Int J Adv Manuf Technol (2016) 83:99–115
DOI 10.1007/s00170-015-7565-5

ORIGINAL ARTICLE

A weighted assembly precedence graph for assembly sequence planning

Yong Wang[1] · **De Tian**[1]

Received: 17 February 2015 / Accepted: 5 July 2015 / Published online: 19 July 2015

Abstract Assembly sequence planning is one of the well-known combinatorial optimization problems in manufacturing. An assembly is often represented as an assembly relation graph or precedence graph. The traditional methods are used to generate a large number of feasible assembly sequences and then find the optimal sequence through evaluation. A lot of computation resources are needed. To reduce the complexity of assembly sequence planning, the assembly is converted into a weighted assembly precedence graph considering multiple assembly constraints, i.e., the qualitative and quantitative constraints. The vertices in the weighted precedence graph are the parts or components. The qualitative constraints including the topological and geometrical assembly constraints guarantee to derive the feasible assembly sequences. Some process constraints are also taken as the qualitative constraints. They are represented as the directed edges in the weighted assembly precedence graph. The other assembly constraints, such as the stable support, connector strength, changes of assembly directions, and tools and so forth, are quantified as indices to compute the cost of assembly relations with the fuzzy analytical hierarchy process. The costs are taken as the heuristic information to find the optimal or near-optimal assembly sequences. With the weighted assembly precedence graph, the search space of the optimal assembly sequence will be reduced. We design a minimum spanning tree-based algorithm to detect the optimal assembly sequence based on the weighted assembly precedence graph. The optimal assembly sequences are found in $O(n^3)$ computation time, where n is the number of the discrete parts.

Keywords Assembly sequence planning · Weighted assembly precedence graph · Fuzzy analytical hierarchy process · Minimum spanning tree-based algorithm

1 Introduction

The research and development of complex products advocate the advanced design methods and the manufacturing technologies as well. Assembly technologies play an important role to connect the discrete parts into a whole product. Different from the automatic machining, assembly process is generally costly and time-consuming for most mechanical products. It is reported that the assembly occupies 20–50 % of the manufacturing time and approximately above 40 % of manufacturing cost are used for assembly [1]. For the special micro-electro-mechanical systems, it reaches 90 % of the manufacturing cost [2]. To improve the assembly efficiency and reduce the assembly cost, assembly planning is critical to produce the good assembly plans. It generally includes the demanding assembly sequences, the particular ways that each part or component is assembled [3]. The assembly plans heavily affect the assignments of assembly tasks and the related production resources including the engineers, energy and power, materials, work stations, tools, etc. [4]. Assembly sequence planning (ASP) generates the competitive assembly sequences as an important part of assembly plans [5], which bridges the product design and manufacturing. The assembly sequences indicate a series of assembly operations to assemble the independent parts orderly. The properties of products (relations between parts, geometry of parts, parts' materials and tolerance, etc.) and assembly resources (assembly line, equipments, tools, etc.) are

---

✉ Yong Wang
wangyyong100@126.com

[1] Renewable Energy School, North China Electric Power University, Beijing 102206, China

comprehensively considered to generate the optimal assembly sequence. The optimal assembly sequence is able to improve the assembly efficiency and reduce the assembly cost by rationally assigning the assembly resources considering assembly structures.

ASP is so important that it is regarded by many scholars in the fields of manufacturing [6–11]. However, it is not simple to tackle for complex products. In theory, ASP is one NP-complete problem [10, 12] under assembly constraints. The number of assembly sequences increase exponentially in proportion to the scale of assembly (the number of parts and assembly relationships). It is impossible to find the optimal assembly sequence by evaluating all of them. Once the assembly becomes complex, the current methods and algorithms are usually used to detect the near-optimal assembly sequences [12, 13]. In practice, the assembly process of the whole product is much more complex [14] than that of the single part machining. All assembly resources under multiple constraints are considered to find the optimal assembly sequence to lower the assembly cost and improve assembly efficiency.

Due to its theoretical and practical values, assembly sequence planning becomes one of the most active research topics in the CAAPP. Today, the optimal assembly sequence is usually generated with the computers (CAAPP). The subjects on ASP mainly focus on the assembly representation for ASP, the methodologies to generate the assembly sequences, and the evaluation of assembly sequences. The three aspects coordinate with each other to acquire the optimal assembly sequence. For example, the necessary assembly constraints are used to build the assembly representation models. Based on the assembly representation models, the reasoning methods and algorithms are selected to search the optimal assembly sequence meeting the demands of evaluation criteria. The efficiency of the reasoning methods and algorithms are various according to different assembly representation models. Once the assembly representation models and criteria are changed, the optimal or near-optimal assembly sequence will also be different.

ASP is a typical of discrete optimization problem in mathematics. The discrete topological structures are used to represent the assembly for ASP. The graphs, trees, and their variations are extensively used as assembly representation models [15, 16]. In the early stages, Bourjault [6] introduces the assembly liaison graph (actually an undirected graph) to denote the assembly. The "ask-answer" method is utilized to distinguish the assembly precedence between parts. The feasible assembly sequences are produced and represented as a directed assembly state graph. In the following, the ask-answer method is improved [17, 18] to reduce the times of ask and answer. The minimum number of questions proposed by engineers is reduced to $2n$ for an assembly with n parts or components. The assembly liaison graph only includes the contacts or connections between each pair of parts. The fundamental geometrical and process information are ignored. To generate the valid assembly sequences, the assembly precedence between parts must be pointed out by engineers. Otherwise, the machine is hard to deduce the feasible assembly sequences. In addition, the ask-answer method needs much labor for large scale of assembly. Recently, the assembly liaison graph is utilized for subassembly identification [19] to reduce the complexity of ASP. The sub-sequences of subassemblies are searched with the graph search algorithms and the nested partitions (NP) method is used to merge the sub-sequences into the whole assembly sequence. To support the automatic ASP, the geometrical mates and physical connections are added to the assembly relation (or liaison) graph [20]. Based on the assembly relation graph with geometrical constraints, the "cut-sets" method generates all of the feasible assembly sequences which are illustrated as an "And/Or" graph. The optimal assembly sequence is selected from these feasible sequences by evaluation. In theory, the number of cut-sets is the exponential function of the number of parts for the strong connected assembly (each part is connected to every other part). In practice, the strong connected assembly is the worst case. It seldom happens for most of the assemblies. The assembly process constraints are not taken into account in the assembly relation graph. Otherwise, the number of feasible assembly sequences will be reduced further. It is convenient to represent the assembly with an assembly relation graph. The topological structure of the assembly is clearly demonstrated. On the other hand, many other assembly constraints, such as the geometrical and process constraints [21], are seldom incorporated into the simple assembly relation graphs. With the simple graph model, the search space of the optimal assembly sequence is so large that the current methods are hard to accomplish the ASP of complex products.

To reduce the complexity of ASP, the assembly precedence is added to the undirected assembly relation graph model. Assembly precedence is taken as the strong qualitative assembly constraints to denote the assembly orders between parts. Several examples of qualitative assembly constraints in assembly process are the geometrical interference, assembly stability, accessibility, invisibility, etc. The assembly process will become impossible or difficult if it violates these qualitative assembly constraints. The assembly precedence between each pair of parts is usually represented as a directed edge [22]. The assembly precedence graph replaces the assembly relation graph to represent the assembly for ASP. In the disassembly precedence graphs [23, 24], the disassembly precedence is noted as an arrowed edge (or arc) which points from the dismantled parts to the non-dismantled parts. The assembly precedence considering the geometrical interference can be obtained by "yes-no" queries [6, 17, 18], from CAD draft [21, 25, 26] and 3D CAD models [14, 27]. The assembly precedence denoting the other qualitative assembly constraints will be derived from the CAD models or added by designers to reduce the assembly difficulty or cost. In general, the precedence graph is represented as BOOL matrices [25, 26, 28, 29] for algorithms to deduce the assembly sequences. With the assembly precedence graph, the search space of the optimal assembly sequence is decreased to some extent. Although the assembly precedence graph includes the qualitative constraints, the number of the feasible assembly sequences is still very large [30]. Given an assembly precedence graph with n parts, R is the number of ancestor parts, and r_i is the number of precedent edges leaving the ith ancestor part, the number of feasible assembly sequences N is computed as formula (1) [10].

$$N=\frac{n!}{\prod_{i=1}^{R}\left(r_i+1\right)} \tag{1}$$

If N is big and the number of directed edges r_i is small, the number of assembly sequences will be very large. It is time-consuming to find the optimal assembly sequence by evaluating them one by one.

Besides the popular assembly relation graph, And/Or graph, and assembly precedence graph, there are other assembly models for ASP. Zhou et al. [31] integrate the assembly relation graph and assembly process model together to support automatic ASP. Gu et al. [32] introduce the ordered binary decision diagram (OBDD) to substitute the And/Or graph and precedence graph. The storage of the assembly sequences is reduced. Banerjee et al. [33] improve the scene graph structure (a multiple hierarchy structure) into the behavior scene graph for virtual assembly sequence planning. The geometry, location, material properties, precedence rules between parts, motion parameters, etc. are described as a 4-tuples to prohibit the infeasible execution operations. Zhu et al. [13] formulate a transformed network flow model to represent the assembly process. The factors determining the assembly complexity, such as the selection of parts, tools, fixtures, and assembly procedures, are considered comprehensively. The entropy of choice is defined to measure the process complexity. Based on the network flow model, a dynamic programming method is designed to find the optimal assembly sequence. In the disassembly area, Lambert [23] uses the disassembly precedence graph to represent the disassembly process. The profits to dismantle the assembly liaisons are given first to evaluate the disassembly process. A greedy heuristics and an exact branch and bound algorithm are developed to search the optimal disassembly sequence. In literatures [13, 23], the time complexity of the two exact algorithms is the exponential function of the scale of the assembly. In the above research on automatic ASP, the methods usually include three stages. Firstly, the assembly is represented as an undirected or directed graph (or trees) or their variations. Secondly, the reasoning methods or algorithms are used to generate the feasible assembly sequences under the qualitative assembly constraints, such as the topological and geometrical constraints. Finally, these assembly sequences are evaluated in view of the assembly criteria to find the optimal assembly sequence [34, 35]. The assembly criteria to evaluate the assembly sequences are relevant to the assembly process constraints and usually represented as the assembly cost [36–38], time [34, 39], or complexity [13]. These methods of the three stages have an obvious shortcoming. The optimal assembly sequence is hardly found until all of the assembly sequences are traversed. If the parts are assembled one by one, the directed or undirected graph model cannot provide the heuristic information to detect the optimal sub-sequences. To find the optimal sub-sequences, the quantitative criteria must be used in the search process before a whole assembly sequence is obtained. In reference [13], the authors assign the entropy of assembly complexity to every operation. The optimal assembly sequence is found by evaluating the edges' entropy. Lambert [23] also attaches the profit to the assembly relations to select the next part with the maximum profit for disassembly. In the virtual disassembly, Behdad et al. [40] use the risk aversion to evaluate the disassembly operations. The disassembly sequence with the maximal value of the risk aversion is taken as the optimal disassembly sequence. These quantitative criteria play an important role to find the optimal sub-sequences until the optimal assembly sequence is acquired. Furthermore, a lot of non-optimal subsequences are neglected in the computation process so that the efficient of algorithms will be enhanced.

