DE GRUYTER
OPEN

⊃mper

COMPUTER AIDED ASSEMBLY SEQUENCE GENERATION

Marcin Suszyński, Jan Żurek

Poznań University of Technology, Faculty of Mechanical Engineering and Management, Poland

Corresponding author:
marcin.suszynski
Poznań University of Technology
Faculty of Mechanical Engineering and Management
Piotrowo 3, 60-965 Poznań, Poland
phone: (+61) 665-22-51
e-mail: marcin.suszynski@put.poznan.pl

ABSTRACT
The purpose of the paper is to explore the problem of modeling technological assembly process, particularly generating assembly sequence for parts and machinery sets. A new computer program Msassembly is introduced. The program was invented by the authors on the basis of an algorithm for determining assembly sequence for parts and machinery sets. The algorithm is based on hypergraphs and directed graphs, as well as on assessment of transitions between assembly states. The principles of operation of Msassembly are presented on the example of modelling the assembly sequence of a ball joint. At the end of the paper, research findings are submitted.

KEYWORDS
assembly sequence generation, graph theory.

Introduction

The technological process of assembly is a final and the most important phase of the production process. It determines not only the final parameters of the product but also its time-consumption. Thus it is relevant to work out the optimal assembly technology to be used in certain conditions. It is worth pointing out though, that it is difficult to find the perfect solution as many criteria need to be taken in consideration, and once the technology is adopted, it is hard to change it.

The problem is widely discussed in literature, yet in a rather limited scope. In the paper [1] a review of current assembly design techniques is presented, and potential problems that may occur in the assembly process are discussed. The problems include identifying the key aspects regarding relations between individual parts of technological operations. The next paper [2] is a thorough analysis of designing process leading to production and assembly, and it also explains DFMA methods and selected tools. The third work [3] presents methodology of supporting decisions with parallel product designing (including selecting product assembly sequence). The authors

point out that product assemblability (the ability of a part to be joined into complex assembly units) is an underestimated feature not only in the body of literature, but also by the industrial sector. They suggest that the field of generating assembly sequence should be divided into three dependent parts: 1. defining structure (with respect to parts), 2. defining sequence limitations (in the light of their validity and evaluation), 3. generating sequence (i.e. adding parts, describing features of parts and connections) with decision supporting processes. In the next paper [4] a systematic, comprehensive approach to modelling and designing units was presented including important techniques and methods of generating assembly sequence with the use of logical mathematical formulation. The works [5–8] are comprehensive analyses regarding generating assembly sequence with the use of graph theory and various graph types, including records of matrix structures of assembly units and final products. They also comprise suggestions as to generating and evaluating assembly sequence with the use of selected algorithms.

The analysis of the literature on the subject shows that while modelling the technological assembly process, one should consider in particular the se-

quence of joining assembly units in a way that allows limiting their time consumption and complexity of assembly movements (e.g. length and shape of trajectory) and thus cutting costs.

This is why the most essential parameters of the assembly sequence model should be the following:

method of joining parts within an assembly nit or product (machine), relations between individual parts and assembly units, orientation of parts and assembly units in space and joining directions [1–9].

An integral part of each system supporting the generation of assembly sequence is the model that defines its structure. To describe the model, various graph structures for the products assembled, cluster methods and part grouping are used. However, graphs, assembly trees, and transition diagrams demand to be described in the form of a matrix that can be computer processed, and that will enable the user to apply the selected searching algorithms [2, 5–8].

Computer-aided modeling of assembly process is beneficial with regard to considerable time-consumption of its planning and programming. Manual planning of assembly process, even if the product consists of few parts, does not guarantee finding a good solution, and what is important, does not let the user react quickly to the ever-changing market demand. This in turn may be achieved by using the computer program proposed and described by the author of this paper.

Msassembly computer program

Msassembly computer program was developed in Visual Basic and it supports determination of assembly sequence using directed hypergraph and graph, state matrix and Dijkstra's algorithm. An important part of the proposed program is the method of determining the assembly sequence and its binding to the computing power of the computer, mentioned in e.g. the works [6–8].

