An Intelligent Methodology for Assembly Tools Selection and Assembly Sequence Optimisation

Atul Mishra and Sankha Deb

Abstract In the present paper, an intelligent methodology has been used for assembly tool/gripper selection and determination of the optimal assembly sequence. Since it is well known that assembly sequence planning (ASP) is an NP-hard combinatorial optimization problem, in particular, with increase in number of components in the assembly, the computational complexity involved in searching for the optimal assembly sequence in such a large solution space also increases. Furthermore, assembly process planning, tool/gripper selection is also an important decision making task and becomes tedious and time consuming for an assembly with large number of components. Keeping the above in mind, in the present work, a knowledge based system has been developed for selection of assembly tools and grippers for performing the assembly, while a Genetic Algorithm (GA) based approach has been used to determine the feasible and optimal assembly sequences considering minimum number of tool changes and assembly direction changes.

Keywords Assembly sequence planning · Knowledge-based system · Tool selection · Optimisation · Genetic algorithm

1 Introduction

Assembly sequence planning is concerned with determination of sequence of assembly operations to assemble the components into the final product. It can not only improve the quality and efficiency of assembly, but can also reduce the product development cycle and cost (Yin et al. 2003). However, there are various constraints that can drive the choice of a feasible assembly sequence. Moreover, with increase in the number of components of which the product is made, the number of

A. Mishra · S. Deb (✉)
Mechanical Engineering Department, IIT Kharagpur, Kharagpur 721302, India
e-mail: sankha.deb@mech.iitkgp.ernet.in

A. Mishra
e-mail: atulmishra.chdi@gmail.com

D.K. Mandal and C.S. Syan (eds.), *CAD/CAM, Robotics and Factories of the Future*, Lecture Notes in Mechanical Engineering,
DOI 10.1007/978-81-322-2740-3_31

feasible assembly sequences possible also increases considerably and the task of manually determining the optimal assembly plan becomes laborious and time consuming. To automate this task, numerous approaches have been discussed in the open literature. They include use of mathematical algorithms (Gottipolu and Ghosh 2003), graph theoretic approaches (Laperriere and ElMaraghy 1994), as well as with the advent of Artificial Intelligence (AI) techniques various approaches like expert system, fuzzy logic and evolutionary soft computing based optimization techniques (Rashid et al. 2011) which include GA (Chen et al. 2001) and swarm intelligence based approaches (Tseng et al. 2011; Youhui et al. 2012), etc. Zha et al. (1999) developed an expert system for concurrent product design and assembly planning including generation of assembly sequences as well as tool and gripper selection. Barnes et al. (2004) also proposed a knowledge based approach for selection of base part and generation of assembly sequence. Dong et al. (2007) used a knowledge based approach for generation of assembly sequences with the use of reasoning based on geometric and non-geometric knowledge about connection types. Hsu et al. (2011) also used a knowledge-based system with an embedded neural network engine to predict a near-optimal feasible assembly sequence in accordance with geometric constraints and precedence relations of the parts to be assembled represented using an APD tree. Li and Tian (2013) generated all the feasible sequences by using a knowledge-based petri net system. GA has been used by some researchers for assembly sequence optimisation. Chen et al. (2001) proposed an adaptive GA (where genetic-operator probabilities vary according to certain rules) for efficiently finding global-optimal or near-optimal assembly sequences. Choi et al. (2009) proposed the optimisation of assembly sequences using GA in which precedence preservative crossover and swap mutation have been used. Tseng et al. (2010) presented an integrated assembly and disassembly planning optimisation using GA approach where assembly precedence graph has been used to transform into a precedence matrix. In the present paper, an intelligent assembly process planning methodology based on GA has been developed to automatically generate feasible and optimal assembly sequences subject to various assembly precedence constraints based on minimization of number of orientation changes of the assembly and tool changes. Furthermore, a knowledge-based system has been also developed for automatic tool/gripper selection, which is used as input to the above GA program to calculate number of tool changes.

