Complex Assembly Variant Design in Agile Manufacturing.
Part II: Assembly Variant Design Methodology

Aihu Wang

Department of Management Science and Engineering, School of Business Administration
South China University of Technology
Guangzhou, Guangdong, P.R.China 510640

Bahattin Koc, and Rakesh Nagi*
Department of Industrial Engineering, 342 Bell Hall
University at Buffalo (SUNY)
Buffalo, NY 14260, USA

March 31, 2004

Abstract

In the first paper of this two-part series, the assembly variant design system architecture and complementary assembly methodology have been presented. The general complementary assembly models, hierarchical assembly model and relational assembly model, are established which were further specified as assembly variants model (AVM) and assembly mating graphs (AMG) respectively to cater for the needs for assembly variant design. This paper discusses the assembly variant design methodology which is based on these assembly models. The matching components are searched and retrieved from the AVM and then the constraint groups are identified by manipulating the AMGs. Then the assembly variant design process is formulated as a mixed integer (linear or nonlinear) programming problem which is solved using a standard solver or heuristic. This methodology provides a systematic approach to facilitate the variant design of complex assembly products in the agile manufacturing environment. Finally, a prototype system is developed and examples are presented.

Keywords: Variant Design, Agile Manufacturing, Assembly, Constraint Group.

---

*To whom correspondence should be addressed, nagi@buffalo.edu, Tel: 716-645-2357x2103, Fax: 716-645-3302

1 Introduction

The agile manufacturing environment is characterized by product mass customization, master proliferation and reduced order quantities [13]. Another irreversible trend that is not unique to agile manufacturing is that manufactured products are becoming more and more complex in both structure and technology content [33]. A methodology that helps create new complex assembly designs based on existing related product designs is in great demand. Variant design has been in common practice to relieve the designers from iterating similar design processes, hasten product development, reduce cost, and finally enable the manufacturing companies to develop individualized products based on existing mature designs. However a literature review reveals that most current variant design methods are focused on the redesign of single components. The research is motivated by this serious lack of assembly product oriented variant design methodologies.

Compared with the variant design of single components, the variant design process of complex assemblies is complicated by the relations between the constituent components. The assembly variant design methodology developed rests on the concept of constraint group (CG). Physical contacts between constituent components of an assembly create constraints on the relative positions of the components [21]. As a result, the dimensional change of a component might require the appropriate change of the other components involved in the constraint. An inter-related subset of components are named as a constraint group. The assembly variant design is driven by customer needs which are cascaded down to the requirements of constituent components. Based on the operation of assembly variants model, matching components are searched and retrieved. Then for each assembly variant, the constraint groups are identified by manipulating corresponding assembly mating graphs. The variant design process is finally formulated as a mixed integer nonlinear programing problem which is solved using a standard solver. The optimal solution provides insights about how to modify existing component designs to form a new assembly design with optimal redesign time and cost.

The paper is organized as follows. Relevant literature is reviewed in Section 2. Section 3 describes the assembly variant design methodology. The software implementation is presented in Section 4. Finally, the concluding remarks are presented in Section 5.

2 Literature Review on Variant Design

Design is an ill-structured domain where knowledge required for problem solving cannot be formalized into a robust model [32] and there are no well-defined, unique and optimum solutions [11]. This nature of design motivated a large body of research. Finger and Dixon [9] classify the research on the design process in mechanical engineering into two categories: descriptive models and prescriptive models. No matter what perspective researchers take, as function implementation design process [27] or cross-functional design decision making process [18] or design specification transformation process [29], the ultimate purpose is to

capture the insight of successful designs so that the experience and practice can be used in new designs. It is known that designers use their experience of design along with combinations and adaptations of previous designs or parts of designs in creating a new design [2]. This is the foundation of the variant design method that is aimed at increasing design standardization, reducing product proliferation, and reducing design lead time and cost by using existing mature designs.

Stutz and Kashyap [28] separate design into three categories distinguished by the degree of creativity involved in each case.

- Original design is the process of creating an entirely new artifact by means of a revolutionary new method or group of methods.

- Adaptive design is the process of applying a known solution principle to a problem which is largely different than any previously solved.

- Variant design refers to the process of applying known solution principles to a set of specifications which has only a slight variance from a known solution to a largely similar problem.

