

## A New Effective Algorithm for Mechanical Assembly Sequence Planning

Michał Szaśiadek<sup>1</sup>, Maciej Niedziela<sup>2</sup>, Waldemar Woźniak<sup>3</sup>,  
Tomasz Jachowicz<sup>4\*</sup>, Nikoleta Mikušová<sup>5</sup>

<sup>1</sup> Institute of Mechanical Engineering, University of Zielona Góra, 4 Prof. Szafran Street, 65-516 Zielona Góra, Poland

<sup>2</sup> Institute of Mathematics, University of Zielona Góra, 4 Prof. Szafran Street, 65-516 Zielona Góra, Poland

<sup>3</sup> Institute of Materials and Biomedical Engineering, University of Zielona Góra, 4 Prof. Szafran Street, 65-516 Zielona Góra, Poland

<sup>4</sup> Department of Polymer Processing, Faculty of Mechanical Engineering, Lublin University of Technology, 38 Nadbystrzycka Street, 20-618 Lublin, Poland

<sup>5</sup> Institute of Logistics and Transport, Faculty of Mining, Ecology, Process Control and Geotechnology, Technical University of Kosice, Letna 9, 042 00 Kosice, Slovakia

\* Corresponding author's e-mail: [t.jachowicz@pollub.pl](mailto:t.jachowicz@pollub.pl)

### ABSTRACT

This paper presents a new approach for searching the most favourable mechanical assembly sequence. The proposed method is described and its basic modules are presented. The concept is based on the assumption that the method should help the engineer in specifying the best assembly sequence with the short possible time calculation, taking into account the principles of design for assembly at an early stage of development of the product design. The algorithm for generating an accurate assembly sequence was described in detail and its application on real-life example was presented.

**Keywords:** assembly sequence planning, algorithm, engineering design process, design for assembly.

### INTRODUCTION

Machines and mechanical devices assembly process, consisting in combining of their individual components in the assembly units of a higher order, up to the complete assembly, is one of the most important step in the process of product implementation. According to literature, the assembly process constitutes about 30-50% of the cost of the product manufacturing [1, 12, 15]. Labour intensity of the assembly process grows along with the number of product parts, and the proper conduct of this process depends on the order and the correct connection of the parts [5, 17, 19]. Hence, it is important to determine the best sequence of connecting the components (assembly sequence)

of the product [10, 19, 30]. With a large number of components, the issue of determining the best assembly sequence is complex [9, 24]. Determining the optimal (in accordance with the assumed requirements) assembly sequence without computer help is much more difficult with complex products.

Determining the optimal assembly sequence with simultaneous construction evaluation including the assembly requirements is significant, as noted above, especially at an early stage of the product design development, in its design conceptual phase [7, 13, 21]. It is then possible to design the product concurrently with its assembly process, which greatly speeds up the product final implementation. Moreover, such an approach allows early identification of flaws in the construction (or

its components) and necessary (from the assembly viewpoint) corrections at the design stage.

In the literature one can find a lot of views on the assembly sequence generating [2, 31] formulated an algorithm for generating all the permissible assembly sequences, which was based on a list of questions. These questions resulted in obtaining relations for the analysed constituents of a product. A similar algorithm is the one by De Fazio and Whitney [8], however it is based on determining relations for assembly operations, which characterise pairs of combined parts. Sanderson and Homem de Mello [22] developed an algorithm allowing to build a relational model, on the basis of which, using graph operations (graph cuts of and/or type), a set of all the possible assembly sequences was gained. Other studies related to determining the assembly sequences use for instance exploded views of the products, artificial intelligence methods. Recently, you can see a different approach to generating sequences for assembly, such as the use of artificial intelligence [16], augmented reality [18], harmony search algorithm [14], Taguchi method [29], ant colony [28] and other algorithms [20]. All the above approaches are applicable in the case of a previously developed product structure. Similarly, other approaches make the analysis of the assembly process possible, but only after the manufacturing stage, when the product components are ready and their assembly process is planned [3, 11, 23]. In this case any construction changes are really expensive and involve redesign of the product and repeated production of components which have undergone construction changes.

## CHARACTERISTICS OF THE METHOD OF ASSEMBLY SEQUENCE PLANNING

The implemented method is proposed to determine the best assembly sequence of the product components. The assumption is that the method should be applied at the early stage of design (preliminary design) once the structure of the product has been determined. At this stage, after the proposed method had been applied, the combinations for joining parts of the product can be generated. Also, it is possible to evaluate designated sequences and every combination in them due to efficiency of its assembly. As a result, after an analysis of the output (best) sequences, it is possible to change some of the design features of the product

components (or even its structure) to streamline the process of assembly. In this case, any changes to the initial solution will be implemented before the detailed design phase, without the involvement of unnecessarily large amount of time, and, above all, without the excessive cost [25, 26, 27].

The method is based on the Easyassembly algorithm, described in earlier publications [26, 27]. The distinguishing feature of the Easyassembly algorithm is the analysis of all relations between the product's components and, based on the restrictions defined, a determination and evaluation of all permissible assembly sequences. Only on this basis can the best solution - or equivalent solutions - be determined. Proposed method is aimed at determining the optimal assembly sequence, without having to search through all potential solutions.

It is assumed that application of the proposed method could contribute to: high product quality, reduction of time and cost of launching the product to the market, simplification of the process of connecting the product components, minimization of positioning, adjustment of the structure components for ease of handling, selection (or change) of the connection type to shorten the process, reduced complexity of the parts shape, reduction of the force necessary to set the elements in relation to each other, etc. [4, 6, 9].