2 Proposed Methodology for Assembly Sequence Planning

2.1 Knowledge-Based System for Selection of Tools and Grippers

A knowledge-based system has been used to select the tools and grippers for performing the assembly. A database has been developed to store all the information about assembly tools and grippers available in the shop floor. It includes

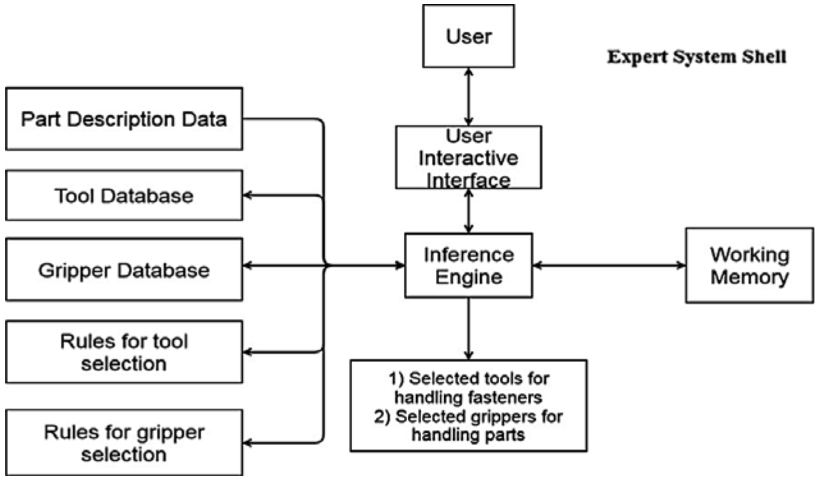

Fig. 1 Architecture of the developed knowledge-based system

```
(deftemplate MAIN::tool_available
(slot assembly_type(type SYMBOL)(allowed-symbols manual robotic))
(slot actuation_type (type SYMBOL)(allowed-symbols manual automatic))
(slot number (type INTEGER))(slot name (type SYMBOL))
(slot type (type SYMBOL) (allowed-symbols slotted_screw_driver philips_screw_driver
_press glue_gun))
(slot size_1 (type NUMBER))(slot size_2 (type NUMBER))
(slot min_torque(type NUMBER)(default 0))(slot max_torque(type NUMBER)(default 0))
(slot max_load(type INTEGER)(default 0))(slot cont_load(type INTEGER)(default 0)))
```

```
(deftemplate MAIN::tool_selected
(slot assembly_type(type SYMBOL)(allowed-symbols manual robotic))
(slot actuation_type (type SYMBOL)(allowed-symbols manual automatic))
(slot number (type INTEGER))(slot name (type SYMBOL))
(slot type (type SYMBOL)(allowed-symbols slotted_screw_driver philips_screw_driver
_press glue_gun))(slot for_fastener (type SYMBOL))
(slot size_1 (type NUMBER))(slot size_2 (type NUMBER))
(slot min_torque(type NUMBER)(default 0))(slot max_torque(type NUMBER)(default 0))
(slot max_load(type INTEGER)(default 0))(slot cont_load(type INTEGER)(default 0))
(slot fastening_time(type NUMBER)))
```

Fig. 2 Templates for entering the available and selected tool data in databases

type of assembly (manual or robotic), actuation type (manual or automated, the type of assembly tool, sizes etc. in case of the assembly tool, and the gripper, name, shape handling capability, maximum opening, gripping force, actuation type, adaptability, number of fingers and type of tip, etc. in case of the gripper. Assembly tools include tools for manual and robotic assembly, actuation types of manual and automatic, slotted, and Phillips screwdrivers, Allen keys, open end wrenches, adjustable wrenches, etc. Figure 1 shows the architecture of the developed knowledge-based system implemented in CLIPS expert system shell. Figures 2 and 3 respectively show the templates for entering the data on available and selected tools in the databases and the tool data entered in the databases following the above templates.