Variant design makes up approximately 30% of the activities in mechanical design [28]. Similarly, Prebil *et al.* [26] classifies design into novel design, adaptive design and variant design. They argue that approximately 70% of design work consists of adaptive and variant design, and only a minor part can be characterized as novel design according to analyses made in large companies.

Variant design methods are widely used in component design and process planning stages. One fundamental issue of variant design is to code the design scheme. The most successfully used mechanism to capture design and manufacturing attributes of mechanical components is Group Technology (GT) and hence is prevalent in variant mechanical design. GT recognizes and exploits similarities by performing like activities together, standardizing similar tasks and efficiently storing and retrieving information about recurring problems [14]. In industry, it is applied in design to reduce product proliferation, and in manufacturing to enhance productivity, quality and profitability [4]. Historically, classification and coding schemes were used to represent important design and manufacturing attributes through a string of alphanumeric characters, i.e., GT codes. Some of these include: Opitz [24], DCLASS [3], MICLASS [23] and KK3. A limiting factor in the extensive use of GT is the enormous effort required to derive the GT code for each part in a company's database. The coding process has largely been manual, allowing for inconsistencies and errors [17].

As to the methods to perform the variant design, Fowler [10] mentions two most widely used variant design methods: analogical reasoning applied to design and case-based reasoning applied to design. The latter is also called case-based design [32]. Analogical reasoning applied to design focuses on how to map existing problem solutions to new problems and emphasizes on reasoning and decision-making. The following issues

are of major concern.

- **Representation:** What information about the original design decisions is needed in order to retrieve them, and how should it be expressed?

- **Acquisition:** How can this information be captured?

- **Retrieval:** Given a problem, how can relevant previous designs be found?

- **Correspondence:** Which objects, goals, constraints, etc. in the new design corresponding to which ones in the old design?

- **Appropriateness:** When should a given plan or plan step be replayed?

- **Adaptation:** How can a previous plan be altered to fit a new problem?

- **Partial reuse:** Which parts of a plan can be replayed by themselves?

Case-based reasoning applied to design encompasses aspects of analogical problem-solving while more closely examining the issues of what aspects of existing problem solutions need to be represented in conjunction with how to select existing problem solutions relevant to the problem at hand.

The widespread use of variant design methods are impeded by the following factors.

- Though the computer-based tools currently employed in the design process have progressed dramatically in terms of geometric coverage, analysis capabilities, accuracy, visualization capability and speed, the fact that designers will need to redesign existing designs, or gain insights from the knowledge captured in an existing design, has not changed, nor is it likely that it ever will [10].

- Feature-based and parametric modeling capabilities of current CAD systems support variant design only in the sense that they may ease aspects of redesign but do not by themselves make for an environment enabling variant design [10].

- Analogical reasoning and case-based design illustrate the complex issues that need to be considered for autonomously retrieving and applying existing designs to solve new design problems [10].

- It is widely used in medicine where the design attributes are typical and relatively easy to capture and civil engineering where the structure of functional units are highly standardized. Unfortunately this method is not readily available in mechanical design owing to the geometrical complexity of mechanical components and the complicated relationships among components.

It is observed that most current variant design methods are aimed at single components. In this research, the variant design scope is expanded to include complex assemblies. It is challenging since the variant design process is complicated by the relationships between constituent components and hence a new variant design methodology needs to be developed for complex assemblies. The following sections cover literature related to assembly level variant design approaches, and how our approach is distinguished from them.

2.1 Parametric Scaling and Product Family Design

In a related sense, parametric scaling and product family design research has been conducted. Nayak *et al.* [22] present the Variation-Based Platform Design Method (VBPDM) for product family design to satisfy a range of performance requirements using the variation of the product design in the family. A decision support problem is used in their paper to solve the tradeoff between satisfying the variety requirement and minimizing the deviation of the input design variables. The authors used the developed method to design a family of universal electric motors to meet a range of torque requirements. DSouza *et al.* [8] used genetic algorithms to generate feasible designs to satisfy a variety of performance criteria based on varying levels of commonality. They found that increased commonality can adversely impact the performance of the products. To reduce the impact of the commonality on performance, Hernandez *et al.* [12] proposed to specify different levels of commonality for the various features and components of the product family. In Messac *et al.* [19], the authors used a product family penalty function to select common and scaling parameters for families of products from scalable product platforms. The authors used the non-platform variables as the scale factors to design variant models while common parameters were used to provide commonality. However, they do not consider the variant design of assembly products and specifically the components mating relationship during parametric scaling of each individual component. More specifically, our approach allows "mixing-and-matching" components from different assemblies to develop the variant assembly design. In the process some dimensions can be scaled up while others can be scaled down.

