Abstract

The tool selection is an important process in assembly planning. It is often required for industry to develop compact products in the competitive market. An efficient assembly depends on proper tools used in the assembly. It is a challenging task to embed tool selection in assembly planning because the tool selection in assembly is based on the tool feasibility that is highly dynamic, complex and variable. Therefore, current tool selection in assembly planning relies on planners’ experience. This paper presents an approach to tool selection-embedded optimal assembly planning. It aims to generate an optimal assembly plan with its tool alternatives.

1. Introduction

Assembly planning defines sequences of product assembly, by which initially separated parts are gathered to form a complete product. An optimal sequence results in less fixturing, less tooling, and more reliable operations during the assembly of a product. Moreover, the assembly plan plays a major role in aiding shop floor control, production system design, and scheduling activities because it can provide vital manufacturing information and impose constraints in the selection of production equipment and alternative routines [1]. These make assembly planning a crucial process through all stages of product development [2].

The tool selection is a key process to complete assembly planning. In particular, high fidelity assembly planning is accomplished by a proper tool selection with assembly planning. It is often required for industry developing more compact products for their competitiveness in market, which have reduced sizes and sufficient space for performing assembly or disassembly operations. Unlike other constraints in assembly planning, however, embedding a tool selection in assembly planning is a challenging task because the tooling process is based on the tool feasibility that is highly dynamic, complex and variable. Although much research has been developed toward a high-efficient and automatic assembly planning with the aid of computers, a success like tool selection-embedded assembly planning is yet to accomplish. Therefore, current tooling process relies on assembly planners with experience and expertise.

This paper presents an approach to tool selection-embedded optimal assembly planning. It aims to generate both an optimal assembly plan and its tool alternatives. To achieve this objective, the proposed approach is based on a fast tool feasibility analysis and a genetic algorithm (GA) based sequence planning. In particular, an assembly-tool database (DB) is designed to store a huge amount of tool information available in a dynamic manufacturing environment. It efficiently supports the assembly planning process and the tool selection. Moreover, a system is developed to simulate tool applications on a computer generated three-dimensional (3D) environment. The simulation can clearly verify that a tool set is appropriate for the optimized assembly plan.

2. Literature review

There have been many attempts to embed a tool selection in assembly or disassembly planning. In the research performed by Mello and Sanderson, attachments were included in a relational model of assemblies for fastening operations [3]. Although the attachments can be used to plan an assembly or disassembly sequence, the detailed tool applications required to remove these attachments were not modelled in their work. Miller and Hoffman [4] described a system that requires the access space for a fastener removal. However, the simple tests consisting of ray casting and box tests were used to roughly distinguish between feasible and infeasible tool applications.

The most detailed work was examined by Wilson [5]. The author developed a tool representation that includes the tool use volume, the minimum space that
should be free in an assembly to apply the tool. In his research, the author also mentioned the notion of fast tool reasoning that allows selection of a tool from a set of possible tools to execute an assembly operation. However, his approach requires the computation of the configuration space (C-space) representation of obstacles. The computation is too expensive to deal with a large number of possible tool sets for a fastener, which are retrieved from an assembly-tool DB.

Gupta et al [6] proposed a method to avoid the expensive computation of the C-space. With the articulated tool representation, the method combines collision detection methods and randomised via-point path planners. However, the method was also based on the same assumption used by Wilson [5], where tool place constraints and use volumes are not variable during a tool application. Because of limitations of their methods to deal with variations in a tool application, the feasibility of an assembly tool with 1-DOF was only discussed in their research.

Recently, Lazzerini and Marcelloni discussed the importance of a number of assembly tools applied and tool changes in optimal assembly sequence planning [7]. However, the authors did not incorporate this notion into their work. Researchers such as Kuo [8], Tseng and Li [9], and Yin et al [1] focused on fasteners in assembly or disassembly planning, but they also failed to notice assembly tools into their work, which may be feasible or infeasible for the fasteners in a given part configuration. In the research performed by Léon et al [10], a fast sequence generator for assembly or disassembly was proposed to support the early design process of a product development. Although the authors discussed about the significance of assembly tooling in generating an efficient assembly or disassembly sequence, no tool feasibility test was executed in their work. Moreover, Tseng et al [11] incorporated a number of tool changes into their objective function used to plan an optimal assembly sequence. In their research, however, the authors only used four types of general tools classified by the applied force magnitude.