As per availability of tools, a set of knowledge based rules are used to select the tools along with their fastening times based on the selected assembly method i.e. manual or robotic assembly and fastener's characteristics viz. size, type and torque required. The fastening times are evaluated based on the part design features such as accessibility of assembly location, ease of operation of assembly tool, visibility of assembly location, ease of alignment and positioning during assembly, etc. (Design for assembly user guide 2009). Figure 4 shows an example of a knowledge-based rule for selection of tools. In the same way, a set of knowledge based rules are used to select the grippers for handling the components on the basis of part mass, its enveloping shape and size, and gripper's shape handling capability, maximum opening, gripping force, etc. The user provides detailed information about the assembly which consists of part types (i.e. functional part or fastener), mass,

```
;fact_list 2
(deffacts MAIN::tool_data
(tool_available (assembly_type manual)(actuation_type manual)(number 1)(name slotted_screw_driver1)(type slotted_screw_driver)(size_1 0.6)(size_2 0.4))
.....
(tool_available (assembly_type manual)(actuation_type manual)(number 16)(name allen_key1)(type allen_key)(size_1 1.5))
.....
(tool_available (assembly_type manual)(actuation_type manual)(number 41)(name open_end_wrench1)(type open_end_wrench)(size_1 6))
.....
(tool_available (assembly_type manual)(actuation_type manual)(number 66)(name adjustable_wrench1)(type adjustable_wrench)(size_1 13))
.....
(tool_available (assembly_type manual)(actuation_type automatic)(number 81)(name slotted_screw_driver1)(type slotted_screw_driver)(size_1 0.6)
(size_2 0.4)(min_torque 0.05)(max_torque 0.3))
.....)
```

Fig. 3 Tool data entered in the databases

```
(defrule MAIN::tool_selection_1
(or
(component (number ?a) (component_type fastener)(name ?c)(type screw) (specification ?s) (head_type $? slotted $?)(head_size_1 ?f)(head_size_2 ?h)(torque_required ?t)
(part_insertion_property added_and_secured)(part_tool_reachability easy)(obstructed_access no)(restricted_vision no)(aligning_positioning difficult)(torsional_resistance no))
(component (number ?a) (component_type fastener)(name ?c)(type screw) (specification ?s) (head_type $? slotted $?)(head_size_1 ?f)(head_size_2 ?h)(torque_required ?t)
(part_insertion_property added_and_secured)(part_tool_reachability easy)(obstructed_access no)(restricted_vision no)(aligning_positioning easy)(torsional_resistance yes))
...)
(or(tool_available (assembly_type ?z)(actuation_type ?at)(number ?m) (name ?d) (type slotted_screw_driver) (size_1 ?e)(size_2 ?g))
(and (tool_available (assembly_type ?z)(actuation_type ?at)(number ?m) (name ?d) (type slotted_screw_driver)(size_1 ?e)(size_2 ?g)(min_torque ?mint)(max_torque ?maxt))
(test (< ?t ?maxt))(test (> ?t ?mint))))(test (= ?f ?e))(test (= ?h ?g))
=>
(assert(tool_selected(assembly_type ?z)(actuation_type ?at)(number ?m) (name ?d) (type slotted_screw_driver)(for_fastener ?a)(size_1 ?e)(size_2 ?g)(fastening_time 8))))
```

Fig. 4 A sample rule for selecting the assembly tools

material, enveloping shape, dimensions, their contacts with other parts and contact directions in case of functional parts and head size, torque required, thread numbers, fit size, repeat counts, joined components and the other parameters like hole type, presence of chamfer, alpha and beta symmetry in case of fasteners. The developed knowledge based expert system is then used for selection of tools and grippers to be used in performing the assembly. Next the developed GA based optimization approach discussed in the following section is used to determine a feasible and optimal assembly sequence by minimizing the number of tool changes as well as direction changes.

2.2 Genetic Algorithm Based Approach for Assembly Sequence Optimisation

A GA based optimisation approach has been used to determine the feasible and optimal assembly sequences and implemented using MATLAB. The very first step in GA is to generate the initial population which is generated randomly, but since there can be both feasible and infeasible solutions, the infeasible solutions are avoided using a precedence matrix (number of rows and columns equal to number of parts) which consists of the values 0 and 1 to indicate whether the part possess any precedence or not with other parts. The feasible initial population is evaluated using the fitness function which is composed of number of direction orientation changes and tool/gripper changes and feasibility of the sequence. Figure 5 shows the flowchart of the implemented GA.