2.2 Parametric Design and Constraint-based Modeling

Parametric or variational computer-aided design (CAD) systems are based mostly on feature-based representation of solid objects. Constraint-based based modeling techniques are commonly used in parametric modeling when defining or editing the shape of features or their positions and orientations as well interrelations [5]. The constraints can be applied by the designer or can be detected automatically. In a parametric CAD system, the shape of a part or an assembly can be designed once and the variants can be derived by changing the dimension values, engineering parameters or the feature history of the part [7]. In this representation method, feature operations such as generated features, modifying features and datum features are used to represent the components. The feature-based model is then processed by a design compiler to define and

place the features by solving specified constraints. With respect the present research, this body of work can be critiqued as follows: (i) this approach is restricted to designs that are specified in parametric/constraint-based models and the vast volume of legacy designs would be unsupported by this method, (ii) indirect relationships (refer to Part I) are rarely specified in these models and the designer will have to address them manually, (iii) the designer does not have cost of time guidance on how to change dimensions/parameters, certainly not in a holistic manner (see Section 3.6), and (iv) when "mixing-and-matching" components from various constraint-based models, new constraints have to be reestablished, which could make variant design very tedious.

3   Assembly Variant Design Methodology

The assembly variant design methodology developed in this paper is based on the assembly models, assembly variants model and assembly mating graphs, described in the first paper [31]. First, the matching components are searched and retrieved from the assembly variants model and are organized as the so called projected assembly variants model. Then the constraint groups are identified by manipulating corresponding assembly mating graphs. Finally, the variant design of each constraint group is formulated as a mixed integer linear programming problem. This section discusses the details of the variant design methodology.
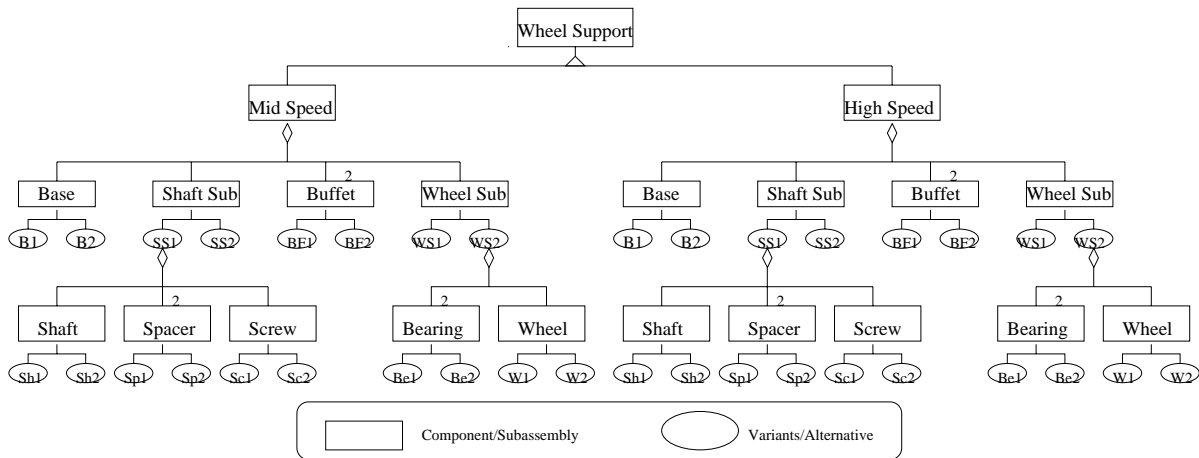