The remainder of this paper is organized as follows: First, a method is proposed for fast tool feasibility analysis in this research. Next, a tool DB is designed to efficiently support the developed tool feasibility analysis and a tool-embedded optimal assembly planning discussed in the following section. Then, an example, a PC model, is experimented via the system developed in this research. Subsequently, a final section summarizes the paper, and draws conclusions and further work.

3. Fast assembly-tool feasibility analysis

Global accessibility cone with depth

The concept of an accessibility cone (GAC) is adopted for fast assembly-tool feasibility analysis instead of C-spaces or collision detections to make computation inexpensive. A GAC is used to represent feasible angles of attack toward an object. The concept of GACs has been extensively investigated, and leads to promising results for the accessibility analysis of tools in many applications such as multi-axis machining, laser scanning and coordinate measuring machine (CMM) [12-15].

In particular, based on the concept of discrete GACs [14, 15], a modified GAC is proposed in this research, named the global accessibility cone with depth (GAC\(^d\)). The GAC\(^d\) consists of 180×360 pixels to make total 64,800 directions on a sphere. The number of pixels is exactly matched with 180 colatitude angles (\(\varphi\)) and 360 longitude angles (\(\theta\)) in a spherical coordinate. The difference from normal GACs is that the GAC\(^d\) additionally includes depth information between obstacles and the center point. It is advantageous in considering both fastener visibility and tool feasibility that is based on complicated tool applications against obstacles. Figure 1 shows the process to form a GAC\(^d\).

![Figure 1. Construction of a GAC\(^d\).](image)

To form a GAC\(^d\), the initial position of a fastener becomes its center point, and the fastener removal direction is aligned to its y-axis. A triangle patch of a part with the shortest distance from the centre point of a GAC\(^d\) is first mapped into a spherical triangle [16] shown in Figure 1. The spherical triangle creates a pixel boundary defined by \(\theta_{\text{min}}, \theta_{\text{max}}, \varphi_{\text{min}}\) and \(\varphi_{\text{max}}\). Then, each direction defined by a pixel \((\theta, \varphi)\) within the pixel boundary is projected onto the triangle patch.
to find an intersection point. The intersection point \((X)\) on the triangle patch can be determined as follows.

\[
X = \frac{[(O - P_x) \cdot n]H - [(H - P_x) \cdot n]O}{[(O - P_x) \cdot n - (H - P_x) \cdot n]} \quad (1)
\]

Where, \(X = x_i + x_j + x_k \cdot \hat{n}\). \(H = \infty \hat{T}(\theta, \phi)\). \(\hat{T}(\theta, \phi)\) is the unit vector of a pixel \((\theta, \phi)\), and \(n\) is the normal vector of the triangle patch as shown in Figure 1. Subsequently, the intersection point \(X\) by Eq. (1) is tested whether it is inside the triangle patch. This test [17] is executed for three vertices on the triangle patch. If the intersection point \(X\) is inside the triangle patch, the direction at pixel \((\theta, \phi)\) within the pixel boundary is not accessible from the outside. In this case, the depth between the point \(X\) and centre point \((O)\) is calculated and stored at the pixel \((\theta, \phi)\) within the triangle patch. This process may rotate about a fastener.

**Feasibility analysis of an assembly tool**

The feasibility analysis of an assembly tool is a searching process with a constructed GACd. 4-articulated devices constituting an assembly tool are individually analysed for determining their feasibility against surrounding obstacles. For instance, following is the feasibility analysis for the effective handle.

An effective handle has most movements among the articulated devices constituting a tool. For instance, it may rotate about \(y\)-axis with variations in the access angle \(\alpha\) and the fastener removal displacement \(e_f\). In order to analyse its feasibility, a searching range based on these variations is defined along the \(\phi\) direction at a longitude angle \(\theta\). The defined searching range at the angle \(\theta\) is used for the interference check of the effective handle with a GACd. Thus, this check is executed until the required minimum tool-application angle \(\beta\) is found within the GACd.