In this paper, we introduce a weighted assembly precedence graph model for ASP. Most of the assembly constraints for ASP can be integrated into the model. With the weighted assembly precedence graph, the optimal assembly sequence can be searched automatically with the algorithms. The assembly constraints are classified into the qualitative and quantitative constraints. The qualitative assembly constraints include the topological assembly relations among parts (or components) and geometrical interference between adjacent parts (or components). A portion of process constraints are also taken as the qualitative constraints to reduce the assembly cost or complexity, such as the constraints to facilitate the invisibility, accessibility, etc. The qualitative assembly constraints are represented as the directed (or undirected) edges in the weighted graph to denote the assembly precedence (or relations). The qualitative assembly constraints guarantee to produce the feasible assembly sequences. Most of them are not the optimal or near-optimal sequences. The other assembly process constraints are considered as the quantitative constraints, such as the stable support, reorientations and assembly equipment or tools, etc. They conclude the assembly efficiency, cost, or complexity of the assembly process. These quantitative constraints are defined with indices. The indices are used to compute the cost of assembly relations with the fuzzy analytical hierarchy process methods (FAHP), which represent the cost of assembly operations to complete the assembly relations. The costs are attached to the edges in the weighted assembly precedence graph. The edges' costs are taken as a kind of heuristic information to find the optimal assembly sub-sequences and sequence for algorithms.

After the qualitative and quantitative assembly constraints are integrated into the weighted assembly precedence graph model, the ASP becomes an asymmetrical TSP. The current efficient methods, e.g., the graph search algorithms, branch and bound, cutting planning and dynamic programming, etc., are able to resolve the ASP [41]. Comparing with the other assembly models, the weighted assembly precedence graph model has three merits: (1) Most of the assembly qualitative and quantitative constraints are able to integrate into the weighted assembly precedence graph. The quality of the assembly sequences is guaranteed due to the consideration of multiple constraints. (2) The weighted graph model is flexible to meet various assembly environments. The costs of assembly relations are computed according to multiple quantitative assembly constraints. The importance of the quantitative constraints can be adjusted by the engineers with the FAHP method. Once the assembly requirements are changed, a different weighted assembly precedence graph model will be computed. (3) The complexity of ASP will be reduced based on the weighted graph model. With the heuristic information on the edges, the algorithms do not need to check all of the feasible assembly sequences to find the optimal sequence. In addition, some formulae or inequalities can be constructed based on the edges' costs as constraints to reduce the search space. Most feasible sub-sequences and sequences will be neglected in the search process if they violate the formulae or inequalities [42]. The computation time of the efficient algorithms for ASP will be reduced.

The merit of the research is to present a weighted assembly precedence graph model for ASP. The necessary qualitative and quantitative assembly constraints are integrated into the graph model. Based on the comprehensive model, the ASP can be resolved with the current efficient methods. The structure of the paper is listed as below. In Section 1, the weighted assembly precedence graph model is briefly introduced. Section 2 focuses on the qualitative assembly constraints and their representation in the weighted assembly precedence graph. The quantitative assembly constraints to evaluate the operations to complete the assembly relations are summarized in Section 3. The indices of the qualitative constraints are defined and computed. The cost of assembly relations are computed with the FAHP methods and attach to the edges in the weighted assembly precedence graph. In Section 4, a minimum spanning tree-based (MST-based) algorithm is designed to search the optimal assembly sequence. Based on the weighted assembly precedence graph, the time complexity of the algorithm is $O(n^3)$. Two examples are demonstrated to show the advantage of the weighted assembly precedence graph in Section 5. The efficiency of the MST-based algorithm is analyzed and the experimental results are compared with those generated by the other methods. Finally, the method is concluded and the possible future work is given.

2 The weighted assembly precedence graph

Graph is often used to represent the topological structure of discrete objects, especially for the objects composed of many independent instances. For a weighted graph G with n vertices, it is often noted as $G=<V, E, W>$, where $V=\{v_1, v_2,\ldots, v_n\}$ are the vertex sets and $E=[e_{ij}]_{n\times n}$ are the edges matrix, v_i ($1\leq i\leq n$) is the vertex, and $e_{ij}(1\leq i, j\leq n)$ is the edge linking the vertices v_i and v_j. The weights matrix noted as $W=[w_{ij}]_{n\times n}$ are assigned to the corresponding edges $[e_{ij}]_{n\times n}$. For the directed graph, w_{ij} is generally unequal to w_{ji}. The directed weighted graph has many applications in science and engineering problems. For example, it is often used to represent the internet and transportation networks. The weights on the edges are the cost, distance, etc. between each pair of vertices. An assembly is just this kind of topological structure. It is connected with a group of discrete parts to reach the product function. It is convenient to represent an assembly as an undirected graph without weights. However, the search space of the optimal assembly sequence is so large that all of the exact algorithms cannot complete ASP within an acceptable compute time for complex assemblies. To reduce the complexity of ASP, it is advocated to utilize the weighted assembly precedence graph to represent the assembly. Due to the complexity of the assembly structure and assembly process, it is not an easy task to change an assembly into a weighted assembly precedence graph. Many assembly constraints must be handled properly and merged into the weighted assembly precedence graph. Therefore, it is rarely to see the weighted assembly graph model for ASP in the previous literatures.

Given an assembly with six parts, the weighted assembly precedence graph is shown in Fig. 1. The vertices noted with numbers are the parts or components and the edges (or arcs) between them are the qualitative assembly constraints, such as the assembly relations or assembly precedence. The assembly relations include the contacts, connections, or functional dependence between parts. The assembly relations note the

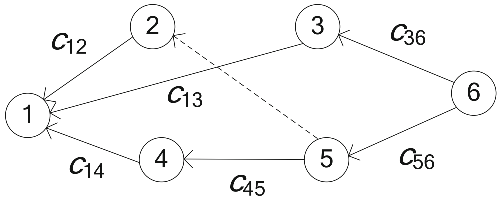


Fig. 1 The weighted assembly precedence graph of an assembly

topological assembly structure, and they do not represent the assembly orders of parts. They are represented as the real lines in the weighted assembly precedence graph. Assembly precedence is another kind of comprehensive qualitative assembly constraints. It concludes the assembly orders of parts. It is represented as an arrowed edge between two parts. The arrow point from the successor parts to the ancestor parts, which means the successor parts are assembled after the ancestor part. When the two parts have an assembly relation and one assembly precedence, the assembly precedence is noted as a real directed edge, such as edges e_{12} and e_{14} in Fig. 1. If the two parts do not have any assembly relations but only the assembly precedence, the assembly precedence is noted as a dashed directed edge, for example the dashed edge e_{25} in Fig. 1. The assembly precedence graph is intuitive for human whereas it is difficult to handle on computers. They are usually converted into the matrix for computer process. The weights on the edges indicate the costs of the assembly operations to accomplish the assembly relations. Considering multiple quantitative assembly constraints, the cost of assembly relations is computed with the FAHP method.

3 Qualitative assembly constraints and their representation

In this section, we concern the qualitative assembly constraints and merge them into the weighted assembly precedence graph for ASP. For mechanical assembly, the following four rules are useful to reduce the complexity and cost of assembly process. (1) The inside parts (or components) of products are assembled before the outside parts. This rule facilitates the invisibility and accessibility of assembly operations. (2) The underneath parts (or components) are assembled before the upper parts. The support of the lower parts to the upper parts is regarded to reduce the brackets or fixtures. (3) The heavy parts or components are assembled ahead of the light parts. (4) The functional parts are assembled first and then the connectors are later. To make the assembly process successive, the other two rules are also important. (5) The successor part or component has at least one assembly relation with one of the ancestor parts or components. Otherwise, the auxiliary jigs or brackets will be necessary. (6) The ancestor parts or components cannot prevent the successor parts, or the following assembly process will be prohibited by the ancestor parts. The above six rules are used to extract the qualitative assembly constraints for ASP.

Given an assembly, each part or component contact the others with its outer geometrical features, such as dots, lines, and surfaces. The contacts between parts are the simplest assembly relation as qualitative assembly constraints. Several types of contacts between parts are summarized in Fig. 2. In general, every part has contacts with its adjacent parts and has no contacts with the further parts. Some contacts may realize the mechanical functions, such as the clearance fit. Before they are activated, there is no physical force on the contacts except the gravity. We call this kind of contacts as loose contact. Support contact is another kind of loose contact. The simple geometrical mates without force are also loose. On the other hand, most of contacts are the tight contacts. If the contacts are enforced with connectors, a big force will emerge on these contacts. The connectors are an important way to fasten the discrete parts together with the physical force. The connectors are classified into the dismountable and non-dismountable. The dismountable connectors include screw joint, key joint, pin joint, etc. The weld joint, rivet joint, glue joint, interference fit, etc. belong to the non-dismountable connectors.

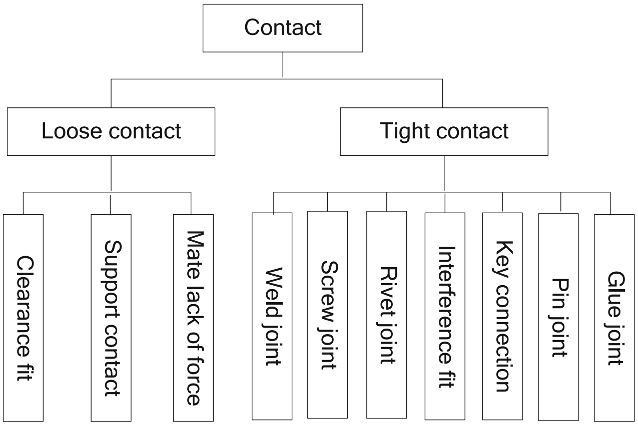


Fig. 2 The classification of contacts

Provided that the assembly precedence is not included, the assemblies are usually represented as an assembly relation graph or liaison diagram. An assembly relation denotes a contact between each pair of parts. For computer process, the assembly relation graph is usually changed into a contact matrix [43]. The BOOL elements in the matrix imply the contacts between each pair of two parts. If parts p_i and p_j are assembled with a kind of contact, the BOOL variable $l_{ij}=1$ ($i \neq j$). Otherwise, $l_{ij}=0$. Obviously, the contact matrix is symmetrical. Two parts may have several contacts on different positions. For ASP, all these contacts between two parts are simplified into one integrative contact. If an assembly includes n parts, the contract matrix is represented as $LM=[l_{ij}]_{n\times n}$. For example, the simple transmission [4] with the 11 parts is given in Fig. 3a and its assembly relation graph is shown in Fig. 3b. The assembly relation graph is apt to construct with respect to the assembly model as well as the LM. However, it only includes the contact information between parts and the other assembly constraints, such as the geometrical and process constraints, are neglected. Without the other qualitative assembly constraints, the number of assembly sequences is huge and a lot of them are invalid. For example, in Fig. 3b, if part 1 is taken as the base part, the number of whole assembly sequences is

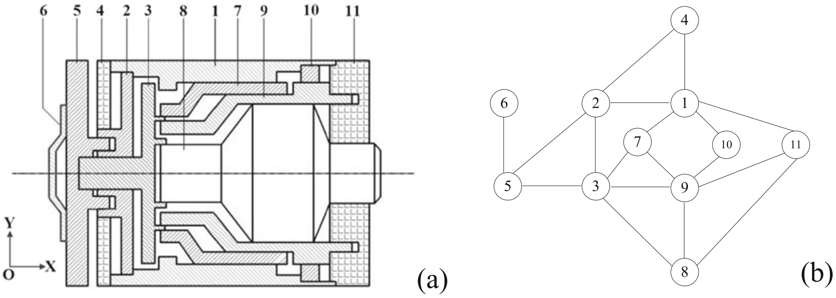


Fig. 3 A simple transmission assembly (**a**) and its assembly relation graph (**b**)

more than 90,000. For complex products, it is difficult to search the optimal assembly sequence based on the assembly relation graph. To reduce the complexity, the other qualitative assembly constraints must be considered.