## Problem of determining the assembly sequence

Determination of the feasible (possible) sequence assemblies is a complex issue, and as the complexity of an assembly (machine) increases it leads to the phenomenon of the so-called 'combinatorial explosion'. The number of possible combinations for increasing number of components grows according to the factorial function (permutations of  $n$ -element set, where  $n$  is the number of the product components). Hence, when the parts number is greater, a systematic analysis of all variants is virtually impossible without a proper instrument. Such an analysis is not necessary after all, as there are plenty of unfeasible and not recommended variants in this theoretical set. Effective assembly sequence generation algorithm should be characterised primarily by elimination of all the solutions from the result set, which in terms of actual implementation are unfeasible or incorrect. It can be accomplished by defining the appropriate constraint requirements.

### General characteristics of the method

The proposed method for determining the set of feasible assembly sequences and their evaluation and selection belongs to the category of classic formal methods. In this method four basic and implemented sequentially modules can be distinguished. In the first module the product design is mapped in the form of design structure matrix  $P$ . In the second module, all relations stored in the  $P$  matrix are evaluated. The result of the implementation of the first two modules is a record of the product design structure in the form of  $P$  matrix and assigning an evaluation indicator to each relation stored in this matrix. In the third module precedence constraints for the assembly connections are defined (generated on the basis of the  $P$  matrix). They are taken into consideration to determine the feasible assembly process. The last module of the method is the algorithm for generating the optimal of feasible assembly sequence. The next sections describes the method's modules.

Explanations of the basic modules of the method and the accompanying notation of entries, as well as the operation of the algorithm for generating the most advantageous assembly sequence, will be presented in the well-known example from the literature [22, 30] of a 6-element, ball-point pen, as shown in Figure 1. It was assumed that all the components of the pen are solid. They are marked, respectively:  $X_1$  - Cap,  $X_2$  - Head,  $X_3$  - Body,  $X_4$  - Ink,  $X_5$  - Tube,  $X_6$  - Button.

#### Notation in the product structure design

All contact relations between the components of the product are identified on the basis of product design structure. Contact relation is understood as the possibility of combining two parts. Established relations (connections) are stored in the form of a graph (connection diagram) and the corresponding matrix  $P$ , called further the relationship matrix or structure design matrix, given in the following form:

$$P(i, j) = \begin{cases} 1, & \text{if connection of element } X_i \text{ to } X_j \text{ is possible} \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

It is assumed the notation  $X_i \rightarrow X_j$  to record connection between the components  $X_i$  and  $X_j$ . This matrix has a size of  $n \times n$ , where  $n$  is the number of the product components. If there is no relationship between the elements (or if it is not possible connect two elements) no type of relation is assigned and the corresponding entry of matrix  $P$  stays empty. The suitable form of the relationship matrix  $P$  with all the possible connections between the components for the six-parts ball-point pen mentioned earlier (Fig. 1) we can read as:

$$P = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \quad (2)$$

In order to evaluate an optimal assembly sequence we apply the weights matrix  $Q$  with positive components:

$$Q(i, j) = \begin{cases} q_{ij}, & \text{if } P(i, j) = 1 \\ \infty, & \text{if } P(i, j) = 0 \end{cases} \quad (3)$$

The finite indicator  $q_{ij}$  describes the weight of connection  $X_i \rightarrow X_j$  and it is usually designated in the literature [4]. It was developed on the basis of experts' knowledge and multiple analysis conducted in actual companies, and described in publications [1, 4, 25]. Thus the weight matrix for the ball-point pen example has the following form:

$$Q = \begin{bmatrix} \infty & \infty & 1.0 & \infty & \infty & \infty \\ \infty & \infty & 1.4 & 2.0 & 1.4 & \infty \\ 1.0 & 1.4 & \infty & \infty & \infty & 1.4 \\ \infty & 2.0 & \infty & \infty & 4.0 & \infty \\ \infty & 1.4 & \infty & 4.0 & \infty & \infty \\ \infty & \infty & 1.4 & \infty & \infty & \infty \end{bmatrix} \quad (4)$$

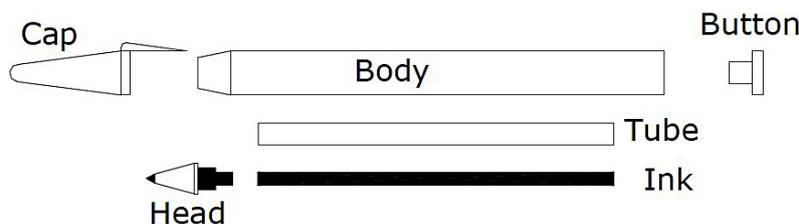


Fig. 1. The structure of the ball-point pen: 1 - cap, 2 - head, 3 - body, 4 - ink, 5 - tube, 6 - button

**Defining the constraints**

Determination of the correct assembly sequences requires appropriate precedence constraints. They are related to the set of connections recorded in the *P* matrix and defined in the list *L*. This type of constraints are called blocking connections and prevent or limit of getting a complete assembly in the later course of the assembly process. They have a direct influence on generating correct order of combining the parts, in terms of the selection completeness. The list of blocking connections *L* is characteristic of those preceding connections, which prevent the realisation of the connection for which they are defined. This way the possibility of incorrect sequence when combining the parts is eliminated. It is assumed the blocking connections need to be defined with the operator ‘AND’ ( $\wedge$ ) and ‘OR’ ( $\vee$ ). In the first case, assigning the ‘AND’ operator to the blocking connections ( $I_1 \wedge I_2 \wedge \dots \wedge I_n$ ) means that connection  $I_n$ , for which the blocking connections are defined, can be executed before every blocking connection is made. Thus, it is possible to make  $n - 1$  blocking connections before the connection  $I_n$ , for which blocking connections were defined. If all the blocking connections are executed, it is impossible to achieve complete assembly

of the whole product because realisation of the connection  $I_n$  is blocked. In the second case, assigning the ‘OR’ operator to the blocking connections ( $I_1 \vee I_2 \vee \dots \vee I_n$ ) means that connection  $I_n$ , for which the blocking connections are defined, has to be executed before any of them. Even if one of the blocking connections is made, it is impossible to achieve complete assembly of the whole product because realisation of the connection  $I_n$  is blocked. Moreover, it is possible to define blocking connection sequences (with the ‘AND’ operator) separating them by the use of the ‘OR’ operator. The suitable constraints for our example model are presented in the Table 1 and 2.