**Tool representation and classification**

An assembly tool in this research is defined as a set of 4-articulated devices: an effective handle, a head, an extension and a base. For instance, the effective handle includes the maximum length of a tool handle, not involving the portion that is grasped and applied by human hands or robot arms. The base is the portion occupying a fastener head. Based on these descriptions, as shown in Figure 3, 15-geometric parameters are defined to represent an assembly tool, which are \(l_e\), \(w\), \(h\), \(r_o\), \(r_e\), \(e_v\), \(e_r\), \(e_b\), \(e_c\), \(e_s\), \(\alpha_{min}\), \(\alpha_{max}\) and \(\beta\).
\[ \phi_b = \cos^{-1}(Y/L) - \Delta \phi/2, \phi_i = \phi_b + \Delta \phi \] (4)

Where, \( X = l_i \sin \alpha + e_s, \ Y = l_i \cos \alpha + e_a + e_e + e_f, \)
\[ L = \sqrt{X^2 + Y^2 + e^2_s - e^2_e}, \ \Delta \phi = \tan^{-1}\left( \frac{h}{L \cos(\alpha - \cos^{-1}(Y/L))} \right), \]
\[ e' = \sqrt{e^2_s + e^2_r}, \ A = \frac{h}{\sqrt{2}} \cos\left( \frac{\alpha - \pi}{4} \right), \quad \text{and} \quad B = \frac{h}{\sqrt{2}} \sin\left( \frac{\alpha - \pi}{4} \right). \]

In addition, the required tool-application angle \( \beta \) is converted as \( \Delta \theta \) shown in Figure 4, which includes the angle generated by the width of an effective handle. Thus, the converted tool-application angle \( \Delta \theta \) is calculated as follows.
\[ \Delta \theta \equiv \beta + \tan^{-1}(w/X) \] (5)

In Eq. (5), particularly, \( \Delta \theta \) does not include \( \beta \) for a TAT, according to the tool classification defined in this research. With the searching range defined by four colatitude angles \( \phi_0, \phi_1, \sigma_0 \) and \( \sigma_1 \), and the converted tool-application angle \( \Delta \theta \), a searching process is executed with a GAC\textsuperscript{d}.

A criterion to determine the feasibility of an effective handle can be different depending on angle configurations that are formed by four angles \( \phi_0, \phi_1, \sigma_0 \) and \( \sigma_1 \). In this research, three different criteria are examined for three different angle configurations: (a) \( \sigma_0 < \phi_0, \sigma_1 < \phi_1 \), (b) \( \sigma_0 < \phi_0 \) and \( \sigma_1 > \phi_1 \), and (c) \( \sigma_0 > \phi_0, \sigma_1 < \phi_1 \). For instance, the following criterion derived in Figure 5 is used to examine the feasibility (A) for the case (a).

\[ A = \begin{cases} 1 & \text{if } (r^v(\phi) \leq R(\theta, \phi)) \lor (r^v(\phi) \geq R(\theta, \phi)) \\ 0 & \text{otherwise} \end{cases} \] (6)

Where, \( R(\theta, \phi) \) represents the defined depth to the direction at a pixel \((\theta, \phi)\) of a GAC\textsuperscript{d}. A distance \( r^v(\phi) \) can be simply determined in Figure 5, and expressed by following Eq. (7).

\[ r^v(\phi) = \frac{L \sin(\alpha - \phi^*)}{\sin(\alpha - \phi)}, \phi^* = \phi_0 \text{ or } \phi_1 \] (7)

In particular, a longitude angle \( \theta \) in Eq. (6) is not constant for an angle \( \phi \) although an effective handle is positioned at a longitude angle \( \theta_b \) shown in Figure 5. It results from the parameter \( e_n \) that has the effective handle misaligned with the angle \( \theta_b \). Therefore, Eq. (6) needs a conversion to the actual \( \theta \) according to a colatitude angle \( \phi_i \) and it is determined as follows.
\[ \theta = \cos^{-1}\left( \frac{e_n \sin(\alpha - \phi)}{L \sin(\alpha - \cos^{-1}(Y/L))} \right) + \theta_b - \tan^{-1}\left( \frac{X}{e_n} \right) \] (8)

![Figure 5. A detailed angle configuration of the effective handle.](image)

![Figure 6. Overall procedure of tool feasibility analysis.](image)
Similarly, geometric analyses are developed and applied to determine the feasibility of other articulated devices. Figure 6 shows a flow chart describing the overall procedure of testing the tool feasibility of an assembly tool. With variations in the tool access angle $\alpha$ and the fastener removal displacement $e_f$, a searching process is repeatedly executed on a constructed GAC. The process includes the four individual feasibility analyses for the articulated devices of an assembly tool and it continues until a fastener is completely disassembled without any collision with surroundings.