The assembly relation graph conforms to rule (5), and the other fundamental rules are not considered. The assembly interference will happen for most of the infeasible assembly sequences. The geometrical constraints will often result in assembly interference if they are not handled properly. Owing to the assembly structures and geometrical shapes of parts, some parts must be assembled prior to the other parts to avoid the assembly interference. For example, in Fig. 3a, parts 4 cannot be assembled to part 2 if parts 2 and 5 are assembled first. To prevent the assembly interference, the geometrical constraints are certainly taken into account in the assembly process. The geometrical constraints have various representations, such as the interference matrix [12], assembly precedence [14], disassembly precedence [10], etc. The assembly precedence directly indicates the assembly orders of parts. In addition, they are apt to be derived from the 3D assembly model [44]. It is convenient to represent as an arrowed edge in the directed graph and BOOL variables used to deduce the assembly sequences automatically. Therefore, we transform the geometrical constraints into the assembly precedence which is represented as an arrowed edge in the weighted assembly precedence graph. The arrow points from the successor parts to the ancestor parts. For example, in Fig. 1, the part 1 is assembled first and the parts 2, 3, and 4 are assembled to the part 1 in the next steps.

It notes that the parts without contacts may have assembly precedence due to the restrictions of assembly structures and assembly processes. For example, in Fig. 3a, part 4 must be assembled before part 5 if part 1 is assembled first. This kind of assembly precedence between parts without contacts should be added to the weighted assembly precedence graph. Such assembly precedence between the parts without contacts is noted as the dashed lines. The arrows also point from the successor parts to the ancestor parts. For instance, in Fig. 3, the directed graph model with the geometrical constraints is given in Fig. 4, where part 1 is viewed as the base part. Referring to Fig. 4, we find a part may have many ancestor parts. To prove the safety of the assembly process, we make the seventh rule. (7) A part cannot be assembled until all its ancestor parts have been assembled. This rule may be a little mandatory, but it has advantages in actual assembly process. Firstly, it will generate the optimal assembly sequence under the consideration of the actual assembly constraints. Secondly, the rule (7) will omit many non-optimal assembly sequences and the search space of the optimal assembly sequence is decreased greatly.

The geometrical constraints have been merged into the directed graph model. To generate the feasible assembly sequences automatically, the assembly precedence between parts p_i and p_j is represented as another BOOL variables pr_{ij}. For two parts p_i and p_j, if part p_i must be assembled after part p_j, $pr_{ij}=1$ $(i \neq j)$. Otherwise, $pr_{ij}=0$. Given an assembly with n parts, the precedence matrix is represented as $PM=[pr_{ij}]_{n\times n}$. In the ith line, the part p_i cannot be assembled until all the elements $pr_{ij}=0$. Different from the contact information, the assembly precedence is asymmetrical, which proves some parts have to be assembled before the other parts. With the geometrical constraints in Fig. 4, the number of feasible assembly sequences of the simple transmission becomes 140. It is clear that geometrical constraints play an important role to reduce the complexity of ASP.

It notes that assembly precedence is a kind of comprehensive assembly constraints. Besides the geometrical constraints, the designers can add the other qualitative constraints according to the above rules or delete some redundant

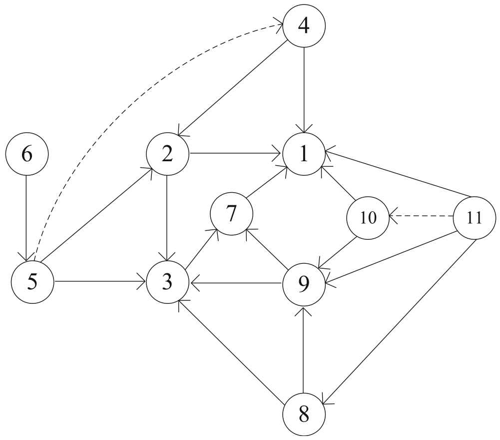


Fig. 4 The assembly precedence graph with geometrical constraints

qualitative constraints to generate the favorite assembly sequences. For example, the process constraints of assembly line layout, the special assembly operations, and the invisibility and inaccessibility of operations will increase the number of assembly precedence. If necessary, these qualitative assembly constraints will be added according to the assembly structures and assembly environments.

Before we execute the assembly process, we first appoint a base part with respect to its characteristics. The base part owns the assembly baselines. It is usually heavy, big, and assembled at the first step. Then, the other discrete parts or components are assembled to the base part or subassemblies one after another until they compose the whole product. Generally, a global coordinate system is attached to the base part (or component). The positions of every part or component are determined in the coordinate system. The outer and inner parts are visible in view of their solid models and the coordinate system. The geometrical interference between parts is also able to discern. After the qualitative assembly constraints are acquired, they are represented as the directed edges in the weighted assembly precedence graph model to ensure the assembly precedence between parts.

4 Quantitative assembly constraints and their representation

Considering the qualitative assembly constraints, the valid assembly sequences will be generated. The goal of ASP is not only to generate the feasible assembly sequences, but also to obtain the optimal assembly sequence according to assembly criteria. Given a fitness function of the assembly sequences, the optimal assembly sequence has the minimum (or maximum) fitness value. For complex products, the number of feasible assembly sequences is still very large in the assembly precedence graph. It is impossible to generate all of the assembly sequences due to the limitation of the computation resources. Furthermore, the optimal sub-sequences are not generated due to the lack of local heuristic information. The efficient methods, such as the graph search algorithms, branch and bound method, cutting planning method, dynamic programming method, etc., cannot find the optimal sub-sequences based on the assembly precedence graph. Therefore, the heuristic information is necessary to add to the assembly precedence graph. When we select the next part to assemble, the heuristic information will show us which part is the most appropriate. Provided that each operation assembles one part, the assembly is assembled by a set of ordered assembly operations. In each operation process, the successor part is added to its ancestor parts through setting up their assembly relations. The costs or time of the assembly operations to accomplish the assembly relations is naturally taken as the heuristic information to choose the proper part to assemble. Referring to Fig. 4, the costs of assembly operations can be attached to the edges representing the assembly relations as their costs. In this case, the assembly precedence graph becomes a weighted precedence graph. Comparing with the unweighted graph, the weighted assembly precedence graph has the following merits. (1) It is an integrative assembly model for ASP. Most of the assembly constraints including the qualitative and quantitative assembly constraints are integrated into the model together. The model can adapt the various demands of assembly structures and assembly backgrounds. (2) The complexity of ASP will be reduced much after multiple assembly constraints are taken into consideration. In addition, the heuristic information will lead the algorithms and methods to find the optimal sub-sequences and sequence. A lot of non-optimal sub-sequences and non-optimal sequences will be neglected in the search process. (3) It is a simple graph model and apt to be converted into the matrix for computer process. Most of the algorithms and reasoning methods can resolve ASP based upon the graph model.

There are many criteria to evaluate the assembly operations. An assembly relation will be accomplished in each assembly operation. The parts and their assembly relations realize the total or a portion of the product functions. The assembly relations to realize the product functions are designed as the special contacts on the important parts. Each contact is composed of one or more pairs of particular features [45] on different parts. These parts are functional parts, and they are usually assembled prior to the other parts. Secondly, the accomplished assembly relations are desired to maintain the stability of subassembly. Otherwise, the auxiliary brackets or jigs must be provided. Thirdly, the successor part in the next operation should have the same assembly directions as that of the adjacent ancestor part to reduce the number of reorientations. Additionally, it is best assembled with the identical tools as those used by the adjacent ancestor part to reduce the changes of assembly tools. In literature [46], Wang et al. gave an evaluation model to compute the decision graph for assembly unit partition. Here, the similar assembly constraints are summarized as the quantitative constraints to compute the costs of the assembly relations for ASP.

The quantitative assembly constraints are shown in Fig. 5. The quantitative assembly constraints are classified into three types: functional, structural, and process constraints. The stable support and transmission belong to the functional constraints, which realize the design intent of products. The stable support and transmission contacts are designed as the special features on the important parts. Regardless of loose or tight contacts, the parts are fastened together to form the product structures. The parts are gathered by all kinds of connectors and mates. Some parts are fastened with connectors or joints. The other parts mate with each other with particular structures [47]. No matter whatever the assembly structure is, the degree

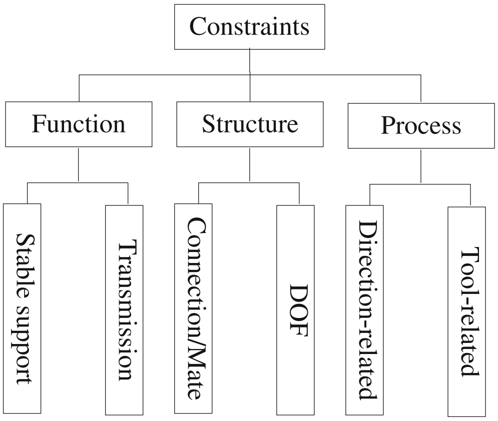


Fig. 5 The quantitative assembly constraints to evaluate assembly relations

of freedom (DOF) of the assembled parts will be reduced. The DOF of parts is taken as the second structure constraints to evaluate the stability of assembly relations. At last, the assembly process related to the assembly complexity is regarded. The frequent changes of assembly directions and tools will lengthen the assembly time and improve the assembly cost through raising the assembly complexity. The two factors are considered as the process constraints to evaluate the assembly relations. We will compute the costs of assembly relations in view of these quantitative constraints. The costs are assigned to the edges in the weighted assembly precedence graph. The smaller the weight of one edge is, the less the cost to accomplish the assembly relation is. In the assembly process, we will select the successor part with the minimum assembly relation cost to assemble. If all the parts are assembled with their minimum costs, the assembly sequence will be optimal. Two steps are necessary to obtain the cost of the assembly relations. In the first step, the assembly constraints in Fig. 5 will be quantified as their indices. Secondly, the weights to hint the importance of the assembly constraints are given by engineers and the cost of assembly relations are computed with the FAHP method.