The main menu of the program and available commands are shown in Fig. 1. The main part of the program is formed by six tabs: 1) list of parts, 2) construction of hypergraph, 3) graph, 4) evaluation of transitions, 5) state matrix, and 6) assembly sequence. The tabs enable the insertion and editing of data necessary to generate assembly sequence in accordance with the procedures [6–8]. The editing window of the tab "List of parts" allows to enter parts and / or assembly units involved in the assembly, including their design documentation (Fig. 1). The tab "Construction of hypergraph" allows to build a hypergraph of the construction structure of assembly units or the whole product and to generate a direct-

ed graph on its basis (in case of incorrect filling in individual vertices of the hypergraph, the program prevents transition into a directed graph). The upper triangle of the state matrix is filled with the values predefined in the tab "Evaluation of transitions". The lower triangle allows the user to save other characteristics of the transition in the form of a numeric code – e.g. the direction of joining parts. In the tab "Assembly sequence" all possible paths are generated (assembly sequence) taking into account the criterion of construction constraints and their evaluation score.
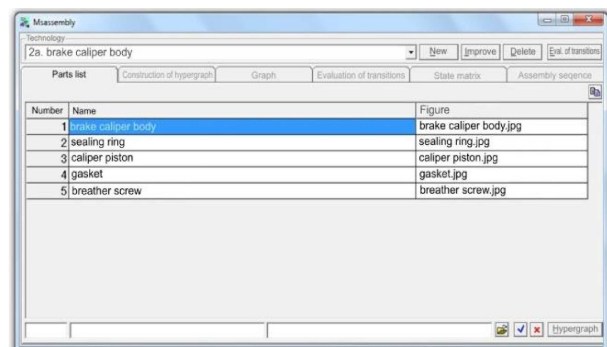

Fig. 1. The main menu of the Msassembly program.

The tab "Evaluation of transitions" allows the user to define and edit codes and the name of the indicators characterizing the process of assembling units (e.g. the indicators A, C, F, G, H, S, W; Fig. 2).
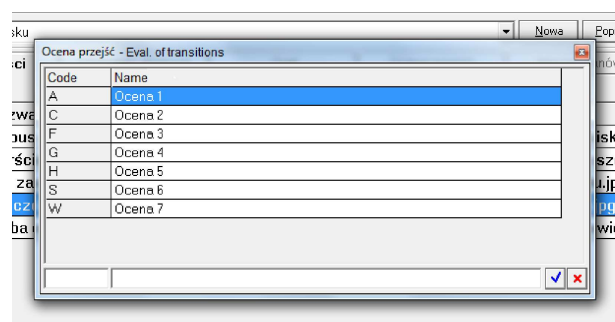

Fig. 2. The edit menu of the transition assessment (Msassembly program).

The evaluation criteria for the assembly sequence process, as expressed by the indicators, may be e.g.: assembly time, complexity of movements of assembly units, the degree of difficulty in reaching the next state of the process, the number of direction changes during assembly, size and weight of the assembled units, or even economics and correctness of the assembly process. In the described implementation into computer program it has been reduced to assess

the transition between two states of assembly characterized by possible connection combination of assembly units. The selected examples of indicators for assessing the sequence of transitions between assembly states (presented by the indicators f_p) are shown in Fig. 3 [1, 8].
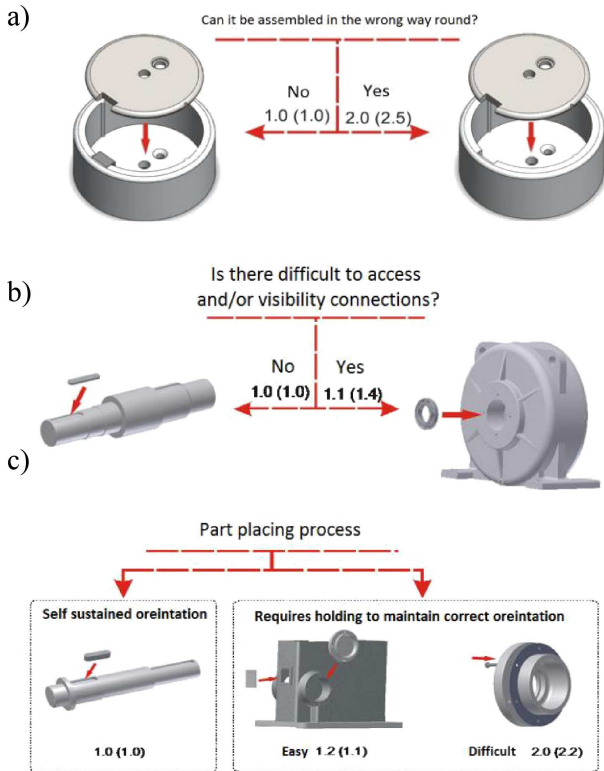
a)

b)

c)

Fig. 3. The selected characteristic indicators f_p: a) index A, b) index F, c) index C.