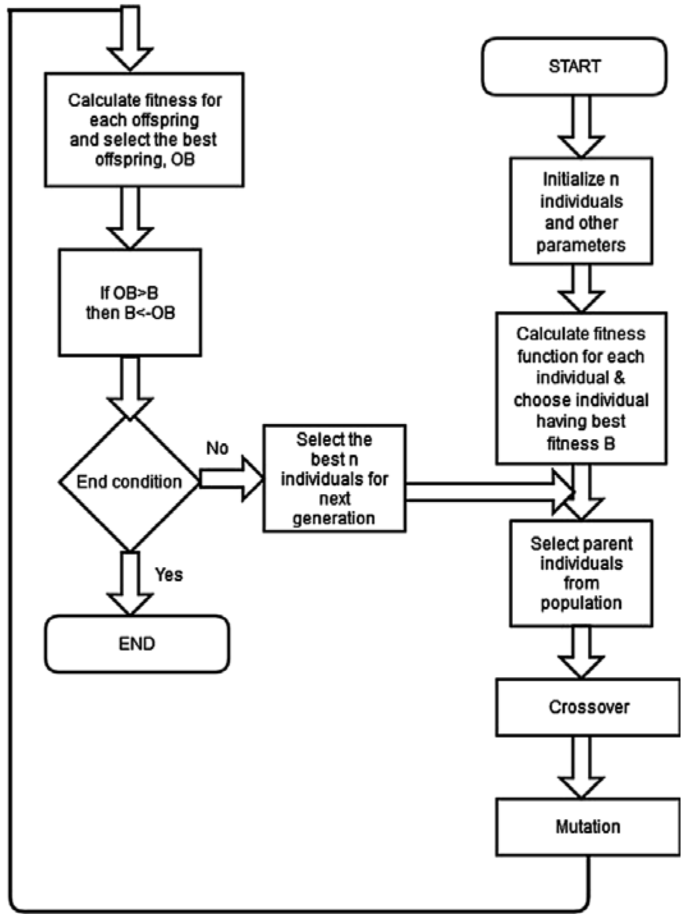

Fig. 5 Flowchart of the implemented GA

Selection

Roulette wheel selection has been applied for selecting the solutions which go into the mating pool. The principle behind it is that first using the fitness function, all the solutions are evaluated, and then according to their fitness values the ranges are created. Random numbers (the number of random numbers is equal to the population size) are generated and then the range, in which they fall in, is found out. Those solutions are inserted into the mating pool on which crossover operation is performed.

Crossover

The partially matched crossover (PMX) has been used here. In PMX crossover, it builds an offspring by choosing a subsequence of a solution from one parent and preserving the order and positions of as many genes as possible from the other parent. A subsequence is selected by choosing two random points, which serve as boundaries for the swapping operations. But in ASP problem, no parts in the assembly sequence can be repeated, which is taken care of by the above crossover operator.

Mutation

The swap mutation has been used here. In case of swap mutation, the genes are swapped randomly inside a solution.

Fitness Function

A fitness function has been built which comprises of mainly, number of orientation changes and tool/gripper changes. Solutions with lower fitness value possess high chance of survival as the objective of this work is to minimize the assembly time and cost, which are affected by the number of direction changes and number of tools/gripper changes. Hence the function is chosen so as to convert it into a maximization problem. The initial population (feasible population) is generated randomly. To avoid generating the infeasible population, after crossover and mutation the fitness function is also equipped with a feasibility index. During the solutions' evaluation, if they are found infeasible, a definite value (i.e. penalty) is deducted from the fitness function. Since, for the next generation, only the better solutions are picked up the algorithm gradually starts giving only feasible solutions. The Eq. (1) shows overall fitness function (FF) which has been used in the present work. Symbol w_x and w_y in the Eq. (1) are the weightages which are to be set by the user.

$$FF = 1/(w_x * number\ of\ direction\ changes + w_y * number\ of\ tools\ changes - penalty) \tag{1}$$