4.1 Quantization of assembly constraints

The indices of the quantitative assembly constraints in Fig. 5 are defined or given by the designers to compute the cost of assembly relations.

4.1.1 The indices of functional constraints

Two functional constraints, i.e., the stable support and transmission, are considered here. The two constrains are generally determined in the design process to realize the product function. The parts to achieve the product functions are important. In the assembly process, the important parts are usually assembled prior to the other parts.

Given two parts p_i and p_j, the part p_j is on the part p_i. If the support force of p_i to p_j is equal to or bigger than the gravitation of part p_j, we name part p_i as a stable support of part p_j. It notes that the support forces are vectors. The direction of the support forces is coincident with that of the gravity of the supported parts. If part p_i is a stable support of part p_j, the index of the stable support is defined as $SI_\mathrm{ij}=0$. Otherwise, $SI_\mathrm{ij}=1$. The unstable support will need the extra auxiliary brackets to secure the assembly process. We advise to reduce the unstable supports in the products as few as possible to decrease the complexity of assembly process. The stable support indices SI_ij between parts will constitute a stable matrix SM and it is asymmetrical.

Transmission is another important function of mechanical products. The movements and driving forces are transferred by various kinematic pairs. Besides the materials and shapes, the tolerances are specially designed on the kinematic pairs [48]. If p_i and p_j are assembled to realize the transmission function, the index of p_i and p_j is given as $\mathrm{TMI}_\mathrm{ij}=0$. Otherwise, $\mathrm{TMI}_\mathrm{ij}=1$. Similarly, the transmission matrix TMM is filled by the indices TMI_ij between each pair of two parts. Obviously, it is symmetrical.

The two indices denote the functional dependence between parts p_i and p_j. They are merged into the functional index as formula (2).

$$FI_\mathrm{ij} = w_1 \times SI_\mathrm{ij} + (1-w_1) \times TMI_\mathrm{ij} \tag{2}$$

Where FI_ij is the functional constraint index of assembly relation l_ij formed with parts p_i and p_j, w_1 is the weight of stable support, which illustrates its importance, $0\le w_1\le 1$, $0\le i, j\le n$, and n is the number of parts in the assembly.

4.1.2 The indices of structural constraints

The structural constraints of products keep every static part staying at their right locations and the dynamic parts moving in their working space. Some parts are fastened with various connectors, such as weld, rivets, screw, etc. in Fig. 2, to from the steady assembly structures. Some parts only mate with each other and there is no force on their contacts. In the assembly process, the parts fastened with connectors or joints are preferred. The connectors are apt to keep the steady structure of the subassemblies. The unsteady states of the assembly structure will be avoided to reduce the complexity of assembly procedures. For different types of connectors, the strength they provide has distinctions. Referring to Lee's work [49], the strength of the connectors is classified into eight categories. The weld strength is the strongest and the loose mate (attachment) is the weakest. In this research, a small connection index is assigned to the strong connectors to compute the small costs of assembly relations. The strength indices of the connectors are shown in Table 1. If two parts p_i and p_j are

Table 1 The strength indices of the connectors

Connectors	Weld	Rivet	Screw	Push
Index/CI	0.1	0.2	0.3	0.4
Connectors	Interference fit	Sticking	Clearance fit	Loose mate
Index/CI	0.5	0.6	0.8	1.0

fastened with a connector, the strength index of the assembly relation is represented as CI_{ij}. The connectors are used to fasten the functional parts and they are not taken into account in the ASP. The strength indices CI_{ij} between parts compose the connection strength matrix *CM* and it is symmetrical.

The connectors join the pair of parts with forces. The other parts have relative motions after they are assembled with kinematic pairs. Although these parts can move in their working area, their DOFs are restricted by the kinematic pairs. A part has 6 DOFs in an open space. The DOF becomes less than 6 after some of its assembly relations are realized. Each assembly relation on the part will constraint one or more of its degrees. The DOF of parts hint the second assembly structure constraint determined by the assembly relations. Given part p_i is the ancestor part, part p_j is assembled to the part p_i. The DOF of part p_j is noted as DOF_{ij} after they are assembled. The index of DOF of part p_j is computed as $\mathrm{DOFI}_{ij}=\mathrm{DOF}_{ij}/6$. The smaller the index DOFI_{ij}, the steadier the subassembly structure is after part p_j is assembled. The DOF indices between parts compose the DOF matrix (DOFM) which illustrates their structure stability when two parts are assembled. The DOFM is also symmetrical.

The two indices imply the structure constraints to prove the assembly structure stability in the assembly process. They are combined together as formula (3).

$$\mathrm{AI}_{ij} = w_2 \times \mathrm{CI}_{ij} + (1-w_2) \times \mathrm{DOFI}_{ij} \tag{3}$$

Where AI_{ij} represents the structure constraint index of assembly relation l_{ij} when part p_j is assembled to part p_i, w_2 is the weight of connection strength, $0 \le w_2 \le 1$, $0 \le i, j \le n$, n is the number of parts in the assembly.

4.1.3 The indices of process constraints

For an assembly sequence, the frequent changes of assembly directions and tools will increase the assembly time and cost. We expect the parts to be assembled in a few directions and with a small number of tools so that the assembly time and cost will be saved. Given an ancestor part or component, the possible parts to be assembled in the next step will be multiple. We prefer to select the successor parts with the same assembly direction and tools used by the adjacent ancestor part. A part may have many assembly relations with the ancestor parts. The assembly directions and tools to complete these assembly relations are different in most cases. We give the following assumptions to simplify the ASP. (1) In the 3D coordinate system, the assembly directions are strictly along $\pm X$, $\pm Y$, and $\pm Z$ directions. This hypothesis is suitable for most of the assembly processes. (2) If one assembly relation between two parts p_i and p_j is accomplished, the pair of two parts is assembled together. (3) Given the ancestor part or component, the successor part is assembled to it with one assembly tool along one assembly direction. Based on these assumptions, we assign the assembly directions and tools to each part with respect to the assembly relations. For each assembly relation l_{ij}, it connects two parts p_i and p_j. In the coordinate system, the assembly direction of part p_i or p_j is fixed in view of the assembly relation l_{ij}. If part p_j is assembled to p_i in the X direction, the part p_i will be assembled to p_j in the reverse direction. To derive the assembly directions of parts, we build the direction matrix $DM=[d_{ij}]_{n\times n}$. For the element d_{ij} in the matrix, it means the assembly direction of part p_i if part p_j is the ancestor part. $d_{ij} \in \{X, -X, Y, -Y, Z, -Z\}$ and $d_{ij}=-d_{ji}$.

For two parts p_i and p_j with an l_{ij} and p_i is the ancestor part, if the assembly direction of the successor part p_j is the same as that of the adjacent assembled part p_k in the assembly process, the index of the assembly direction constraint between the parts p_i and p_j is noted as $DI_{ij}=0$. Otherwise, $DI_{ij}=1$. It notes that part p_k may be identical to or different from part p_i.

As the direction constraint, we would like to assemble the successor part p_j with the same tool as that used by the adjacent assembled part p_k. However, the confirmation of the assembly tools is more complex than that of the assembly directions. Given an assembly relation l_{ij}, it can be accomplished by different tools. On the other hand, every assembly tool is able to assemble several similar types of assembly relations. To select the proper tools, the types of assembly relations between parts are given first. For example, there are m kinds of assembly relations in an assembly. These assembly relations are noted as a relation set $R=\{r_1, r_2, \ldots, r_m\}$. For each type of assembly relation r_i, s kinds of tools $T=\{t_1, t_2, \ldots, t_s\}$ are able to accomplish it in one particular assembly environment. Therefore, a tool matrix TOM will be suggested with respect to the assembly relations. In the assembly process, the identical or similar tools are selected to realize the following uninstalled assembly relations. For two parts p_i and p_j with an l_{ij} and p_i is the ancestor part, if part p_j is assembled closely after part p_k with the tools which are the same as those used by the part p_k, the index of assembly tool constraint between parts p_i and p_j is represented as $\mathrm{TOI}_{ij}=0$. Otherwise, $\mathrm{TOI}_{ij}=1$. The part p_k may be identical to or different from part p_i.

These two indices represent the assembly complexity determined by the assembly process. They are the base to compute the index of assembly process constraint as formula (4).

$$PI_{ij} = w_3 \times DI_{ij} + (1-w_3) \times TOI_{ij} \tag{4}$$

Where PI_{ij} is the process constraint index of assembly relation l_{ij} when part p_j is assembled to part p_i, w_3 is the weight of assembly directions, $0 \le w_3 \le 1$, $0 \le i, j \le n$, and n is the number of parts in the assembly. Different from the indices of functional and structural constraints, the indices of assembly process constraints are computed in the assembly process. The PI_{ij} between parts p_i and p_j relies on the middle part p_k (p_k may be different from part p_i). Therefore, the PI_{ij} between parts p_i and p_j varies for different assembly sequences.

It notes that the stable support, direction-related and tool-related constraints are asymmetrical for each pair of two parts with an assembly relation l_{ij}. The other three constraints are symmetrical for two assembled parts. The weights w_1, w_2, and w_3 are assigned by the designers to denote the importance of the assembly constraints. In the following section, the assembly cost of the assembly relations are computed with the FAHP method based upon these indices.

4.2 Computation of the assembly cost of assembly relations

FAHP method [50] originates from the analytical hierarchy process (AHP) introduced by Satty [51]. It has a lot of applications to the fuzzy making-decision problems in the engineering fields. The evaluation of costs of the assembly relations is just this kind of multiple criteria decision making (MCDM) problems. In view of Fig. 5, the functional, structural, and process constrains are considered to evaluate the costs of assembly relations for ASP. The indices of these constraints are computed in previous section. They are used to compute the cost of assembly relations with the FAHP method. There are four main steps to compute the cost of assembly relations.

Step 1 Give the evaluation model of the assembly relations for ASP. The evaluation model of the assembly relations is illustrated in Fig. 6. The top level is the cost of assembly relations to compute. The second level is the important criteria (the functional, structural, and process constraints). The bottom level is the assembly relations. In the hierarchical model, the assembly relations are evaluated according to the three kinds of assembly constraints. The smaller the cost of an assembly relation, the less the cost is to accomplish it.