**The new algorithm for mechanical assembly sequence planning**

In order to realize computer aided analysis for mechanical assembly sequence planning, the following algorithm was developed. On the basis of the relationship matrix *P*, the weights matrix *Q* and the list *L* of blocking connections AND/OR, it allows us to determine complete and correct assembly sequence. The executed sequence of connections of parts and the minimum value of cost function  $W^*$  are given as an output of algorithm (Fig. 2).

**Table 1.** The list *L* of defined OR type restrictions for the six components ball-point pen assembly

Assembly connection	Blocking connections of OR type
$X_2 \rightarrow X_3$	$X_1 \rightarrow X_3 \vee X_3 \rightarrow X_1$
$X_2 \rightarrow X_4$	$X_2 \rightarrow X_3 \vee X_3 \rightarrow X_2$
$X_2 \rightarrow X_5$	$X_2 \rightarrow X_3 \vee X_3 \rightarrow X_2$
$X_3 \rightarrow X_2$	$X_1 \rightarrow X_3 \vee X_3 \rightarrow X_1$
$X_4 \rightarrow X_2$	$X_2 \rightarrow X_3 \vee X_3 \rightarrow X_2$
$X_5 \rightarrow X_2$	$X_2 \rightarrow X_3 \vee X_3 \rightarrow X_2$

**Table 2.** The list *L* of AND type restrictions for the six components ball-point pen assembly

Assembly connection	Blocking connections of AND type
$X_2 \rightarrow X_4$	$(X_2 \rightarrow X_3 \wedge X_3 \rightarrow X_6) \vee (X_3 \rightarrow X_2 \wedge X_6 \rightarrow X_3)$ $\vee$ $(X_2 \rightarrow X_3 \wedge X_6 \rightarrow X_3) \vee (X_3 \rightarrow X_2 \wedge X_3 \rightarrow X_6)$
$X_2 \rightarrow X_5$	$(X_2 \rightarrow X_3 \wedge X_3 \rightarrow X_6) \vee (X_3 \rightarrow X_2 \wedge X_6 \rightarrow X_3)$ $\vee$ $(X_2 \rightarrow X_3 \wedge X_6 \rightarrow X_3) \vee (X_3 \rightarrow X_2 \wedge X_3 \rightarrow X_6)$
$X_4 \rightarrow X_2$	$(X_2 \rightarrow X_3 \wedge X_3 \rightarrow X_6) \vee (X_3 \rightarrow X_2 \wedge X_6 \rightarrow X_3)$ $\vee$ $(X_2 \rightarrow X_3 \wedge X_6 \rightarrow X_3) \vee (X_3 \rightarrow X_2 \wedge X_3 \rightarrow X_6)$
$X_5 \rightarrow X_2$	$(X_2 \rightarrow X_3 \wedge X_3 \rightarrow X_6) \vee (X_3 \rightarrow X_2 \wedge X_6 \rightarrow X_3)$ $\vee$ $(X_2 \rightarrow X_3 \wedge X_6 \rightarrow X_3) \vee (X_3 \rightarrow X_2 \wedge X_3 \rightarrow X_6)$