4. Design of tool database

Based on the relational DB representation [18], an assembly tool DB is modelled in this research. The assembly tool DB includes the hand tool and power tool DB; the entity-relationship diagram (ERD) of the hand tool DB is illustrated in Figure 7.

As shown in Figure 7, the structure of the hand tool DB consists of 37 tables and relationships, except fastener related tables. For each table, basic tool data including geometry information are obtained from a tool catalogue by Snap-on Company [19]. This geometry information of each basic tool is used to define tool parameters for a complete tool that is stored in the 'HandTool' table. Thus, a selected tool retrieved from the 'HandTool' table is directly used for a feasibility analysis based on the tool classification and tool parameters.

The 'HandTool' table in Figure 7 has the many-to-many relationship [18] with the 'Fastener' table so that various tool alternatives can be retrieved and analysed for a given fastener. In particular, drive hand tools make many tool alternatives based on types of handles and the use of extensions or universal joints. Moreover, these tool alternatives generated by the combination of tool components dramatically increase when an adapter is used, which allows a tool handle (i.e., 3/8" drive) to use other extensions or sockets with different drive sizes (i.e., 1/4"). With the support of object-oriented programming (OOP) technologies combined with the assembly tool DB, our goal in this research is efficiently achieved by quickly examining various assembly-tool alternatives for a given fastener.

5. Tool selection-embedded optimal assembly planning

Construction of fastener-based assembly structure

The tool embedded assembly planning starts with constructing a fastener-based assembly structure. Although this notion has been applied by many researchers [1, 9], we propose the fastener-based assembly structure based on tool feasibility. To form an assembly structure via the tool feasibility analysis, the developed system in this research retrieves assembly tools for fasteners in a product. Each fastener in a product stores all tools available in the assembly tool DB without considering any constraint. In particular, to efficiently store all information regarding a fastener, a fastener class is defined via Java language. Figure 8 shows the data structure of the fastener class with object-oriented tool DB (OOTDB).

In particular, the fastener class shown in Figure 8 includes parts connected with each fastener. The fastener-part connectivity matrix (FP) is automatically
formed via the method proposed by Chung and Peng [20]. At every assembly level, the developed system computes the tool feasibility for all tools retrieved for a fastener. After computing for all the fasteners in a product, the system suggests removable parts to a user, which are based on the assembly tool feasibility analysis and the matrix FP. Only fasteners connecting user-selected parts are stored in the fastener classes. Moreover, their OOTDBs are updated by feasible assembly tools that passed the tool feasibility test. Subsequently, this process is repeatedly carried out until all parts in a product are selected by users.

Genetic algorithm-based assembly planning

In this research, a GA-based approach is used to search an optimal assembly plan with a tool selection. Many GA-based approaches have been successful in the field of assembly planning [11, 21-24]. Based on results of the research, a GA is proved superior to traditional graph or matrix-based searching methods.

A two-tier GA shown in Figure 9 is introduced in this research because of the importance of initial populations in a GA searching a global optimum.

As shown in Figure 9, the initial population is manually set in the developed system. In addition, their OOTDBs are updated by feasible assembly tools that passed the tool feasibility test. Subsequently, this process is repeatedly carried out until all parts in a product are selected by users.

\[ f = 2m - (w_d n_d + w_b n_b + w_c n_c) \]  

Where, \( m \) is the total number of fasteners in a product. \( w_d \) is the weight of tool-change property, \( w_b \) is the weight of direction-change property, and \( w_c \) is the weight of distance property. These weights are manually set in the developed system. In addition, \( n_d, n_b \) and \( n_c \) have a value between 0 and 1. \( n_d \) is determined via tool information while \( n_b \) and \( n_c \) depend on fastener positions. The details are listed as follows:

- \( n_d \): 1 if a tool changes between two fasteners, \( F_i \) and \( F_{i+1} \) in an assembly plan; otherwise, 0.
- \( n_b \): a pre-defined discrete value between 0 and 1, which is based on the criterion of a direction-change by two removal vectors of fasteners, \( F_i \) and \( F_{i+1} \) in an assembly plan (Figure 10(a)).
- \( n_c \): a pre-defined discrete value between 0 and 1, which is based on the criterion of a distance from a fastener \( F_i \) to the next fastener \( F_{i+1} \) in an assembly plan (Figure 10(b)).

