and practical challenges*," Journal of Manufacturing Technology Management, 24(7), pp. 998-1018. DOI: 10.1108/jmtm-02-2012-0021.
[10] Bi, Z. M., Lang, S. Y. T., Shen, W., and Wang, L., 2008, "*Reconfigurable manufacturing systems: the state of the art*," International Journal of Production Research, 46(4), pp. 967-992. DOI: 10.1080/00207540600905646.
[11] Putnik, G., Sluga, A., ElMaraghy, H., Teti, R., Koren, Y., Tolio, T., and Hon, B., 2013, "*Scalability in manufacturing systems design and operation: State-of-the-art and future developments roadmap*," CIRP Annals-Manufacturing Technology, 62(2), pp. 751-774. DOI: 10.1016/j.cirp.2013.05.002.
[12] Son, S.-Y., Lennon Olsen, T., and Yip-Hoi, D., 2001, "*An approach to scalability and line balancing for reconfigurable manufacturing systems*," Integrated manufacturing systems, 12(7), pp. 500-511. DOI: 10.1108/09576060110407815.
[13] Maler-Speredelozzi, V., Koren, Y., and Hu, S., 2003, "*Convertibility measures for manufacturing systems*," CIRP Annals-Manufacturing Technology, 52(1), pp. 367-370. DOI: 10.1016/S0007-8506(07)60603-9.
[14] Spicer, P., and Carlo, H. J., 2006, "*Integrating Reconfiguration Cost Into the Design of Multi-Period Scalable Reconfigurable Manufacturing Systems*," ASME Journal of Manufacturing Science and Engineering, 129(1), pp. 202-210. DOI: 10.1115/1.2383196.
[15] Koren, Y., Wang, W., and Gu, X., 2017, "*Value creation through design for scalability of reconfigurable manufacturing systems*," International Journal of Production Research, 55(5), pp. 1227-1242. DOI: 10.1080/00207543.2016.1145821.
[16] Wang, W., and Koren, Y., 2012, "*Scalability planning for reconfigurable manufacturing systems*," Journal of Manufacturing Systems, 31(2), pp. 83-91. DOI: https://doi.org/10.1016/j.jmsy.2011.11.001.
[17] Bryan, A., Hu, S. J., and Koren, Y., 2013, "*Assembly System Reconfiguration Planning*," ASME Journal of Manufacturing Science and Engineering, 135(4), pp. 041005-041005-041013. DOI: 10.1115/1.4024288.
[18] Zhang, B., Katinas, C., and Shin, Y. C., 2018, "*Robust Tool Wear Monitoring Using Systematic Feature Selection in Turning Processes With Consideration of Uncertainties*," Journal of Manufacturing Science and Engineering, 140(8), pp. 081010-081010-081012. DOI: 10.1115/1.4040267.
[19] Wang, Y., Mendis, G. P., Peng, S., and Sutherland, J. W., 2018, "*Component-Oriented Reassembly in Remanufacturing Systems: Managing Uncertainty and Satisfying Customer Needs*," Journal of Manufacturing Science and Engineering, 141(2), pp. 021005-021005-021012. DOI: 10.1115/1.4042150.
[20] Tapia, G., King, W., Johnson, L., Arroyave, R., Karaman, I., and Elwany, A., 2018, "*Uncertainty Propagation Analysis of Computational Models in Laser Powder Bed Fusion Additive Manufacturing Using Polynomial Chaos Expansions*," Journal of Manufacturing Science and Engineering, 140(12), pp. 121006-121006-121012. DOI: 10.1115/1.4041179.
[21] Ammar, O. B., Hnaien, F., Marian, H., and Dolgui, A., 2016, "*Optimization Approaches for Multi-level Assembly Systems Under Stochastic Lead Times*," Metaheuristics for Production Systems, E.-G. Talbi, F. Yalaoui, and L. Amodeo, eds., Springer International Publishing, Cham, pp. 93-107. DOI: 10.1007/978-3-319-23350-5_4.