3 Illustrative Example

A motor drive assembly (Design for assembly user guide 2009) with a total 12 parts has been considered as an illustrative example for showing the working of knowledge based expert system and the developed GA based optimization approach. Figure 6 shows the assembly with all its parts. The parts of the assembly have been classified according to whether they are functional parts or fasteners. There are 7 functional parts and 5 different fasteners (like screws, standoffs, and grommet) in the final assembly. The different fasteners include, there are motor screws of specification M5 and head type slotted with head sizes 0.6 and 0.4 mm, set screw of specification M1.5 and head type hexagonal slot with head size 1.5 mm, end plate screw slotted screws, Phillips cover screws, bolts (standoffs) and

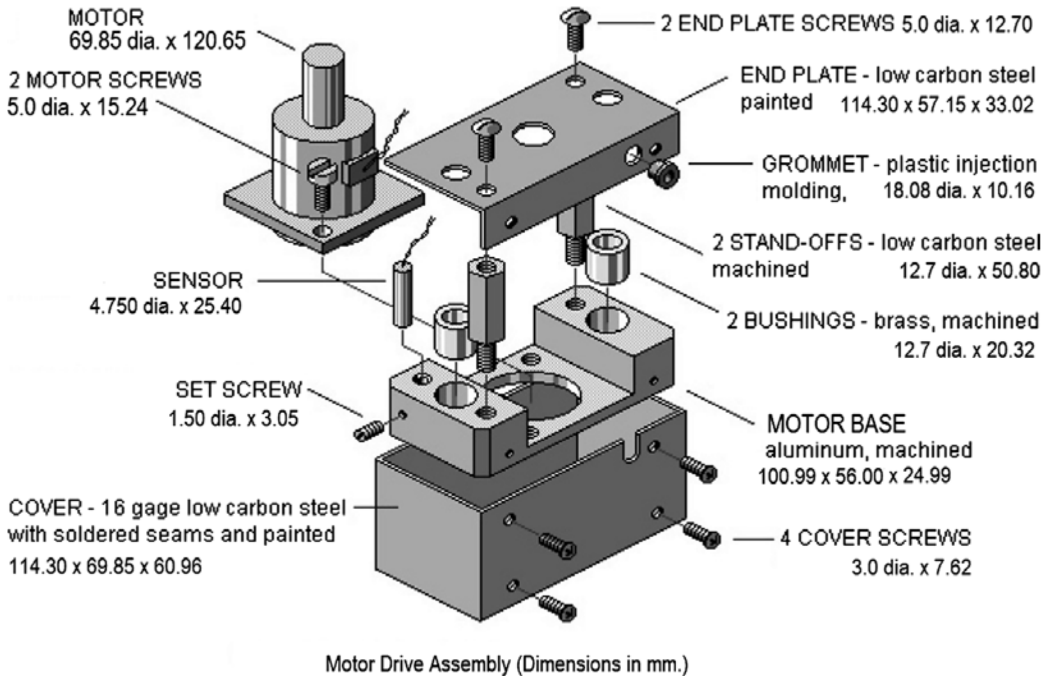

Fig. 6 A motor drive assembly (Design for assembly user guide 2009)

one grommet. All the data facts about the above components have been entered following the template format and loaded onto CLIPS environment. Next, the set of knowledge base rules for selection of assembly tools and grippers are loaded onto CLIPS and the program is executed. On execution of the program, the knowledge-based system automatically selects all the tools and grippers to be used in performing the assembly. For generating a feasible and optimal assembly sequence by using the GA based optimization approach, the user has to additionally input the directions in which the parts need to be assembled, the base component and the precedence matrix containing the information about precedence relationships among the parts shown in Table 1. These inputs are in addition to the tools and grippers information obtained from the output of the knowledge based system. Using the above input data, the developed GA program automatically generates a feasible and optimal assembly sequence.