As shown in Figure 10(a), 4-discrete values for \( n_b \) are predefined by the angular criterion between two removal vectors of fasteners, \( F_i \) and \( F_{i+1} \) in an assembly plan. Similarly, \( n_c \) shown in figure 10(b) has 4-discrete values based on a given distance criterion (\( \delta \)) in this research.

Several GA operators are used in this research, which include a population selection, a crossover and two mutations. These operators deal with the genetic materials-genes in a chromosome. The features are described below [24-26].

Population selection uses the roulette wheel method to generate mating pool from the previous generation [26]. The roulette wheel is divided into slots weighted in proportion to the fitness values of the chromosomes. The wheel is spun and the selected member is the slot corresponding to the final position of the spinner. Clearly, the chromosomes with higher fitness values have higher chances of being selected to the mating pool.
Based on a given probability, the crossover operator selects two parent chromosomes. A crossover point is randomly chosen, which divides upper and lower nodes in a parent chromosome. After a child is identically copied from the parent chromosome, its upper nodes remain the same while its lower nodes are rearranged to follow the sequence in the other parent chromosome. By doing so, the genetic traits of the parent chromosomes are preserved in the child. As shown in Figure 11(a), for instance, child 1 consisting of five fasteners inherits the upper fasteners, $F_1$ and $F_2$, from parent 1; and the rest fasteners, $F_3$, $F_4$ and $F_5$, are rearranged by the fastener sequence in parent 2.

For two cases shown in Figure 12, the analyses respectively take 0.38 and 0.53 CPU sec on a Pentium 4 1.7 GHz PC with 512M RAM. The CPU time here only includes the time for the tool feasibility analysis without the time to construct a GAC, which takes a few seconds. In particular, a speeder with a universal joint tested in Figure 12(b) has the wide range ($\alpha=30°-60°$) of a tool access angle. This test searched the exact side hole of a part by repeatedly searching a GAC in the allowable range of the tool access angle.

A PC model is applied for tool selection-embedded assembly planning in this research. The PC shown in Figure 13 consists of 21 parts and 50 fasteners including 37 screws (#6 cross-head) and 13 bolts (#8 hex-head). In particular, the same types of screws and bolts are respectively used for the PC model in this research.

The first step of the tool selection-embedded assembly planning is to retrieve all the available tools from the tool DB developed in this research. For this example, 7 and 14 different assembly tools are respectively retrieved for any screw and any bolt in the PC model. 7 tools for the screw type include screw drivers with...
various lengths, ratchet screw drivers, and socket drivers combined with speeders and ratchets, whereas 14 tools for the bolt type include nut drivers, various wrenches, and sockets combined with speeders and ratchets. For each fastener in the PC model, the retrieved tools are stored at its OOTDBs.

To construct a fastener-based structure for assembly of the PC model, this research considers not only tool feasibility but also topological part assemblability. By constructing GAC’s for all remaining fasteners at each assembly level, the fast tool feasibility analyses with their OOTDBs are performed to determine a set of fasteners with tool feasibility. Then, removable parts based on the fasteners are suggested to a user for selection via the developed system shown in Figure 14. Subsequently, user-selected parts are converted into fasteners via the matrix $FP$ [20], which are added to a fastener-based structure. In this example, three assembly levels are defined for assembling the PC model. As shown in Figure 14, for instance, the second level includes three cables, RAM, cards, speaker, and switch. Moreover, the third level includes three covers, a front and two side covers. During the process, feasible tools for a fastener are also determined and newly stored in its OOTDBs.

In addition to the tool feasibility test, as shown in Figure 14, the fastener visibility test can be chosen by a user-selection. The test checks whether a fastener is visible from the outside. A GAC is used to simply find its empty pixel that doesn’t have depth information. It is not the necessary condition but the sufficient condition for a reliable assembly that may require tasks such as correct tool positioning or tool application monitoring. Figure 15 shows the notion of the fastener visibility and tool feasibility used in this research. As shown in Figure 15, a fastener in part configuration can be removed because an assembly tool is feasible although it is not visible by an observer.