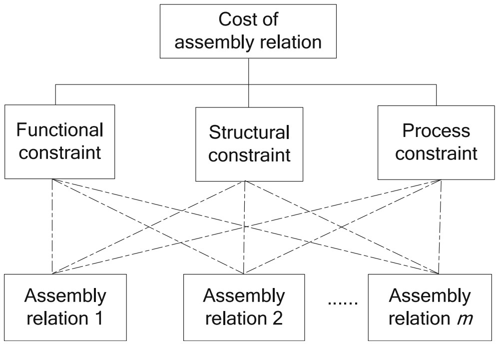


Fig. 6 The evaluation model of assembly relations

Step 2 Compute the fuzzy judgment matrix of the quantitative assembly constraints. The elements of judgment matrix are computed with formulae (2)–(4). To add the fuzziness of judgment, the elements in the judgment matrix are changed into the triangle fuzzy numbers. The reflection between the real numbers and the fuzzy numbers is given in Table 2. With the mapping rules, each real number is converted into a triangle fuzzy number. If the assembly includes n assembly relations and m indices are considered, the fuzzy judgment matrix is noted as $\tilde{B} = \left[\tilde{B}_{ij}\right]_{n\times m}$ $(1 \le i \le n, 0 \le j \le m)$. $\tilde{b}_{ij}$ is a triangle fuzzy number and it is noted as $\tilde{b}_{ij} = (u, v, w)$. u, v, and w are integer numbers between 1 and 9.

Step 3 Appoint the fuzzy weight vector. Three kinds of quantitative constrains are regarded for ASP. The importance of them is distinctive for different assembly structures and various assembly environments. The weights w_F, w_S, and w_P of the functional, structural, and process constrains are assigned by engineers and $w_F + w_S + w_P = 1$. The suggestions of the experts are merged into the FAHP method. After the three weights are given, they are changed into the fuzzy numbers according to Table 2. If m quantitative assembly constraints are considered, the fuzzy weight vector is represented as $\tilde{W} = [\tilde{w}_i]_{m\times 1}$ $(1 \le i \le m)$. Each element $\tilde{w}_i$ is a triangle fuzzy number.

If the AHP method is used, the weight judgment matrix of the assembly constraints is given by the engineers. The weights are computed as the feature vectors of the weight judgment matrix. Due to the simplicity of the weight judgment matrix (just a 3×3 matrix), the weights w_F, w_S, and w_P are assigned by experts according to the actual assembly process. The experiences of the engineers are merged into the FAHP method to generate the practical assembly sequences.

Step 4 Calculate the cost of assembly relations. The fuzzy cost of assembly relations is equal to the product of fuzzy judgment matrix and fuzzy weight vectors. It is the formula (5), where $\otimes$ is the times sign for triangle fuzzy numbers, the superscript T denotes the transposition of matrix.

$$\tilde{C}_{n\times 1} = \left[\tilde{b}_{ij}\right]_{n\times m} \otimes \left[\left(\tilde{w}_i\right)^{T}\right]_{m\times 1} \tag{5}$$

Table 2 The reflection between the real numbers and the triangle fuzzy numbers

Real number	0	(0, 0.2]	(0.2, 0.4]	(0.4, 0.6]	(0.6, 0.8]	(0.8, 1.0]
Triangle fuzzy number	(0,0,0)	(1,1,3)	(1,3,5)	(3,5,7)	(5,7,9)	(7,7,9)

In the assembly process, the fuzzy cost of each assembly relation is computed. In the next, they are converted into the cost of assembly relations for ASP. The cost of assembly relation is the average value of the triangle fuzzy numbers. For example, if $\tilde{c} = (u, v, w)$, the cost c of the assembly relation is computed as formula (6).

$$c = \frac{u+v+w}{3} \tag{6}$$

The cost is attached to the edges representing the assembly relations in the assembly precedence graph. Then we will obtain a weighted assembly precedence graph model for ASP. The efficient methods and algorithms can resolve the ASP based on the weighted assembly precedence graph.

5 A MST-based algorithm based on the model

With the weighted assembly precedence graph, the ASP turns into a variation of asymmetrical TSP under the qualitative assembly constraints. Given an assembly with n parts or components, the integer program of ASP is shown as model (7) with four formulae (a)–(d).

$$\begin{cases} \min \sum_{i=1}^{n} \sum_{j=1, j\neq i}^{n} c_{ij} x_{ij} & \text{(a)} \\ s.t & \\ \sum_{i=1, i\neq j}^{n} x_{ij} = 1, \forall 1 \leq i \leq n & \text{(b)} \\ \sum_{j=1, j\neq k}^{n} pr_{jk} = 0, \forall 1 \leq k \leq n & \text{(c)} \\ x_{ij} \in \{0, 1\} \forall 1 \leq i \neq j \leq n & \text{(d)} \end{cases} \tag{7}$$

In formula (a), c_{ij} means the cost of assembly relation l_{ij} between parts p_{i} and p_{j}, where part p_{i} is the ancestor part of part p_{j}. For each successor part p_{j} to be assembled, it may have many assembly relations with the ancestor parts p_{i}. Only one assembly relation is considered to assemble the part p_{j} in one operation, this is the function of the formula (b). This formula also guarantees that at least one assembly relation exists between the part p_{j} and the ancestor parts p_{i}. Formula (c) is the precedence constraints between part p_{j} and the other parts. It means that part p_{j} can be assembled after all its ancestor parts p_{k} are assembled. Formula (4) proves n-1 assembly relations are accomplished in one assembly sequence.

Although ASP is represented as a variation of TSP, it has the following obvious distinctions from the asymmetrical TSP. (1) Generally, the weighted assembly precedence graph is not complete. For a concrete assembly, each part usually has assembly relations with its several adjacent parts. The average degree of the weighted assembly precedence graph will be much less than n-1 provided that the assembly includes n parts or components. (2) The assembly precedence is unilateral. For example, if part p_{i} must be assembled before part p_{j}, part p_{j} cannot be as the ancestor of part p_{i}. (3) In an assembly sequence, the subsequent parts at least have one assembly relation with one of their ancestor parts. (4) All the assembly relations are accomplished after the parts are assembled together. In formula (7), the n-1 assembly relations to join the n parts together are considered to generate the optimal assembly sequence. The optimal assembly sequence has the minimum assembly cost.

Except for the four distinctions, the ASP can be taken as an asymmetrical TSP under the qualitative assembly constraints. In the weighted assembly precedence graph, one part may have several assembly relations with the other parts. How can we select the proper assembly relation to assemble? In the assembly process, the parts are divided into two groups. Given the first group of parts has been assembled and the second group of parts is not, the costs of the assembly relations between the two groups of parts are computed first. Then the assembly relation with the minimal cost is selected to assemble. Meanwhile, the part with the assembly relation in the second group is assembly to the first group. The assembly relation with the minimum cost meets the demands of the multiple assembly constraints and it is preferred to assemble. If the assembly relation with the minimal cost is realized, the two parts linked by the assembly relation is assembled together.

Given a base part, the costs of the assembly relations linked to it are computed first. The next part connected by the assembly relation with the minimum cost is selected and assembled to the based part. They compose the ancestor subassembly. The costs of the assembly relations between the ancestor subassembly and the other discrete parts are computed to select the third part based on the minimum assembly cost. The computation process is executed until all the parts are traversed. It is obvious that we generate a minimum spanning tree (MST) of the weighted assembly precedence graph. Given an assembly with n parts or components, the MST-based algorithm to search the optimal assembly sequence is given in Fig. 7, where n_1 is the number of assembled parts.

```
Step  The pseudo to find the optimal assembly sequence
1     The assembly relation graph is constructed in view of the contacts between parts.
2     The assembly precedence between parts are derived from the CAD model or appointed by engineers according to the assembly structures and environments and merged into the assembly relation graph.
3     The indices of the quantitative assembly constrains are defined and computed.
4     A base part or component is appointed and add it to the optimal assembly sub-sequence. The assembly direction and tools to assemble the base part or component are given.
5     For(i:=2~n)
6         Classify the parts into two groups which are the assembled parts and the uninstalled parts. The number of assembled parts is n1 and that of the uninstalled parts is n-n1.
7         For(j:=1~n1)
8             For(k:=1~n)
9                 Find the assembly relations between the assembled parts in the optimal assembly sub-sequence and the other uninstalled discrete parts.
10                Compute the cost of the assembly relations between the two groups of parts with the FAHP method.
11                Select the assembly relation with the minimum cost. Record the uninstalled part linked by the assembly relation with the minimum cost.
12            end
13        end
14        Add the current part with the minimum cost into the optimal assembly sub-sequence.
15        Compute the cost of the optimal assembly sub-sequence. Record the assembly direction and tools to assemble the current part.
16        Update the precedence matrix.
17    end
18    Output the optimal assembly sequence, the minimum cost and the corresponding directions and tools to assemble each part in the optimal assembly sequence.
```

Fig. 7 The MST-based algorithm to search the optimal assembly sequence

The previous 4 steps input the necessary assembly constraints to construct the weighted assembly precedence graph for ASP. After the weighted assembly precedence graph is built, the MST-based algorithm is adopted to search the MST under the qualitative assembly constraints. The costs of the assembly relations are computed with the FAHP method in the 10th step. Among the assembly relations between the assembled parts and the uninstalled parts, the uninstalled part linked by the assembly relation with the minimum cost is taken as the next part to assemble and added to the optimal assembly sub-sequence. Simultaneously, the assembly directions and tools for assembling the part are also recorded. After the part is assembled to the subassembly, it cannot prohibit the other uninstalled parts any more. The precedence matrix is updated in the 16th step. The discrete parts are added into the optimal sub-sequence one by one until all of them are assembled together. Steps 5–17 execute the computation to generate the optimal assembly sequence. In the 18th step, the results are output. They include the optimal assembly sequence, the assembly cost, the assembly directions and tools to assemble each part.

The time complexity of the algorithm is mainly concluded by the 5th–17th steps. Given an assembly with n parts, the maximum value of n_1 is n. Hence, the time complexity of the algorithm is $O(n^3)$. The time complexity of the dynamic programming methods is usually bigger than $O(2^n)$ in most cases [13].