[22] Hu, S. J., Zhu, X., Wang, H., and Koren, Y., 2008, "*Product variety and manufacturing complexity in assembly systems and supply chains*," CIRP Annals, 57(1), pp. 45-48. DOI: https://doi.org/10.1016/j.cirp.2008.03.138.
[23] Koren, Y., Gu, X., and Guo, W., 2018, "*Reconfigurable manufacturing systems: Principles, design, and future trends*," Frontiers of Mechanical Engineering, 13(2), pp. 121-136. DOI: 10.1007/s11465-018-0483-0.
[24] ElMaraghy, H., Samy, S., and Espinoza, V., "*A classification code for assembly systems*," Proc. 3rd CIRP Conference on Assembly Technologies and Systems, pp. 145-150.
[25] Hu, S. J., Ko, J., Weyand, L., ElMaraghy, H., Lien, T., Koren, Y., Bley, H., Chryssolouris, G., Nasr, N., and Shpitalni, M., 2011, "*Assembly system design and operations for product variety*," CIRP Annals-Manufacturing Technology, 60(2), pp. 715-733. DOI: 10.1016/j.cirp.2011.05.004.
[26] Lee, G. H., 1997, "*Reconfigurability consideration design of components and manufacturing systems*," The International Journal of Advanced Manufacturing Technology, 13(5), pp. 376-386. DOI: 10.1007/BF01178258.
[27] Youssef, A. M., and ElMaraghy, H. A., 2006, "*Assessment of manufacturing systems reconfiguration smoothness*," The International Journal of Advanced Manufacturing Technology, 30(1-2), pp. 174-193. DOI: 10.1007/s00170-005-0034-9.
[28] Jiao, J., Simpson, T. W., and Siddique, Z., 2007, "*Product family design and platform-based product development: a state-of-the-art review*," Journal of Intelligent Manufacturing, 18(1), pp. 5-29. DOI: 10.1007/s10845-007-0003-2.
[29] Gonzalez-Zugasti, J. P., Otto, K. N., and Baker, J. D., 2001, "*Assessing value in platformed product family design*," Research in Engineering Design, 13(1), pp. 30-41. DOI: 10.1007/s001630100001.
[30] Meyer, M. H., and Lehnerd, A. P., 1997, *The power of product platforms*, Simon and Schuster.
[31] Fellini, R., Kokkolaras, M., Papalambros, P. Y., and Perez-Duarte, A., 2002, "*Platform Selection Under Performance Loss Constraints in Optimal Design of Product Families*," (36223), pp. 613-621. DOI: 10.1115/DETC2002/DAC-34099.
[32] McAdams, D. A., and Wood, K. L., 2002, "*A Quantitative Similarity Metric for Design-by-Analogy*," Journal of Mechanical Design, 124(2), pp. 173-182. DOI: 10.1115/1.1475317.
[33] Thevenot, H. J., and Simpson, T. W., 2006, "*Commonality indices for assessing product families*," Product platform and product family design, Springer, pp. 107-129. DOI: 10.1007/0-387-29197-0_7.
[34] Bryan, A., Ko, J., Hu, S. J., and Koren, Y., 2007, "*Co-Evolution of Product Families and Assembly Systems*," CIRP Annals, 56(1), pp. 41-44. DOI: https://doi.org/10.1016/j.cirp.2007.05.012.
[35] AlGeddawy, T., and ElMaraghy, H., 2012, "*A Co-Evolution Model for Prediction and Synthesis of New Products and Manufacturing Systems*," Journal of Mechanical Design, 134(5), pp. 051008-051008-051012. DOI: 10.1115/1.4006439.
[36] Son, S.-Y., 2000, "*Design principles and methodologies for reconfigurable machining systems*," Dissertation, University of Michigan.

[37] Youssef, A. M., and ElMaraghy, H. A., 2007, "*Optimal configuration selection for reconfigurable manufacturing systems*," International Journal of Flexible Manufacturing Systems, 19(2), pp. 67-106. DOI: 10.1007/s10696-007-9020-x.
[38] Ye, H., and Liang, M., 2006, "*Simultaneous Modular Product Scheduling and Manufacturing Cell Reconfiguration Using a Genetic Algorithm*," ASME Journal of Manufacturing Science and Engineering, 128(4), pp. 984-995. DOI: 10.1115/1.2336261.
[39] Yuan, M., Deng, K., Chaovalitwongse, W. A., and Cheng, S., 2017, "*Multi-objective optimal scheduling of reconfigurable assembly line for cloud manufacturing*," Optimization Methods and Software, 32(3), pp. 581-593. DOI: 10.1080/10556788.2016.1230210.
[40] Ko, J., and Hu, S. J., 2009, "*Manufacturing System Design Considering Stochastic Product Evolution and Task Recurrence*," ASME Journal of Manufacturing Science and Engineering, 131(5), pp. 051012-051012. DOI: 10.1115/1.4000095.
[41] Ko, J., and Hu, S. J., 2007, "*Manufacturing system design considering generational product evolution and task recurrence*," Proceedings of the Asme International Conference on Manufacturing Science and Engineering - 2007(42908), pp. 815-826. DOI: 10.1115/MSEC2007-31090.
[42] Tao, F., Bi, L., Zuo, Y., and Nee, A. Y. C., 2017, "*A Cooperative Co-Evolutionary Algorithm for Large-Scale Process Planning With Energy Consideration*," Journal of Manufacturing Science and Engineering, 139(6), pp. 061016-061016-061011. DOI: 10.1115/1.4035960.
[43] Goyal, K. K., Jain, P., and Jain, M., 2013, "*A comprehensive approach to operation sequence similarity based part family formation in the reconfigurable manufacturing system*," International Journal of Production Research, 51(6), pp. 1762-1776. DOI: 10.1080/00207543.2012.701771.
[44] Gusfield, D., 1997, *Algorithms on strings, trees and sequences: computer science and computational biology*, Cambridge university press.
[45] Jiang, T., and Li, M., 1995, "*On the approximation of shortest common supersequences and longest common subsequences*," SIAM Journal on Computing, 24(5), pp. 1122-1139. DOI: 10.1137/S009753979223842X.
[46] Koza, J. R., 1994, "*Genetic programming as a means for programming computers by natural selection*," Statistics and computing, 4(2), pp. 87-112. DOI: 10.1007/BF00175355.
[47] Eiben, A. E., and Smith, J. E., 2015, "*Introduction to evolutionary computing*," Natural computing series. DOI: 10.1007/978-3-662-44874-8.
[48] Lima, E. B. d., Pappa, G. L., Almeida, J. M. d., Gonçalves, M. A., and Meira, W., "*Tuning Genetic Programming parameters with factorial designs*," Proc. IEEE Congress on Evolutionary Computation, pp. 1-8. DOI: 10.1109/CEC.2010.5586084.

ACKNOWLEDGMENT

The authors thank Prof. S. Jack Hu from the University of Michigan, for providing industry background related to assembly system design that motivates this research.

FUNDING

This research is partially supported by the National Science Foundation (NSF) grants CMMI-1744131 and HRD-1646897.

FIGURE CAPTIONS LIST

TABLE CAPTIONS LIST