The process to construct a fastener-based structure for assembly is a repeated and time-consuming process due to the fact that a tool feasibility can be different in accordance with a generated part configuration—a GAC. In this example, the total 680 CPU sec is elapsed for the complete computation while the mean time for each fastener with OOTDBs is 6 CPU sec.

Based on the defined fastener-based structure for the PC model, the two-tier GA-based approach proposed in this research generates a group of 9 optimal assembly plans with the highest fitness value. After only 10 generations, the fitness value is achieved because of the assembly-level based GA that runs in 50 generations. Figure 16 shows the result of the GA-based assembly planning for this example. Each assembly plan generated for this example has several tool alternatives of assembly-tool sets; an assembly-tool set consisting of two tools is shown via the tool navigation window of the developed system in Figure 16. Comparing to the number of tool alternatives that are generated by initially retrieved tools, about 13% of the tool alternatives, which have the least number of tool changes, are chosen with optimal assembly plans.

Figure 16 also shows a simulation of tool applications via a tool alternative for an assembly plan. The arrows in Figure 16 represent fastener assembly sequences based on the assembly plan. The simulation is performed on the system developed in this research. The system is implemented on a Web-technology including VRML and a Java database connectivity (JDBC) technology. It is based on the three-tier Internet-based structure that consists of a Web server, a remote DB server and clients. In particular, VRML-based models converted from normal CAD software are represented with a VRML node, ‘IndexedFaceSet’. The node consisting of many triangle patches is
advantageous to construct a GAC\(^d\) used in this research. Moreover, using external authoring interface (EAI) that provides Java applets with interfaces to control VRML nodes, the client system enables to carry out graphic-related operations in the definition, interaction and simulation of VRML models.

7. Conclusions and further work

This paper presents an approach to tool selection-embedded optimal assembly planning. It aims to generate both a high fidelity assembly plan embedding assembly tools, and its tool alternatives. A GAC\(^d\) is rapidly defined based on a discrete approximation of surroundings of a fastener, which includes both a set of accessible directions and depth information. The depth is defined as the distance between every non-accessible direction at the pixel of a GAC\(^d\) and its centre point.

Assembly tools are modelled as articulated devices. 15 parameters are extracted to define an assembly tool based on the tool representation in this research. In addition, two types of the tools, FATs and TATs based on the tool motions, are classified to efficiently support a fast feasibility analysis of an assembly tool. The tool parameters and classifications are stored in the assembly-tool DB designed in this research.

The feasibility of an assembly tool is analysed with a defined GAC\(^d\). The analysis is separately performed for each articulated device of a tool. The developed algorithms use a simple algebra in the defined searching range, which can quickly determine the geometric feasibility of an assembly tool. Moreover, the algorithms simply deal with complex variations in a fastener movement and a tool access angle.

A two-tier GA is introduced for the tool selection-embedded optimal assembly planning in this research. The two-tier GA consists of two GA-based methods: an assembly level-based GA for creating the initial population and a product-based GA for generating an optimal assembly plan. It helps to optimize fastener-based assembly sequences in a reduced number of generations. In particular, an object-oriented fastener class with OOTDBs is efficiently used to handle many retrieved tools from the assembly-tool DB during the

**Figure 16.** GA-based assembly planning and simulation of tool applications in a selected assembly plan.
assembly planning with the tool feasibility analysis. Based on the experiment performed in this paper, the results have demonstrated that the developed approach is efficient and practical for high-fidelity assembly planning in a dynamic manufacturing environment. A complete assembly planning can be performed with a large number of real assembly tools in a shop floor.

In addition to the dynamic assembly planning, the approach can efficiently support such disciplines as design, manufacturing, tooling and training through all stages of product development. In a rapid product development, it is efficient not only in dealing with various what-if scenarios regarding assembly but also in preventing re-tooling, special tooling, or design changing.

The further work is the micro selection of the best tool set among all tool alternatives generated in this research. The tool selection in the same types of tools with the same function will be based on cost, time or ergonomic features including a reliable operation.

Acknowledgement

This research is supported by Canadian NSERC Discovery Grants.

References