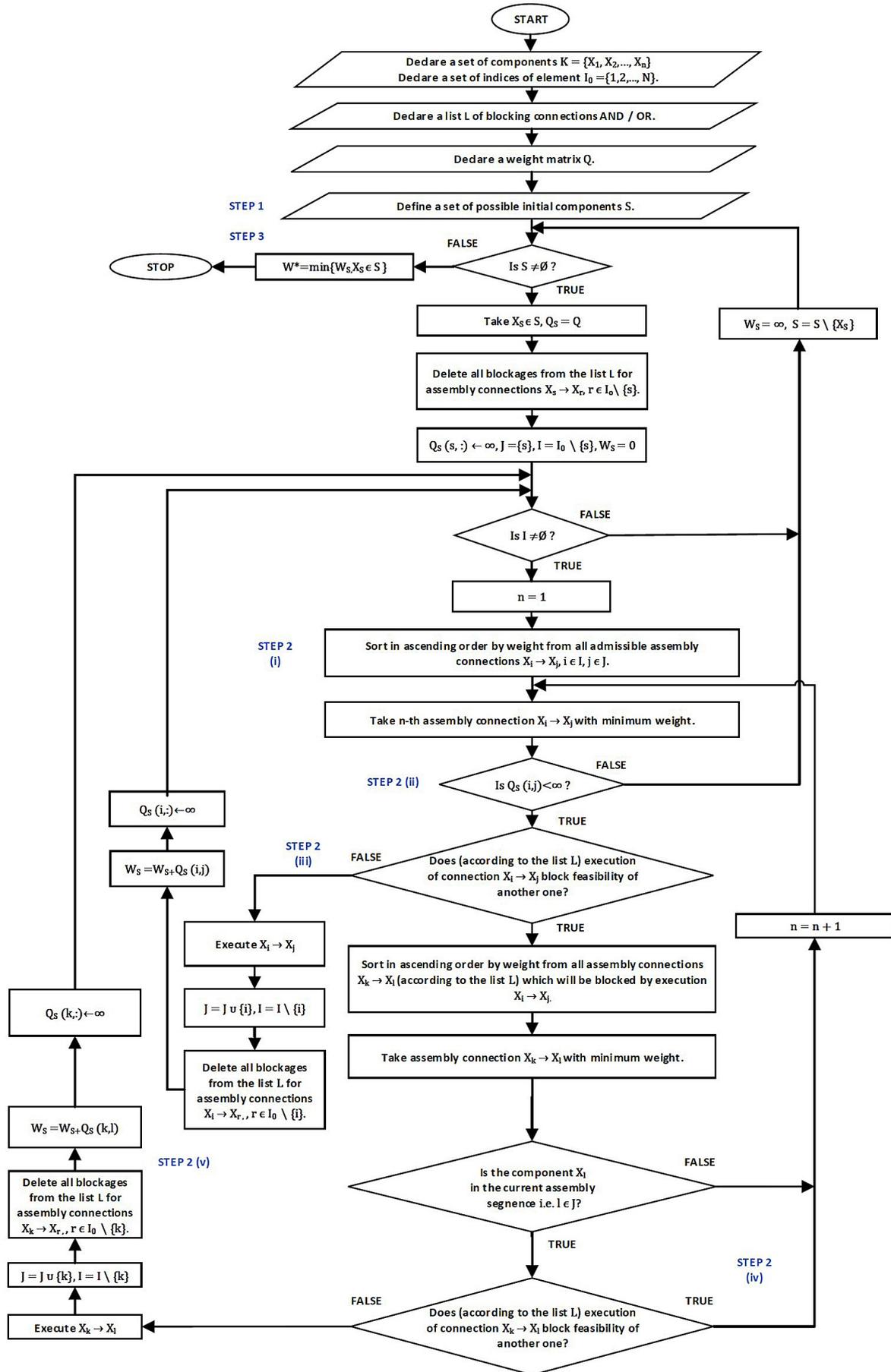


Fig. 2. Algorithm flowchart

The main steps of Algorithm:

1. Defining a set  $S$  of possible initial components. If only there exists in the column  $i$  of matrix  $Q$  element with finite value, the component  $X_i$  could be initial in the assembly process.
2. For each  $X_S \in S$  find the best assembly sequence and calculate the corresponding value of cost function  $W_S$ :
  - i. Find as a sorted vector  $Q_{min}$  of  $Q_S(I,J)$  from minimum to maximum value, where  $I$  and  $J$  denote sets of indices of components which are not yet and are already used in the current assembly sequence. Take  $n = 1$ .
  - ii. Take the assembly connection  $X_i \rightarrow X_j$  with the value of weight matrix  $Q_{min} = Q_S(i,j)$ . If  $Q_S(i,j)$  is not finite, take  $W_S = \infty$  and break the construction process of the assembly sequence for given  $X_S$ .
  - iii. If according to the list  $L$  execution of connection  $X_i \rightarrow X_j$  does not lock feasibility of another one, execute  $X_i \rightarrow X_j$ .
  - iv. If according to the list  $L$  exist connections of components which will be blocked by execution  $X_i \rightarrow X_j$ , force those with minimum weight  $X_k \rightarrow X_l$ . If it is not feasible, i.e. component  $X_l$  is not yet in the current assembly sequence ( $l \notin J$ ) or execution of connection  $X_k \rightarrow X_l$  block feasibility of another one, take  $n = n + 1$  and go to Step 2 (ii).
  - v. After joining component  $X_m$  to the current assembly sequence delete all blockages from the list  $L$  for assembly connections  $X_m \rightarrow X_r, r \in \{1, \dots, m - 1, m + 1, \dots, n\}$ . Update sets  $I$  and  $J$ , value  $W_S$  and matrix  $Q_S$ .
3. Find the best assembly sequence with minimum value of cost function  $W^* = \{W_S, X_S \in S\}$ .

### Case study

A formal description of assembly detection algorithm of mechanical assembly sequence planning can be illustrated with the help of the six components ball-point pen assembly from Fig. 1. Then, as depicted in the section 2, the relationship and the weights matrices have the following forms:

$$P = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}, \quad Q = \begin{bmatrix} \infty & \infty & 1.0 & \infty & \infty & \infty \\ \infty & \infty & 1.4 & 2.0 & 1.4 & \infty \\ 1.0 & 1.4 & \infty & \infty & \infty & 1.4 \\ \infty & 2.0 & \infty & \infty & 4.0 & \infty \\ \infty & 1.4 & \infty & 4.0 & \infty & \infty \\ \infty & \infty & 1.4 & \infty & \infty & \infty \end{bmatrix} \quad (5)$$

and the list of blocking connections is given in the Table 1 and 2.

STEP (1): We start our algorithm by determining the set of possible initial components:

$$S = \{X_1, X_2, X_3, X_4, X_5, X_6\} \quad (6)$$

STEP (2): Let us take as an initial part  $X_1$  (base component). Then we define:

$$J = \{1\}, \quad I = \{2, 3, 4, 5, 6\}, \quad Q_1 = \begin{bmatrix} \infty & \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & 1.4 & 2.0 & 1.4 & \infty \\ 1.0 & 1.4 & \infty & \infty & \infty & 1.4 \\ \infty & 2.0 & \infty & \infty & 4.0 & \infty \\ \infty & 1.4 & \infty & 4.0 & \infty & \infty \\ \infty & \infty & 1.4 & \infty & \infty & \infty \end{bmatrix}, \quad W_1 = 0 \quad (7)$$

Minimum value of submatrix  $Q_1(I,J)$  is equal  $Q_1(3,1) = 1.0$  (step 2(ii)). According to the list  $L$  (Table 1) we notice that execution of connection  $X_3 \rightarrow X_1$  will block feasibility of connections  $X_2 \rightarrow X_3$  and  $X_3 \rightarrow X_2$ . Then the algorithm try to force those with minimum weight (step 2(iv)). Since elements  $X_2$  and  $X_3$  are not yet in the current assembly sequence, the algorithm is not able to force neither of these connections. Hence we take the second minimum element of  $Q_1(I,J)$  (step 2(ii)). However it is not finite and the process of assembly sequence generation is incomplete and broken.