4 Results and Discussions

The results of the developed knowledge based system and the GA program have been shown in the Tables 2 and 3 respectively. In Table 2, part numbers, their names, the assembly tool or gripper that has been selected automatically by the knowledge based system have been shown, along with the fastening time of

Table 1 Precedence matrix for motor drive assembly

	1	2	3	4	5	6	7	8	9	10	11	12
1	0	0	0	0	0	0	0	0	0	0	0	0
2	1	0	0	0	0	0	0	0	0	0	0	0
3	1	0	0	0	0	0	0	0	0	0	0	0
4	1	1	1	0	1	0	1	1	0	0	1	0
5	1	0	0	0	0	0	0	0	0	0	0	0
6	1	1	1	1	1	0	1	1	1	0	1	1
7	1	0	1	0	0	0	0	0	0	0	0	0
8	1	0	0	0	1	0	0	0	0	0	0	0
9	1	1	1	1	1	0	1	1	0	0	1	0
10	1	1	1	1	1	1	1	1	1	0	1	1
11	1	0	0	0	0	0	0	0	0	0	0	0
12	1	1	1	1	1	0	1	1	1	0	1	0
Assembly directions	−z	−z	−z	−z	−z	+z	−z	+x	−z	−y	−z	−z

Table 2 Summary of results of the knowledge based system showing selected grippers and assembly tools

Part no.	Part name	Tool/Gripper name	Fastening time (s)
1	Motor base	2-finger adaptive gripper, 2-finger parallel gripper, 3-finger adaptive gripper	–
2	Bushing_1	2-finger parallel gripper, 2-finger adaptive gripper, 3-finger adaptive gripper	–
3	Motor	2-finger adaptive gripper, 2-finger parallel gripper, 3-finger adaptive gripper	–
4	End plate	2-finger adaptive gripper, 2-finger parallel gripper, 3-finger adaptive gripper	–
5	Sensor	2-finger adaptive gripper, 2-finger parallel gripper, 3-finger adaptive gripper	–
6	Cover	2-finger adaptive gripper, 2-finger parallel gripper, 3-finger adaptive gripper	–
7	Motor screw_1	Slotted screw driver	12
8	Set screw	Allen key	5
9	End plate screw_1	Slotted screw driver	8
10	Cover screw_1	Philips screw driver	10.5
11	Stand off_1	Open end wrench adjustable wrench	–
12	grommet	Hammer	12

assembly fasteners. The system is capable of selecting more than one tool/gripper, if more than one tool/gripper is found suitable. Figure 7 shows the convergence graph of GA, showing the average and current fitness values. For the GA, in case there are multiple feasible assembly sequences with least number of direction changes and

Table 3 Summary of results of the GA program showing a feasible and optimal assembly sequence

GA crossover and mutation probabilities	0.90 and 0.05
Termination criteria used for GA	Number of predetermined iterations
Optimal assembly sequence	(1 2 3 7 11 5 8 4 9 12 6 10)
Number of orientation changes	4
Number of tool changes	9
Fitness function value	0.2222 (assuming $w_x = w_y = 0.5$)

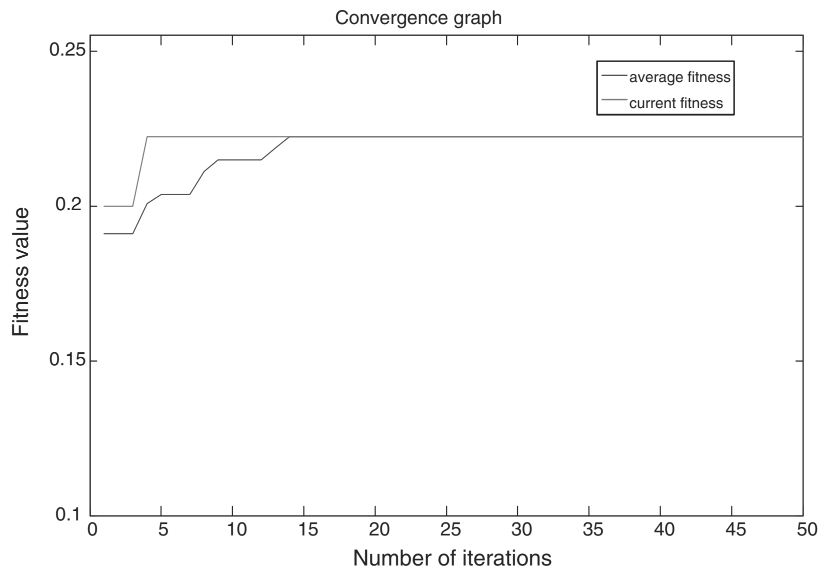

Fig. 7 Convergence Graph of GA

number of number of tool/gripper changes, the user can identify the optimal sequence by giving suitable weightages as per his needs. It has been found that GA takes on an average 2.5 s to determine the optimal solution.