Given an assembly with n parts $\{p_1, p_2, \ldots, p_n\}$, an assembly sequence is represented as $(p_i, p_j, \ldots, p_m)$, where all the parts in the assembly sequence are different. Meanwhile, the assembly directions and tools to assemble each part are represented as the direction and tool sequences.

$$\begin{bmatrix} 0&0&0&0&0&0&0&0&0&0&0\\ 1&0&1&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&1&0&0&0&0\\ 1&1&0&0&0&0&0&0&0&0&0\\ 0&1&1&1&0&0&0&0&0&0&0\\ 0&0&0&0&1&0&0&0&0&0&0\\ 1&0&0&0&0&0&0&0&0&0&0\\ 0&0&1&0&0&0&0&0&1&0&0\\ 0&0&1&0&0&0&1&0&0&0&0\\ 1&0&0&0&0&0&0&0&1&0&0\\ 1&0&0&0&0&0&0&1&1&1&0 \end{bmatrix}$$

(a) *PM*-precedence matrix

$$\begin{bmatrix} 1&0&1&1&1&1&0&1&1&0&0\\ 1&1&1&0&0&1&1&1&1&1&1\\ 1&0&1&1&0&1&1&0&0&1&1\\ 1&1&1&1&1&1&1&1&1&1&1\\ 1&1&1&1&1&0&1&1&1&1&1\\ 1&1&1&1&1&1&1&1&1&1&1\\ 1&1&0&1&1&1&1&1&0&1&1\\ 1&1&1&1&1&1&1&1&1&1&1\\ 1&1&1&1&1&1&1&0&1&1&1\\ 1&1&1&1&1&1&1&1&0&1&1\\ 1&1&1&1&1&1&1&0&0&1&1 \end{bmatrix}$$

(b) *SM*-support matrix

$$\begin{bmatrix} 1&1&1&1&1&1&1&1&1&0&0\\ 1&1&1&1&1&1&1&1&1&1&1\\ 1&1&1&1&1&1&1&0&1&1&1\\ 1&1&1&1&1&1&1&1&1&1&1\\ 1&1&1&1&1&1&1&1&1&1&1\\ 1&1&1&1&1&1&1&1&1&1&1\\ 1&1&1&1&1&1&1&1&1&1&1\\ 1&1&0&1&1&1&1&1&0&1&1\\ 1&1&1&1&1&1&1&0&1&1&1\\ 0&1&1&1&1&1&1&1&1&1&1\\ 0&1&1&1&1&1&1&1&1&1&1 \end{bmatrix}$$

(c) *TMM*-transmission matrix

$$\begin{bmatrix} 1&0.1&1&0.6&1&1&0.5&1&1&0.8&0.8\\ 0.1&1&0.4&0.5&0.4&1&1&1&1&1&1\\ 1&0.4&1&1&0.4&1&0.5&0.5&0.5&1&1\\ 0.6&0.5&1&1&1&1&1&1&1&1&1\\ 1&0.4&0.4&1&1&0.1&1&1&1&1&1\\ 1&1&1&1&0.1&1&1&1&1&1&1\\ 0.5&1&0.5&1&1&1&1&1&0.5&1&0.8\\ 1&1&0.5&1&1&1&1&1&0.5&1&0.8\\ 1&1&0.5&1&1&1&0.5&0.5&1&0.5&1\\ 0.8&1&1&1&1&1&1&1&0.5&1&1\\ 0.8&1&1&1&1&1&1&0.8&0.8&1&1 \end{bmatrix}$$

(d) *CM*-connection matrix

$$\begin{bmatrix} 0.8&0.3&0.8&0.7&0.8&0.8&0.7&0.8&0.8&0.8&0.8\\ 0.3&0.8&0.5&0.7&0.5&0.8&0.8&0.8&0.8&0.8&0.8\\ 0.8&0.5&0.8&0.8&0.5&0.8&0.7&0.7&0.7&0.8&0.8\\ 0.7&0.7&0.8&0.8&0.8&0.8&0.7&0.8&0.8&0.8&0.8\\ 0.8&0.5&0.5&0.8&0.8&0.3&0.8&0.8&0.8&0.8&0.8\\ 0.8&0.8&0.8&0.8&0.3&0.8&0.8&0.8&0.8&0.8&0.8\\ 0.7&0.8&0.7&0.8&0.8&0.8&0.8&0.8&0.7&0.8&0.8\\ 0.8&0.8&0.7&0.8&0.8&0.8&0.8&0.8&0.7&0.8&0.8\\ 0.8&0.8&0.7&0.8&0.8&0.8&0.7&0.7&0.8&0.7&0.8\\ 0.8&0.8&0.8&0.8&0.8&0.8&0.8&0.8&0.7&0.8&0.8\\ 0.8&0.8&0.8&0.8&0.8&0.8&0.8&0.8&0.8&0.8&0.8 \end{bmatrix}$$

(e) *DOFM*-DOF matrix

$$\begin{bmatrix} -1&2&-1&2&-1&-1&1&-1&-1&1&1\\ 1&-1&1&2&2&-1&-1&-1&-1&-1&-1\\ -1&2&2&-1&-1&-1&1&1&1&-1&-1\\ 1&1&-1&-1&-1&-1&-1&-1&-1&-1&-1\\ -1&1&1&-1&-1&2&-1&-1&-1&-1&-1\\ -1&-1&-1&-1&1&-1&-1&-1&-1&-1&-1\\ 2&-1&2&-1&-1&-1&-1&-1&1&-1&-1\\ -1&-1&2&-1&-1&-1&-1&-1&2&-1&1\\ -1&-1&2&-1&-1&-1&2&1&-1&1&1\\ 2&-1&-1&-1&-1&-1&-1&-1&2&-1&-1\\ 2&-1&-1&-1&-1&-1&-1&2&2&-1&-1 \end{bmatrix}$$

(f) *DM*-direction matrix

$$\begin{array}{c} 0.1\\ 0.2\\ 0.3\\ 0.4\\ 0.5\\ 0.6\\ 0.8\\ 1.0 \end{array} \begin{bmatrix} 6&7&8&9&10\\ 5&6&7&8&9\\ 4&5&7&8&9\\ 4&5&6&7&8\\ 3&4&5&7&9\\ 3&4&5&6&8\\ 2&3&4&5&6\\ 1&2&3&4&5 \end{bmatrix}$$

(g) *TOM*-tool matrix

Fig. 8 The information of assembly constraints for ASP

Table 3 The results of the simple transmission

Sequence	(1, 7, 3, 2, 4, 9, 8, 10, 11, 5, 6)	Min cost=583
Directions	$(-X, -X, X, X, X, -X, -X, -X, -X, X, X)$	Number of direction changes=3
Tools	$(t_3, t_3, t_3, t_6, t_3, t_3, t_3, t_3, t_3, t_4, t_6)$	Number of tool changes=4

6 Examples and analysis

The method to compute the weighted assembly precedence graph and the MST-based algorithm are coded with C++ language and run on a computer with a 2 GHz processor and a 512 MB inner memory. After the quantitative and qualitative assembly constraints are provided, the program is used to search the optimal assembly sequence. In the assembly process, the costs of the assembly relations are computed with the FAHP method considering the quantitative assembly constraints.

Two examples are given to show the performance of the weighted precedence graph model for ASP. Given an assembly, the qualitative and quantitative assembly constraints are extracted from the assembly or assigned by the engineers. After the assembly process is initialized, the MST-based algorithm is implemented to find the optimal assembly sequence within the weighted assembly precedence graph. For convenience, we use sets {1,2,3,4,5,6} represent the assembly direction sets $\{X,-X, Y,-Y, Z,-Z\}$. The assembly tools are also noted as integer numbers. After the results are computed, we converted them into the actual assembly directions and tools.

6.1 The simple transmission in Fig. 3

The simple transmission in Fig. 3 is taken as the first instance to show the feasibility of the weighted precedence graph model. The precedence matrix PM, stable support matrix SM, transmission matrix TMM, connection matrix CM, degree of freedom matrix DOFM, direction matrix DM, and tool matrix TOM are given in Fig. 8. The precedence matrix *PM* in Fig. 8a is derived from the assembly precedence graph in Fig. 4, which guarantees to generate the feasible assembly sequences. The other matrices are defined and computed according to the methods introduced in Section 4.1. In the direction matrix-DM, the elements $DI_{ij}=-1$ note that there is no relative assembly direction between the parts p_i and p_j. The assembly direction of each part can be deduced according to DM. The assembly tools are determined by the types of connectors in Table 1. The connectors of the simple transmission are classified into 6 types with respect to the CM. In Fig. 8g, the elements in the left column outside of TOM are the connector types. For each type of connector, we give 5 tools to use and each tool can accomplish the connection. The tools to accomplish different connections may be the same. The tools are represented as integer numbers in the TOM. The bigger the tool number is, the more expensive the tool is. In general, we prefer to select the cheap tools to assemble the parts. Part 1 is viewed as the base part in view of the PM. It has the most assembly relations with the other parts. Its assembly direction is $+X$ or $-X$ in view of DM.

To compute the cost of the assembly relations, the weights w_1, w_2, w_3 and w_F, w_S, w_P should be given. The engineers assign the proper values to these weights considering the actual assembly backgrounds. In this example, $w_1=0.5$ $w_2=0.5$, $w_3=0.5$ and $w_F=0.4$, $w_S=0.4$, $w_P=0.2$. The smaller the weight is, the more important the corresponding assembly constraint is. In view of these weights, the assembly process constraint is more important than the other two assembly constraints. The experts can change the values of these weights to generate the applicable sequences.

With the above assembly constraints information, the results are computed and shown in Table 3. The optimal assembly sequence meets the comprehensive assembly constraints and its cost is the minimum. The minimum cost is rounded to an integer. If we change the quantitative assembly constraints or the weights of these constraints, the optimal assembly sequences will be different.

Wang et al. [52] used the ACO, and Gao et al. [53] adopted the memetic algorithms (MA) and GA to search the optimal assembly sequence of the simple transmission. In their research, they first generate a lot of assembly sequences through iterative computations. Then, the assembly sequences are evaluated with the defined fitness functions. They use the number of reorientations to evaluate the assembly sequences. The comparisons between their results and ours are shown in Table 4. The reorientations of our results are the same as that of GA and bigger than that of ACO and MA. Our qualitative assembly constrains is different from theirs so that the results are different. For example, under the constraints of our PM, part 3 must be assembled before part 2 if part 1 is the ancestor part, which leads to the difference between our optimal assembly sequence and theirs. In our method, more assembly

Table 4 The comparisons between our methods with the others

Method	Optimal sequence	Reorientation
MST-based	(1,7,3,2,4,9,8,10,11,5,6)	3
GA	(1,2,3,4,7,9,8,10,5,6,11)	3
ACO	(11,8,9,7,10,1,2,3,4,5,6)	0
MA	(1,2,3,4,5,6,7,10,9,8,11)	1

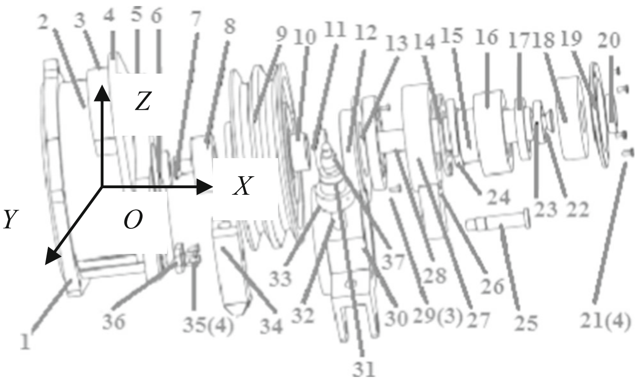


Fig. 9 The transmission assembly with the 37 parts

constraints are considered in the weighted assembly precedence graph and the optimal assembly sequence will be more practical. With the MST-based algorithm, we do not need to generate a lot of assembly sequences to find the optimal assembly sequence. With the costs of assembly relations, the optimal sub-assembly sequences will be detected to compose the optimal assembly sequence. Comparing with the iterative computations of the intelligent algorithms, the computation time will be saved.