In the menu of the program the editing buttons are also placed: "new, correct, delete, save, copy" that enable to fill in and change names, numbers and drawings of parts and a button to save the entire procedure of determining the assembly sequence of the product or unit.

Generating assembly sequence using Msassembly Program

A simple example of determining the assembly sequence of a ball joint with Msassembly program is shown below (Fig. 4).

All parts of the ball joint were entered to the program, and then the base of the assembly process was selected – in this case the body of a ball joint (part 5). The input window together with the drawings is shown in Fig. 5.

a)

b)
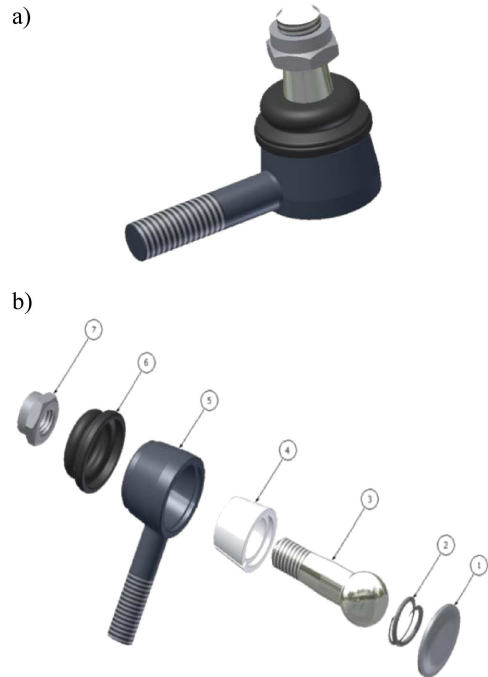
Fig. 4. Ball joint: assembled (a); exploded view (b): 1 – cap, 2 – spring, 3 – bolt, 4 – bowl, 5 – body, 6 – cover, 7 – nut.
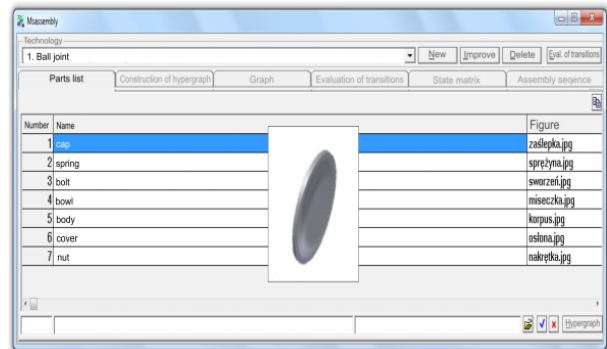

Fig. 5. Input window for parts (Msassembly program).

On the basis of the parts entered into the program, a directed hypergraph of the constructional limitation was built (Fig. 6), and then by minimizing the number of edges (the process takes place automatically) it was presented in the next tab of the program as a directed graph of constructional limitation (Fig. 7). Next, the transitions between different states of assembly were evaluated. The menu of the program for evaluation is shown in Fig. 8, and the result of filling in the state matrix is presented in Fig. 9. For the analyzed example (a ball joint) the values in the upper triangle of the matrix (Fig. 9) present the evaluation of transitions between different states in the assembly, and the values in the lower triangle of the matrix present possible directions of joining parts.
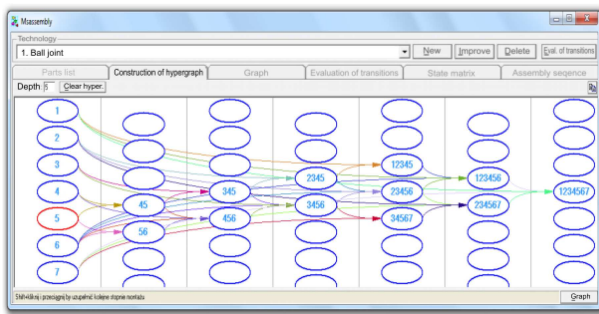
Fig. 6. Directed hypergraph of the construction structure for a ball joint (Msassembly program).
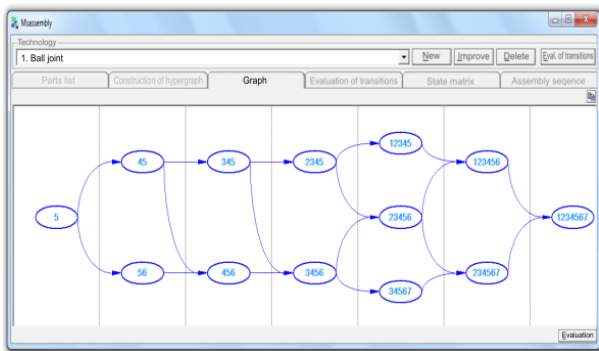