5 Conclusions

In this paper, an intelligent methodology has been presented for assembly sequence planning and optimisation. A knowledge based expert system has been developed and implemented, which is capable of automatically selecting all the various tools/grippers needed for performing the assembly of a given product by considering the shop floor availability and capability of tools and grippers for handling functional parts and fasteners in the product. Further a GA based approach has been developed for assembly sequence optimisation and implemented using MATLAB. It takes as input the information on various tools/grippers from the output generated by the knowledge-based system and additionally it also takes as input from the user the assembly directions of each part, the information about precedence relationships among them and information about the base component. It then automatically generates as output the feasible and optimal assembly sequences based on minimising the number of orientation changes and tool changes. The proposed assembly process planning methodology can be integrated with a CAD based feature extraction module to minimize the user intervention and further reduce the time for generation of the assembly process plans.

References

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

Chen, S., & Liu, Y. J. (2001). An adaptive genetic assembly sequence planner. *International Journal of Computer Integrated Manufacturing, 14*(5), 489–500.

Choi, Y. K., Lee, D. M., & Cho, Y. B. (2009). An approach to multi-criteria assembly sequence planning using genetic algorithms. *International Journal of Advanced Manufacturing Technology, 42*, 180–188.

Design for Assembly user guide. (2009). Version 9.4, Boothroyd Dewhurst Inc., Rhode Island, USA, 5, 37.

Dong, T., Tong, R., Zhang, L., & Dong, J. (2007). A knowledge-based approach to assembly sequence planning. *International Journal of Advanced Manufacturing Technology, 32*, 1232–1244.

Gottipolu, R. B., & Ghosh, K. (2003). A simplified and efficient representation for evaluation and selection of assembly sequences. *Computers in Industry, 50*, 251–264.

Hsu, Y. Y., Tai, P.-H., Wang, M. W., & Chen, W. C. (2011). A knowledge-based engineering system for assembly sequence planning. *International Journal of Advanced Manufacturing Technology, 55*, 763–782.

Laperriere, L., & ElMaraghy, H. A. (1994). Assembly sequences planning for simultaneous engineering applications. *International Journal of Advanced Manufacturing Technology, 9*, 231–244.

Li, R., Tian, Y. (2013). Assembly sequence planning based on assembly knowledge database. In *Proceedings of the International Conference on Information Engineering and Applications* (Vol. 217, pp. 857–864).

Rashid, M. F. F., Hutabarat, W., & Tiwari, A. (2011). A review on assembly sequence planning and assembly line balancing optimisation using soft computing approaches. *International Journal of Advanced Manufacturing Technology, 59*(1–4), 335–349.

Tseng, Y. J., Kao, H. T., & Huang, F. Y. (2010). Integrated assembly and disassembly sequence planning using a GA approach. *International Journal of Production Research, 48*(20), 5991–6013.

Tseng, Y. J., Kao, H. T., & Huang, F. Y. (2011). A green assembly sequence planning model with a closed-loop assembly and disassembly sequence planning using a particle swarm optimization method. *International Journal of Advanced Manufacturing Technology, 57*, 1183–1197.

Yin, Z. P., Ding, H., & Li, H. X. (2003). A connector-based hierarchical approach to assembly sequence planning for mechanical assemblies. *Computer-Aided Design, 35*, 37–56.

Youhui, L., Xinhua, L., Qi, L. (2012). Assembly sequence planning based on ant colony algorithm. In *Future Communication, Computing, Control and Management* (Vol. 141, pp. 397–404). Heidelberg: Springer.

Zha, X. F., Lim, S. Y. E., & Fok, S. C. (1999). Development of expert system for concurrent product design and planning for assembly. *International Journal of Advanced Manufacturing Technology, 15*, 153–162.