We also use the brutal depth-first graph search algorithm to compute the 140 feasible assembly sequences and their costs. The optimal assembly sequence is the same as that computed with the MST-based algorithm. However, the time complexity of the depth-first graph search algorithm is much bigger than that of the MST-based algorithm. In the worst case, the time complexity of the depth-first graph search algorithm is $O(n!)$ based on the weighted graph. It aims to generate all of the assembly sequences and infinite computation time and memory space will be consumed for large scale of ASP. With the MST-based algorithm, the time complexity is $O(n^3)$ for ASP. The ASP will be resolved in a polynomial computation time.

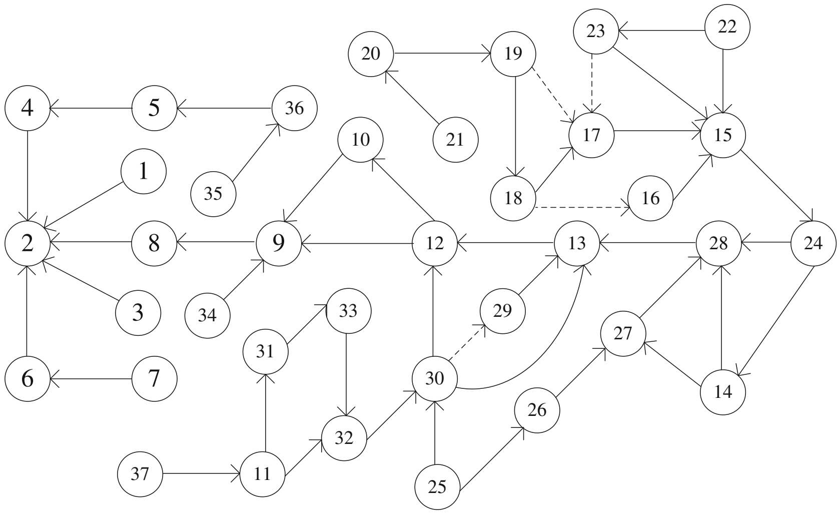
6.2 The second transmission assembly

The second transmission with 37 parts is used to show the performance of the weighted assembly precedence graph model. The assembly is given in Fig. 9, where the parts are noted with numbers. The numbers in the brackets represent the number of such parts in the assembly. Most of the parts are assembled along the $\pm X$ directions. Two parts 26 and 34 are assembled in the $\pm Y$ and the $\pm Z$ directions, respectively. Firstly, the assembly precedence between adjacent parts is extracted or picked up by the engineers. To make the task simple, we only give the necessary 46 precedence relationships among these parts. The assembly precedence graph is illustrated in Fig. 10. It does not include all of the assembly precedence among parts.

The PM is derived with respect to the assembly precedence graph. Although the assembly precedence is not complete, they are sufficient to generate the optimal assembly sequence. Secondly, the indices of the stable support, transmission, connections, DOF, assembly direction-related, and tool-related assembly constraints between each pair of two parts are given or computed. The SM, TMM, CM, DOFM, DM, and TOM are computed and used to evaluate the cost of the assembly relations with the FAHP method. These matrices of the quantitative assembly constraints are not included in the paper due to their big sizes. In view of the assembly precedence graph, the part 2 has the most assembly relations with the other parts and it is taken as the base part. The base part 2 is assembled to the workstation. The assembly direction and tool are $-X$ and t_4, respectively. The cost to assemble the base part is appointed as 100.

The weights w_1, w_2, and w_3 are assigned as 0.5. In this example, we do three experiments by changing the w_F, w_S, and w_P. The weights w_F, w_S, and w_P demonstrate the importance of the functional, structural, and process constraints. The

Fig. 10 The necessary precedence relationships between parts



smaller the weight is, the more important the quantitative assembly constraint is. The MST-based algorithm is executed to compute the optimal assembly sequence with the minimum cost. With the three groups of w_F, w_S, and w_P, the three optimal assembly sequences are generated. The results are illustrated in Table 5. WS notes the workstation to assemble the base part. Anc. and Suc. represent the ancestor part and successor part, respectively. Dir. and Tool are the assembly direction and tool to assembly the successor part. Cost means the cost to assembly each successor part.

The optimal assembly sequences are computed with the MST-based algorithm as well as the assembly directions and

Table 5 The results of the transmission assembly

Step	1st experiment w_F=0.4, w_S=0.4 and w_P=0.2					2nd experiment w_F=0.45, w_S=0.45 and w_P=0.1					3rd experiment w_F=0.4, w_S=0.2 and w_P=0.4				
	Anc.	Suc.	Dir.	Tool	Cost	Anc.	Suc.	Dir.	Tool	Cost	Anc.	Suc.	Dir.	Tool	Cost
1	WS	2	$-X$	t_4	100	WS	2	$-X$	t_4	100	WS	2	$-X$	t_8	100
2	2	6	$-X$	t_4	18	2	6	$-X$	t_4	18	2	6	$-X$	t_8	12
3	6	7	$-X$	t_4	38	6	7	$-X$	t_4	38	2	3	$-X$	t_8	34
4	2	8	$-X$	t_4	28	2	8	$-X$	t_4	28	2	4	$-X$	t_8	34
5	2	3	$-X$	t_6	55	2	3	$-X$	t_6	47	2	5	$-X$	t_8	34
6	2	4	$-X$	t_6	37	2	4	$-X$	t_6	37	5	36	$-X$	t_8	34
7	2	5	$-X$	t_6	37	2	5	$-X$	t_6	37	5	35	$-X$	t_8	34
8	2	1	X	t_6	55	2	1	X	t_6	47	6	7	$-X$	t_2	51
9	5	36	$-X$	t_9	62	5	36	$-X$	t_4	51	2	8	$-X$	t_2	18
10	5	35	$-X$	t_9	37	5	35	$-X$	t_4	37	8	9	$-X$	t_3	73
11	8	9	$-X$	t_9	55	8	9	$-X$	t_4	55	9	10	$-X$	t_3	18
12	9	10	$-X$	t_9	28	9	10	$-X$	t_4	28	5	34	$-X$	t_9	62
13	5	34	$-X$	t_9	37	5	34	$-X$	t_4	37	9	12	$-X$	t_9	45
14	9	12	$-X$	t_9	55	9	12	$-X$	t_4	55	12	13	$-X$	t_9	46
15	12	13	$-X$	t_9	49	12	13	$-X$	t_4	49	12	29	$-X$	t_9	46
16	12	29	$-X$	t_9	49	12	29	$-X$	t_4	49	2	1	$-X$	t_9	62
17	12	30	$-Y$	t_9	67	12	30	$-Y$	t_4	59	12	30	$-Y$	t_9	74
18	30	32	$-Y$	t_9	37	30	32	$-Y$	t_6	47	30	32	$-Y$	t_9	34
19	32	33	$-Y$	t_9	37	32	33	$-Y$	t_4	47	32	33	$-Y$	t_9	34
20	33	31	$-Y$	t_9	49	32	31	$-Y$	t_4	40	33	31	$-Y$	t_9	46
21	32	11	$-Y$	t_9	49	32	11	$-Y$	t_4	49	32	11	$-Y$	t_9	46
22	11	37	$-Y$	t_9	49	11	37	$-Y$	t_4	49	11	37	$-Y$	t_9	46
23	13	28	$-X$	t_1	90	13	28	$-X$	t_4	75	13	28	$-X$	t_1	91
24	28	27	$-X$	t_4	75	28	27	$-X$	t_4	57	28	27	$-X$	t_4	79
25	28	14	$-X$	t_4	45	28	14	$-X$	t_4	45	28	14	$-X$	t_4	39
26	14	24	$-X$	t_4	55	14	24	$-X$	t_4	55	14	24	$-X$	t_4	45
27	27	26	$-Z$	t_4	95	27	26	$-Z$	t_4	87	27	26	$-Z$	t_4	91
28	30	25	$-X$	t_4	67	30	25	$-X$	t_4	59	30	25	$-X$	t_4	74
29	24	15	X	t_4	97	24	15	X	t_4	89	24	15	X	t_4	92
30	15	16	$-X$	t_4	63	15	16	$-X$	t_4	55	15	16	$-X$	t_4	67
31	15	17	$-X$	t_4	45	15	17	$-X$	t_4	45	15	17	$-X$	t_4	39
32	15	23	$-X$	t_4	37	15	23	$-X$	t_4	37	15	23	$-X$	t_4	34
33	15	22	$-X$	t_4	10	15	22	$-X$	t_4	10	15	22	$-X$	t_4	6
34	17	18	$-X$	t_4	77	22	18	$-X$	t_4	67	17	18	$-X$	t_4	63
35	18	19	$-X$	t_4	49	18	19	$-X$	t_4	49	18	19	$-X$	t_4	46
36	19	20	$-X$	t_4	49	19	20	$-X$	t_4	49	19	20	$-X$	t_4	46
37	20	21	$-X$	t_4	49	20	21	$-X$	t_4	49	20	21	$-X$	t_4	46
Sum			8	4	1936			10	4	1836			7	5	1841
Computation time			16 ms			0 ms					0 ms				

*Anc*ancestor part, *Suc* successor part, *Dir* assembly direction, *WS* workstation tools. In each step, a successor part is assembled to its ancestor part. The assembly direction, tools and cost to assemble the successor part are recorded. The changes of the assembly directions, tools and total assembly cost are summed in the bottom line in Table 5. With different weights of the quantitative assembly constraints, we obtain three optimal assembly sequences. The weights of assembly constraints demonstrate their importance in the assembly environment. It is found that the cost of the optimal assembly sequence searched in the 2nd experiment is smallest. The change of the assembly tools is minimal although the number of reorientations is the biggest. We also use the depth-first graph search algorithm to search the optimal assembly sequence based on the weighted assembly precedence graph. It is found that the search space is too big and we have to give up the task. In a few minutes, the program output thousands of assembly sequences. The memory space exceeds 800 MB in a few seconds. The number of the feasible assembly sequences is very large for the transmission with 37 parts. To reduce the number of the feasible assembly sequences, more qualitative assembly constraints need to be added. On the other hand, it will become difficult to construct the assembly precedence graph. However, the computation time is no more than 1 s with the MST-based algorithm for the transmission. Comparing with the other exact algorithms for ASP, the MST-based algorithm is much faster.