Fig. 7. Construction structure digraph of the ball joint (Msassembly program).

Fig. 8. The edit menu of the transition assessment between assembly states of ball joint.

Fig. 9. The state matrix of the ball joint (Msassembly program).

As a result of searching through the state matrix using the program with implemented Dijkstra algorithm, 12 possible sequence variants of assembling a ball joint were obtained together with their evaluation (Fig. 10). This made it possible to designate the most advantageous path at the value of 10.13 units:

$$5 \rightarrow 45 \rightarrow 345 \rightarrow 2345 \rightarrow 12345 \rightarrow 123456 \rightarrow 1234567.$$

Fig. 10. Feasible (due to construction criterion) variants of assembling a ball joint and their assessment (Msassembly program).

The considerations presented above confirm the usefulness of Msassembly program.

Conclusions

The proposal of modeling and determining the assembly sequence using hypergraph, directed graph, graph and state matrix presented by the authors in [5–8] was used to build and develop a computer program Msassembly supporting the work of technologists.

The proposed method of determining the assembly sequence of the product allows one not only to choose the best (rational) variant but it may also contribute to:

- reducing the time-consumption of developing and implementing technological assembly process,
- improving the quality of an assembled product,
- simplifying the process of joining the components and parts into a product,
- preparing quick analysis and rationalizing the assembly technological process,
- reducing the force necessary to match parts,
- facilitating the use of production means,
- reducing costs.

The final decision regarding selection of the most appropriate variant of the assembly sequence, and the structure of the technological process of assembly is taken obviously by a technologist-designer.

The developed Msassembly program should be treated as an auxiliary tool for designing and reorganizing of certain parts of the technological process

of assembly. It is expected to be expanded soon into more possibilities:

- automatic loading of assembly structures of assembly units (products) e.g. from CAD systems,
- developing the module for assessing transitions between different states in the assembly process,
- dividing the product into assembly units (new module),
- distribution of a product on the mounting unit (new module),
- cooperation with other systems, databases, technological measures of assembly and expert systems.

This will allow to generate the assembly process inclusive of the selected documentation.

References

[1] Booker J.D., Swift K.G., Brown N.J., *Designing for assembly quality: strategies, guidelines and techniques*, Journal of Engineering design, 16, 3, 279–295, June 2005.

[2] Boothroyd G., Dewhurst P., Knight W., *Product design for manufacture and assembly*, CRC Press., Boca Raton, 2002.

[3] Barnes C.J., Jared G.E.M., Swift K.G., *Decision support for sequence generation in an assembly oriented design environment*, Robotics and Computer-Integrated Manufacturing, 20, 289–300, 2004.

[4] Whitney D.E., *Mechanical Assemblies*, Oxford University Press, New York, 2004.

[5] Galantucci L., Percoco G., Spina R., *Assembly and Disassembly Planning by using Fuzzy Logic & Genetic Algorithms*, International Journal of Advanced Robotic Systems, 1, 2, 2004.

[6] Suszyński M., Ciszak O., Żurek J., *Modelling of the assembly structure with using directed graph and hypergrtaph and defining of tha assembly sequences*, Archive Production Engineering and Automation (Archiwum Technologii Maszyn i Automatyzacji), 4, 28, 2008.

[7] Suszyński M., Żurek J., Legutko S., *Modelling of assembly sequences using hypergraph and directed graph*, Technical Gazette, 21, 6, 111–120, 2014.

[8] Suszyński M., *Modelling of assemebly sequence using directed graph and hypergraph*, Dissertation, 2011.

[9] Yin Z., Ding H., Li H., Xiong Y., *A connector-based hierarchical approach to assembly sequence planning for mechanical assemblies*, Computer-Aided Design, 35, 1, 37–56, 2003.