Let us take as an initial part  $X_2$  (base component). This time we begin with:

$$J = \{2\}, \quad I = \{1, 3, 4, 5, 6\}, \quad Q_2 = \begin{bmatrix} \infty & \infty & 1.0 & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty \\ 1.0 & 1.4 & \infty & \infty & \infty & 1.4 \\ \infty & 2.0 & \infty & \infty & 4.0 & \infty \\ \infty & 1.4 & \infty & 4.0 & \infty & \infty \\ \infty & \infty & 1.4 & \infty & \infty & \infty \end{bmatrix}, \quad W_2 = 0 \quad (8)$$

We get  $\min Q_2(I, J) = Q_2(3, 2) = 1.4$ , which implies assembly connection  $X_3 \rightarrow X_2$ . On the basis of the list  $L$  we notice that execution of this connection will block feasibility of connections  $X_4 \rightarrow X_2$  or  $X_5 \rightarrow X_2$ . Therefore, the algorithm forces  $X_5 \rightarrow X_2$  (step 2(iv)) and updates

$$J = \{2, 5\}, \quad I = \{1, 3, 4, 6\}, \quad Q_2 = \begin{bmatrix} \infty & \infty & 1.0 & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty \\ 1.0 & 1.4 & \infty & \infty & \infty & 1.4 \\ \infty & 2.0 & \infty & \infty & 4.0 & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & 1.4 & \infty & \infty & \infty \end{bmatrix}, \quad W_2 = 1.4 \quad (9)$$

Further,  $\min Q_2(I, J) = Q_2(3, 2) = 1.4$  gives  $X_3 \rightarrow X_2$  with minimum weight. Since the list  $L$  shows again that execution of this connection will block feasibility of connections  $X_4 \rightarrow X_2$ , the algorithm returns  $X_4 \rightarrow X_2$  with values

$$J = \{2, 4, 5\}, \quad I = \{1, 3, 6\}, \quad Q_2 = \begin{bmatrix} \infty & \infty & 1.0 & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty \\ 1.0 & 1.4 & \infty & \infty & \infty & 1.4 \\ \infty & \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & 1.4 & \infty & \infty & \infty \end{bmatrix}, \quad W_2 = 3.4 \quad (10)$$

Proceeding next lines of the algorithm we have the following results:

Firstly, on the basis of  $\min Q_2(I, J) = Q_2(3, 2) = 1.4$  we force assembly connection  $X_3 \rightarrow X_2$  (step 2(iii)) and get  $J = \{2, 3, 4, 5\}, I = \{1, 6\}, W_2 = 4.8$  and

$$Q_2 = \begin{bmatrix} \infty & \infty & 1.0 & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & 1.4 & \infty & \infty & \infty \end{bmatrix} \quad (11)$$

Secondly, the minimum value of submatrix  $Q_2(I, J)$  is equal  $Q_2(1, 3) = 1.0$ , which implies connection  $X_1 \rightarrow X_3$  and modifies the following values

$$J = \{1, 2, 3, 4, 5\}, \quad I = \{6\}, \quad Q_2 = \begin{bmatrix} \infty & \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & 1.4 & \infty & \infty & \infty \end{bmatrix}, \quad W_2 = 5.8 \quad (12)$$

Finally, from  $\min Q_2(I, J) = Q_2(6, 3) = 1.4$  ( $X_6 \rightarrow X_3$ ) we finish step 2 of the algorithm having get  $J = \{1, 2, 3, 4, 5, 6\}, I = \emptyset$  with the final value of cost function  $W_2 = 7.2$ . Thus, the complete assembly sequence of the product with base component  $X_2$  is given by

$$X_2, X_5 \rightarrow X_2, X_4 \rightarrow X_2, X_3 \rightarrow X_2, X_1 \rightarrow X_3, X_6 \rightarrow X_3 \tag{13}$$

Following the same procedure, we continue our algorithm choosing as an initial parts  $X_4$  and  $X_5$ . Then we will get the following result:

**Base component  $X_4$ :** the final value of cost function  $W_4 = 7.2$ , the complete assembly sequence of the product

$$X_4, X_2 \rightarrow X_4, X_5 \rightarrow X_2, X_3 \rightarrow X_2, X_1 \rightarrow X_3, X_6 \rightarrow X_3 \tag{14}$$

**Base component  $X_5$ :** the final value of cost function  $W_5 = 7.2$ , the complete assembly sequence of the product

$$X_5, X_2 \rightarrow X_5, X_4 \rightarrow X_2, X_3 \rightarrow X_2, X_1 \rightarrow X_3, X_6 \rightarrow X_3 \tag{15}$$

On the other hand, if we choose as base component  $X_3$  or  $X_6$ , the process of assembly sequence generation will be incomplete. In the first case we begin with

$$J = \{3\}, \quad I = \{1, 2, 4, 5, 6\}, \quad Q_3 = \begin{bmatrix} \infty & \infty & 1.0 & \infty & \infty & \infty \\ \infty & \infty & 1.4 & 2.0 & 1.4 & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty \\ \infty & 2.0 & \infty & \infty & 4.0 & \infty \\ \infty & 1.4 & \infty & 4.0 & \infty & \infty \\ \infty & \infty & 1.4 & \infty & \infty & \infty \end{bmatrix}, \quad W_3 = 0 \tag{16}$$