7 Conclusions and suggestions

Considering the assembly relations between parts, an assembly is apt to represent as an assembly relation (or liaison) graph. However, a large number of assembly sequences exist in the assembly relation graph. To reduce the complexity of ASP, the qualitative assembly constraints are summarized and added to the assembly relation graph to form the assembly precedence graph. The assembly precedence graph guarantee to find the feasible assembly sequences whereas the optimal assembly sequence is still difficult to seek due to their large number. The weighted assembly precedence graph is proposed as another representation for ASP. The helpful assembly constraints including the qualitative and quantitative constraints are able to integrate into the model. The qualitative assembly constraints prove the feasibility of the assembly sequences. The quantitative assembly constraints are used as the heuristic information to detect the optimal sub-sequences and sequence. The ASP can be resolved with the current efficient methods and algorithms based on the weighted assembly precedence graph.

In the weighted assembly precedence graph, the assembly relations, geometrical constraints and a portion of the process constraints are taken as the qualitative assembly constraints. They denote the assembly precedence and represented as the arrowed edges in the weighted assembly precedence graph. The functional, structural and some other process constraints are used as the quantitative assembly constraints. The indices of these constraints are defined and used to compute the cost of assembly relations with the FAHP. The costs are taken as the heuristic information to generate the optimal sub-assembly sequences and sequence. A MST-based algorithm is designed to search the optimal assembly sequence based on the weighted assembly precedence graph. The time complexity of the algorithm is $O(n^3)$. The complex ASP will be resolved in a polynomial computation time.

The research mainly focuses on the sequential assembly process whereas the concurrent assembly is not considered. The assembly directions and tools are also simplified. In the future, the method will be expanded to the concurrent assembly procedures. The assembly directions and tools will be handled more precisely for actual assembly process.

Acknowledgments The authors acknowledge the project supported by NSFC (Grant No. 51205129) and the funds supported by the Fundamental Research Funds for the Central Universities (No. 2015ZD10). The work benefits from the facilities of National Key Laboratory of New Energy Power System and the Beijing Key Laboratory of New and Renewable Energy, North China Electric Power University, Beijing, China.

References

1. Boothroyd G, Dewhurst P, Knight A (2002) Product design for manufacture and assembly. Marcel Dekker, Inc, New York
2. Nof SY, Wilhelm WE, Warneke HI (1997) Industrial assembly. Chapman & Hall, London
3. Gottschlich S, Ramos C, Lyons D (1994) Assembly and task planning: a taxonomy. IEEE Robot Autom Mag 1(3):4–12
4. de Mello Homem LS, Lee S (1991) Computer-aided mechanical assembly planning. Kluwer Academic Publishers, London
5. Romeo MM, Lee HS, Luong KA (2006) A genetic algorithm for the optimisation of assembly sequences. Comput Ind Eng 50(4): 503–527
6. Bourjault A, Lhote A (1986) Modeling an assembly process. IEEE Int Conf Autom Manuf Ind 20(2):183–198
7. Homem De Mello L, Sanderson A (1990) And/or graph representation of assembly plans. IEEE Trans Robot Autom 6(2):188–199
8. Dini G, Santochi M (1992) Automated sequencing and subassembly detection in assembly planning. Ann CIRP 41(1):1–4
9. Dutta D, Woo TC (1995) Algorithm for multiple disassembly and parallel assemblies. ASME J Eng Ind 117(2):102–109
10. Lambert AJD, Gupta SM (2005) Disassembly modeling for assembly, maintenance, reuse, and recycling. CRC Press, Florida
11. Makris S, Pintzos G, Rentzos L, Chryssolouris G (2013) Assembly support using AR technology based on automatic sequence generation. CIRP Ann Manuf Technol 62(1):9–12
12. Wang Y, Liu JH (2010) Chaotic particle swarm optimization for assembly sequence planning. Robot Comput Integr Manuf 26(2): 212–222
13. Zhu XW, Hu SJ, Koren Y, Huang NJ (2012) A complexity model for sequence planning in mixed-model assembly lines. J Manuf Syst 31(2):121–130
14. Su Q (2009) A hierarchical approach on assembly sequence planning and optimal sequences analyzing. Robot Comput Integr Manuf 25(1):224–234

15. Hu SJ, Ko J, Weyand L, ElMaraghy HA et al (2011) Assembly system design and operations for product variety[J]. CIRP Ann Manuf Technol 60(2):715–733
16. Zha XF (2000) An object-oriented knowledge based Petri net approach to intelligent integration of design and assembly planning. Artif Intell Eng 14(1):83–112
17. De Fazio TL, Whitney DE (1987) Simplified generation of all mechanical assembly sequences. IEEE J Robot Autom 3(6):640–658
18. Baldwin DF, Abell TE, De Fazio TL, Whitney DE (1991) An integrated computer aid for generating and evaluating assembly sequences for mechanical products. IEEE Trans Robot Autom 7(1): 78–94
19. Zhou W, Zheng JR, Wang JF (2011) Nested partitions method for assembly sequences merging. Expert Syst Appl 38(8):9918–9923
20. Homem de Mello LS, Sanderson A (1991) A correct and complete algorithm for the generation of mechanical assembly sequences. IEEE Trans Robot Autom 7(2):228–240
21. Su Q (2007) Computer aided geometric feasible assembly sequence planning and optimizing. Int J Adv Manuf Technol 33(1/2):48–57
22. Hsu YY, Tai PH, Wang MW, Chen WC (2011) A knowledge-based engineering system for assembly sequence planning. Int J Adv Manuf Technol 55(5–8):763–782
23. Lambert AJD (2005) Generating disassembly sequences using exact and heuristic methods applied to disassembly precedence graphs. The 6th IEEE International Symposium on Assembly and Task Planning: From Nano to Macro Assembly and Manufacturing, NY, pp 47–52
24. Qu SP, Jiang ZH, Tao NR (2013) An integrated method for block assembly sequence planning in shipbuilding. Int J Adv Manuf Technol 69(5–8):1123–1135
25. Gao L, Qian WR, Li XY, Wang JF (2009) Application of memetic algorithm in assembly sequence planning. Int J Adv Manuf Technol 49(9–12):1175–1184
26. Li MY, Wu B, Hu YM, Jin C, Shi TL (2013) A hybrid assembly sequence planning approach based on discrete particle swarm optimization and evolutionary direction operation. Int J Adv Manuf Technol 68(1–4):617–630
27. Niu XN, Ding H, Xiong Y (2003) A hierarchical approach to generating precedence graph for assembly planning. Int J Mach Tools Manuf 43(14):1473–1486
28. Yu JP, Wang CG (2013) A max–min ant colony system for assembly sequence planning. Int J Adv Manuf Technol 67(9–12):2819–2835
29. Li SQ, Liu Y, Wang JF, Zeng HM (2014) An intelligent interactive approach for assembly process planning based on hierarchical classif ication of parts. Int J Adv Manuf Technol 70(9–12): 1903–1914
30. Ramos C, Rocha J, Vale Z (1998) On the complexity of precedence graphs for assembly and task planning. Comput Ind 36(1–2): 101–111
31. Zhou XM, Du PG (2008) A model-based approach to assembly sequence planning. Int J Adv Manuf Technol 39(9–10):983–994
32. Gu TL, Xu ZB, Yang ZF (2008) Symbolic OBDD representations for mechanical assembly sequences. Comput Aided Des 40(4):411–421
33. Banerjee A, Banerjee P (2000) A behavioral scene graph for rule enforcement in interactive virtual assembly sequence planning. Comput Ind 42(2–3):147–157
34. Bai YW, Chen ZN, Bin HZ, Hun J (2005) An effective integration approach toward assembly sequence planning and evaluation. Int J Adv Manuf Technol 27(1/2):96–105
35. Choi YK, Lee DM, Cho YB (2012) An approach to multi-criteria assembly sequence planning using genetic algorithms. Int J Adv Manuf Technol 42(1):180–188
36. Tseng YJ, Chen JY, Huang FY (2010) A multi-plant assembly sequence planning model with integrated assembly sequence planning and plant assignment using GA. Int J Adv Manuf Technol 48(1–4):333–345
37. Zhou W, Zheng JR, Yan JJ, Wang JF (2011) A novel hybrid algorithm for assembly sequence planning combining bacterial chemotax is with genetic algorithm. Int J Adv Manuf Technol 52(5–8):715–724
38. Ibrahim I, Ibrahim Z, Ahmad H, Jusof M, Yusof Z, Nawawi SWI, Mubin M (2015) An assembly sequence planning approach with a rule-based multi-state gravitational search algorithm. Int J Adv Manuf Technol. doi:10.1007/s00170-015-6857-0
39. Fathi M, Ghobakhloo M (2014) A technical comment on "a review on assembly sequence planning and assembly line balancing optimisation using soft computing approaches". Int J Adv Manuf Technol 71(9–12):2033–2042
40. Behdad S, Berg L, Vance J, Thurston D (2014) Immersive computing technology to investigate tradeoffs under uncertainty in disassembly sequence planning. J Mech Des 136(7):1–9
41. Scholl A, Becker C (2006) State-of-the-art exact and heuristic solution procedures for simple assembly line balancing. Eur J Oper Res 168(3):666–693
42. Wang Y (2014) The hybrid genetic algorithm with two local optimization strategies for traveling salesman problem. Comput Ind Eng 70(4):124–133
43. West DB (2006) Introduction to graph theory. China Machine Press, Beijing
44. Morato C, Kaipa KN, Gupta SK (2013) Improving assembly precedence constraint generation by utilizing motion planning and part interaction clusters. Comput Aided Des 45(11):1349–1364
45. van Holland W, Bronsvoort WF (2000) Assembly features in modeling and planning. Robot Comput Integr Manuf 16(4): 277–294
46. Wang Y, Liu JH, Li LS (2009) Assembly sequences merging based on assembly unit partitioning. Int J Adv Manuf Technol 45(7–8): 808–820
47. Michela B, Elisabetta B, Leila DF, George N (1996) Generating assembly and machining sequences from the face-to-face composition model. Comput Aided Des 28(2):101–112
48. Lee S, Yi C (1995) Assemblability evaluation based on tolerance propagation[C]. Proc IEEE Int Conf Robot Autom, Leuven, Belgium 21–27:1593–1598
49. Lee S (1994) Subassembly identification and evaluation for assembly planning. IEEE Trans Syst Man Cybernetics 24(3):493–503
50. Laarhoven P, Pedrycz W (1983) A fuzzy extension of Saaty's priority theory. Fuzzy Sets Syst 11(3):229–241
51. Saaty TL (1980) The analytic hierarchy process. McGraw-Hill, New York
52. Wang JF, Liu JH, Zhong XF (2005) A novel ant colony algorithm for assembly sequence planning. Int J Mach Tools Manuf 25(11–12):1137–1143
53. Gao L, Qian WR, Li XY, Wang JF (2010) Application of memetic algorithm in assembly sequence planning. Int J Mach Tools Manuf 49(9–12):1175–1184