Minimum value of submatrix  $Q_3(I, J)$  is equal  $Q_3(1,3) = 1.0$  ( $X_1 \rightarrow X_3$ ). According to the list  $L$  execution of  $X_1 \rightarrow X_3$  will block feasibility of connection  $X_2 \rightarrow X_3$  (step 2(iv)). Next, the algorithm is not able to force it because execution of  $X_2 \rightarrow X_3$  blocks feasibility of another connections (Table 1). Since components  $X_2, X_4$  and  $X_5$  are not in the current assembly sequence, we take second minimum value of submatrix  $Q_3(I, J)$ . Then, on the basis of  $\min Q_3(I, J) = Q_2(6,3) = 1.4$  we execute  $X_6 \rightarrow X_3$  receiving

$$J = \{3, 6\}, \quad I = \{1, 2, 4, 5\}, \quad Q_3 = \begin{bmatrix} \infty & \infty & 1.0 & \infty & \infty & \infty \\ \infty & \infty & 1.4 & 2.0 & 1.4 & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty \\ \infty & 2.0 & \infty & \infty & 4.0 & \infty \\ \infty & 1.4 & \infty & 4.0 & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty \end{bmatrix}, \quad W_3 = 1.4 \tag{17}$$

However, sorting  $Q_3(I, J)$  in ascending order by finite weight we notice again, that execution each of two assembly connections  $X_1 \rightarrow X_3$  or  $X_2 \rightarrow X_3$  will not be successful. Therefore, the algorithm will break the process of assembly sequence generation assigning values  $W_3 = \infty$ . If we choose as base component  $X_6$  the process of assembly sequence generation will be also incomplete. Despite the algorithm executes the first connection  $X_3 \rightarrow X_6$  with weight  $Q_6(3,6) = 1.4$ , it try to force again either of connections  $X_1 \rightarrow X_3$  or  $X_2 \rightarrow X_3$  in the next step. Hence the algorithm breaks the process of assembly sequence generation updating values  $W_6 = \infty$ .

STEP (3): Finally, on the basis of the above results we find out the minimum value of cost function

$$W^* = \min\{W_1, W_2, W_3, W_4, W_5, W_6\} = 7.2 \tag{18}$$

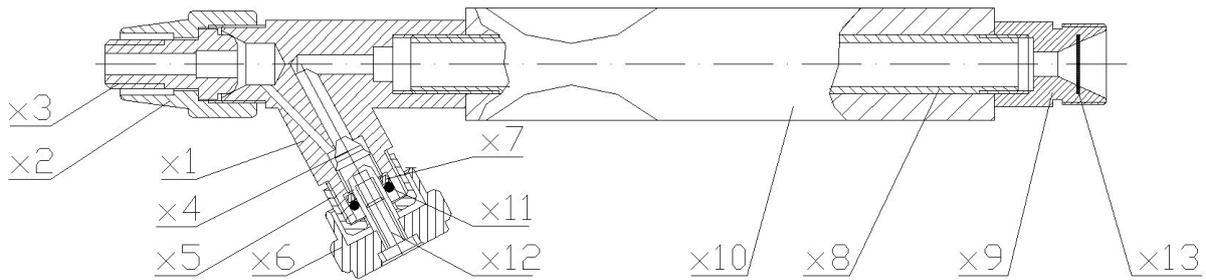
with the choice as an initial part one of the parts  $X_2, X_4$  or  $X_5$ . In this case, the algorithm returns three possible assembly sequence of the product represented by

$$\begin{aligned} X_2, X_5 \rightarrow X_2, X_4 \rightarrow X_2, X_3 \rightarrow X_2, X_1 \rightarrow X_3, X_6 \rightarrow X_3 \\ X_4, X_2 \rightarrow X_4, X_5 \rightarrow X_2, X_3 \rightarrow X_2, X_1 \rightarrow X_3, X_6 \rightarrow X_3 \end{aligned} \tag{19}$$

or

$$X_5, X_2 \rightarrow X_5, X_4 \rightarrow X_2, X_3 \rightarrow X_2, X_1 \rightarrow X_3, X_6 \rightarrow X_3 \tag{20}$$

respectively.



**Fig. 3.** The structure of the gas burner: 1 - frame, 2 - screwed sleeve, 3 - jointing sleeve, 4 - valve, 5 - contract nut, 6 - valve knob, 7 - ring, 8 - handle's connector, 9 - connector's tip, 10 - handle, 11 - O-ring, 12 - screw, 13 - mesh

**Real life example**

In this subsection we apply our algorithm to larger and real-life example. To this end, we consider the gas burner (Fig. 3) which has more complex structure.

The programming software Matlab was used to implement the algorithm into a computer program. Specifying input data we define the relationship matrix  $P$  and the weight matrix  $Q$  in the following form

$$P = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{21}$$

$$Q = \begin{bmatrix} \infty & \infty \\ 2.42 & \infty & 7.62 & \infty \\ 1.7 & \infty \\ 3.4 & \infty \\ 4.11 & \infty \\ \infty & \infty & \infty & 2.06 & \infty \\ \infty & \infty & \infty & 2.06 & \infty \\ 6.6 & \infty \\ \infty & 2.64 & \infty & \infty & \infty & \infty & \infty \\ 7.62 & \infty \\ \infty & \infty & \infty & 10.37 & \infty \\ \infty & \infty & \infty & 3.74 & \infty \\ \infty & 5.54 & \infty & \infty & \infty & \infty \end{bmatrix} \tag{22}$$

Furthermore, we determine the list  $L$  of blocking connections AND/OR (Table 3).

Simulation results of computer program are obtained for all possible initial components from the set

$$S = \{X_1, X_3, X_4, X_8, X_9\} \tag{23}$$

The optimal result is given with the initial part  $X_1$ . The minimum value of cost function is equal  $W^* = 52.44$  and complete assembly sequence of the gas burner has the following form

$$\begin{aligned} X_1, X_3 &\rightarrow X_1, X_2 \rightarrow X_3, X_4 \rightarrow X_1, X_7 \rightarrow X_4, X_{11} \rightarrow X_4, X_5 \rightarrow X_1 \\ X_6 &\rightarrow X_4, X_{12} \rightarrow X_4, X_8 \rightarrow X_1, X_{10} \rightarrow X_1, X_9 \rightarrow X_8, X_{13} \rightarrow X_9 \end{aligned} \tag{24}$$

**Table 3.** The list  $L$  of defined restrictions for the gas burner (Fig. 6)

Assembly connection	Blocking connetions of OR type	Blocking connetions of AND type
$X_3 \rightarrow X_1$	$X_2 \rightarrow X_1$	---
$X_4 \rightarrow X_1$	$X_5 \rightarrow X_1$	---
$X_5 \rightarrow X_1$	---	$(X_4 \rightarrow X_1 \wedge X_6 \rightarrow X_4)$ $\vee$ $(X_4 \rightarrow X_1 \wedge X_{12} \rightarrow X_4)$
$X_{10} \rightarrow X_1$	---	$X_8 \rightarrow X_1 \wedge X_9 \rightarrow X_8$
$X_6 \rightarrow X_4$	$X_{12} \rightarrow X_4$	---
$X_7 \rightarrow X_4$	$X_5 \rightarrow X_1 \vee X_6 \rightarrow X_4 \vee$ $X_{11} \rightarrow X_4 \vee X_{12} \rightarrow X_4$	---
$X_{11} \rightarrow X_4$	$X_5 \rightarrow X_1 \vee X_6 \rightarrow X_4 \vee$ $X_{12} \rightarrow X_4$	---

It must be also emphasized that for the remaining parts from the set  $S$ , the algorithm display the incomplete assembly sequences.

### CONCLUSIONS

The effective algorithm for determining the most advantageous assembly sequence, presented in the article, was tested on many examples, including those taken directly from industry. This algorithm has been improved in relation to the Easyassembly algorithm, previously developed. By design, the Easyassembly algorithm generates a set of all acceptable solutions and, based on an appropriate evaluation of the sequence, determines the best solution, i.e. a sequence with minimal weight. An important disadvantage of this is the high computational complexity of the algorithm. To avoid this problem, a new, effective algorithm has been proposed, as described in this article. Instead of searching the entire set of solutions, it determines the most favourable assembly sequence in each case. Taking into account the construction and the ordering conditions of individual elements of the products analysed and the relationships and/or restrictions between them, the algorithm significantly reduces the time taken to determine an effective solution, without having to generate a set for all solutions.

The model example of a six-element pen, often used for cognitive purposes in publications, presents the basic steps of the algorithm and its effectiveness. The results, that is, the assembly sequences obtained, coincide with those obtained with the Easyassembly algorithm. It is worth mentioning that the time needed to determine the most favourable sequence, is definitely much less in the algorithm proposed.

In the subsection 3.2 of this paper selection of the best assembly sequence is presented on a real-life example of the gas burner. There has been obtained one assembly sequence of thirteen constituent parts of the considered gas burner, which is characterised by the minimum value of the cost function  $W^* = 52.44$ . This result was verified and compared to the results obtained with the use of the Easyassembly method. Making a comparison between the algorithm for selection of the most favourable assembly sequence proposed in this article and the solution obtained by the Easyassembly method shows the same optimal sequence. This result is obtained in relatively short calculation time without the necessity of searching through the complete set of possible solutions.

The distinguishing feature of the algorithm, presented in this article, is the inclusion of the difficulties experienced when implementing it, due to aspects within the assembly process – in accordance with the DfA methodology (Design for Assembly) – and the use of this assessment to evaluate the assembly sequence when assessing individual assembly pairs. In addition, this algorithm allows the most advantageous solution to be quickly determined, with a simultaneous, initial indication of the base element. From a practical point of view, the algorithm developed integrally facilitates planning of the sequence of the assembly of the product, so that it is contemporaneous with its structural and technological development. This gives designers and technologists, as well as process engineers, the possibility to plan a product’s development and, ultimately, to introduce it onto the market quickly, thus ensuring competitiveness for this product.

## Acknowledgments

This work is a part of project APVV-21-0195.

## REFERENCES

6. Anderson D.M. 2010, Design for Manufacturability & Concurrent Engineering, CIM Press. doi: 10.1.1.193.5357.
7. Bahubalendruni, M.R., and B.B. Biswal. 2016. A review on assembly sequence generation and its automation. Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, 230(5), 824–838. doi:10.1177/0954406215584633.
8. Baldwin, D.F., T.E. Abell, M.C. Lui, T.L. de Fazio, and D.E. Whitney. 1991. An integrated computer aid for generating and evaluating assembly sequences for mechanical products. The International Journal of Advanced Manufacturing Technology, 7(1), 78–89. doi:10.1260/1748-3018.5.1.57.
9. Booker, J.D., M. Raines, and K.G. Swift. 2001. Designing Capable and Reliable Products. Butterworth-Heinemann. doi: 10.1016/B978-0-7506-5076-2.X5000-4.
10. Boothroyd, G., and P. Dewhurst. 1991. Design for Assembly. Boothroyd Dewhurst Inc. Wakefield. doi: 10.1109/6.275164
11. Daabub, A.M., and H.S. Abdalla. 1999. A Computer-based Intelligent System for Design for Assembly. Computers & Industrial Engineering, 37, 111–115. doi: 10.1016/S0360-8352(99)00034-0.
12. Dalvi, S. D. 2016. Optimization of Assembly Sequence Plan Using Digital Prototyping and Neural Network. Procedia Technology, 23, 414–422. doi: 10.1016/j.protcy.2016.03.045.
13. De Fazio, T.L., and D.E. Whitney. 1987. Simplified generation of all mechanical assembly sequences. IEEE Journal of Robotics and Automation, 3(6), 640–656. doi: 10.1109/JRA.1987.1087132.
14. Demoly, F., S.Gomes, B. Eynard, X.T. Yan, and D. Kiritsis. 2014. Concurrent Product Design and Assembly Sequence Planning: A Comprehensive Framework. Advances in Computers and Information in Engineering Research, Volume 1. ASME Press. doi: 10.1115/1.860328 ch12.
15. Gottipolu, R.B., and K. Ghosh. 1997. Representation and selection of assembly sequences in computer aided assembly process planning. International Journal of Production Research, 35(12), 3447–3465. doi: 10.1080/002075497194183.
16. Ketzenberg, M., G. Souza, and V.D.R. Jr. Guide. 2003. Mixed assembly and disassembly operations for remanufacturing. Production and Operations Management, 12(3), 320–335. doi: 10.1111/j.1937-5956.2003.tb00206.x. 11
17. Kim, S., J.W. Baek, S.K. Moon, and S.M. Jeon. 2015. A new approach for product design by integrating assembly and disassembly sequence structure planning. Paper presented at the 18th Asia Pacific Symposium on Intelligent and Evolutionary Systems, Springer, Cham, Volume 1, 247–257. doi: 10.1007/978-3-319-13359-1\_20.
18. Kretschmer, R., A. Pfouga, S. Rulhoff, and J. Stjepandić. 2017. Knowledge-based design for assembly in agile manufacturing by using Data Mining methods. Advanced Engineering Informatics, 33, 285–299. doi: 10.1016/j.aei.2016.12.006.
19. Li, X., K. Qin, B. Zeng, L. Gao, and J. Su. 2016. Assembly sequence planning based on an improved harmony search algorithm. The International Journal of Advanced Manufacturing Technology, 84(9–12), 2367–2380. doi: 10.1007/s00170-015-7873-9.
20. Marconnet, B., F. Demoly, D. Monticolo, and S. Gomes. 2017. An assembly oriented design and optimization approach for mechatronic system engineering. International Journal for Simulation and Multidisciplinary Design Optimization, 8, A7. doi: 10.1051/smdo/2016016.
21. Meng, G.J., and D.C. Yang. 2018. Assembly Sequence Planning Based on Genetic-ant Colony Algorithm. Modular Machine Tool & Automatic Manufacturing Technique, 4, 037. doi: 10.1108/AA-11-2016-143.
22. Murali, G.B., B.B.V.L. Deepak, and B.B. Biswal. 2017. A Novel Design for Assembly Approach for Modified Topology of Industrial Products. International Journal of Performability Engineering, 13(7), 1013. doi: 10.23940/ijpe.17.07.p2.10131019.
23. Radkowski, R., J. Herrema, and J. Oliver. 2015. Augmented reality-based manual assembly support with visual features for different degrees of difficulty. International Journal of Human-Computer Interaction, 31(5), 337–349. doi: 10.1080/10447318.2014.994194.
24. Redford, A.H., and J. Chal. 1994. Design for Assembly: Principles and Practice. McGraw-Hill. doi: 10.1002/qre.4680110514.
25. Romeo, M.Marian, H.S. Lee, and K. A. Luong. 2006. A genetic algorithm for the optimisation of assembly sequences. Computers & Industrial Engineering, 50, 503–527. doi: 10.1016/j.cie.2005.07.007.
26. Sabadka D., Molnár V., Fedorko G., Jachowicz T. Optimization of production processes using the yamazumi method. Advances in Science and Technology Research Journal 2017 (11), 4, 175–182.
27. Sanderson, A.C., L.S. Homem de Mello, and H. Zhang. 1990. Assembly sequence planning. AI Mag, 11(1), 62–81. doi: 10.1609/aimag.v11i1.824.

28. Qiang, Su. 2007. Computer aided geometric feasible assembly sequence planning and optimizing. *The International Journal of Advanced Manufacturing Technology*, 33, 48–57. doi: 10.1007/s00170-006-0447-0.
29. Sasiadek, M. 2013. Computer aided mechanical assembly sequence planning. Paper presented at the 7<sup>th</sup> International Conference on Research and Practical Issues of Enterprise Information Systems – CONFENIS 2013. Prague, Czech Republic. 2013. - Linz: Trauner Verlag, 335–343.
30. Sasiadek, M. 2015(a). Assembly Sequence Planning with the Principles of Design for Assembly. Paper presented at the 20th International Conference on Engineering Design (ICED 15) Vol 4: Design to X, Milan, Italy, 31–40.
31. Sasiadek, M. 2015(b). Planning and analysis of mechanical assembly sequences in design engineering – part I: the method. *Technical Gazette* 22(2), 337–342.
32. Sasiadek, M. 2015(c). Planning and analysis of mechanical assembly sequences in design engineering – part II: application. *Technical Gazette* 22(3), 643–648.
33. Wang, H., R. Yiming, and X. Dong. 2014. Mechanical assembly planning using ant colony optimization. *Computer-Aided Design* 47, 59–71. doi: 10.1016/j.cad.2013.09.001. 12
34. Wen-Chin Chen, Hsu Yung-Yuan, Hsieh Ling-Feng, and Tai Pei-Hao. 2010. A systematic optimization approach for assembly sequence planning using Taguchi method, DOE, and BPNN. *Expert Systems with Applications: An International Journal*, v.37 n.1, 716–726. doi: 10.1016/j.eswa.2009.05.098.
35. Whitney, D.E. 2004. *Mechanical Assemblies: Their Design, Manufacture, and Role in Product Development*. Oxford University Press. doi: 10.1108/aa.2006.26.2.167.1.
36. Xiaoming, Z., Pingan, D. A model-based approach to assembly sequence planning. *Int J Adv Manuf Technol* 39, 983–994 (2008). <https://doi.org/10.1007/s00170-007-1272-9>