Figure 1: Similar Components Organized using Assembly Variants Model

3.1   Similar Component Search based on Assembly Variants Model

The assembly variant design is driven by customer needs which are transformed into product specifications and are further broken down to the specifications of each component. The component variants at the component level are then searched and retrieved from the assembly variants model. It is not the focus of this paper to develop a detailed procedure for component search and retrieval. Interested readers are referred

to Iyer and Nagi's work [16]. The retrieved components can be considered as a organized into a format similar to the Assembly Variants Model (AVM). A graphical representation of this information is shown in Fig. 1. The alternatives for each subassembly or component are also shown in the figure as oval nodes. Figure 1 clearly shows the combinatorial nature of the assembly variant design. The assembly product is composed of many subassemblies and components, in turn, each subassembly or component has a number of alternatives. The direct result is the large number of possible assembly configurations even for a moderate size assembly product. The similar components serve as the basis for the specific assembly variant design. Having determined the choice of component alternatives, the following section, captures the relationships between these components using the concept of constraint groups.

3.2 Concept of a Constraint Group (CG)

Physical contacts between constituent components of an assembly create constraints on the relative positions of the components. As a result, the dimensional change of a component might require the appropriate change of the other components involved in the constraint. An inter-related subset of components are named as a Constraint Group (CG). The identification of the constraint groups in an assembly is very useful. As stated by Mullins and Anderson [21], the identified geometric constraints in mechanical assemblies can be used to:

- Provide dimension sensitivity feedback to the designer.

- Identify relationships that the designer may not recognize or fully comprehend.

- Allow for reduction of design time by making it easier to modify the existing design.

- Be useful for both top-down and bottom-up design modes.

The assembly variant design methodology developed is based on the concept of constraint group. For each representative type of assembly variant, the CGs can be identified by manipulating the corresponding assembly mating graph. Then, variant design is performed to make the components in the CGs fit with each other by formulating it as a mixed integer linear programming problem.

3.3 Constraint Group Identification

In this research, the mating conditions are modeled at single form (geometric) feature level. The "fit" is defined as a mating relationship where, for instance, two cylindrical features fit together with the center lines of the two features coincide. Under this circumstance, the CGs are composed either by "fit" or "against" mating relations but not both. As a result, the CGs can be classified into to categories: Fit Constraint Group (FCG) which is a CG that is composed of "fit" mating relations only, and Against Constraint Group

(ACG) which is a CG that is composed of "against" mating relations only. Accordingly, each Assembly Mating Graph (AMG) which corresponds to a specific representative assembly variant can be decomposed into two subsets/subgraphs: Fit Assembly Mating Graph as shown in Fig. 2 and Against Assembly Mating Graph as shown in Fig. 3. The fit AMG as shown in Fig. 2 captures the "fit" mating relations for the medium speed wheel supportive assembly, while the against AMG as shown in Fig. 3 captures the "against" mating relations. The identification of the FCGs and ACGs can be based on the fit AMG and against AMG respectively.
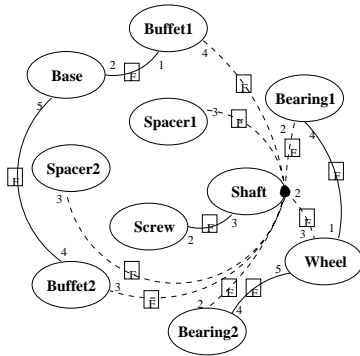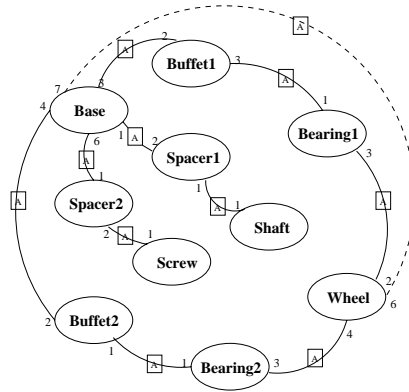

Figure 2: Fit Assembly Mating Graph

Figure 3: Against Assembly Mating Graph

FCG is a group of components constrained by "fit" mating conditions. From a geometric constraint perspective, each fitting pair forms a CG itself. However, the relationships between fitting pairs intricate the variant design. Basically, the FCGs can be classified into two categories: isolated FCG and combined FCG. An isolated FCG is composed of only two components that are fitting with each other by a pair of mating features. Except the mating relationships between themselves, the two mating features have no mating relationships with other features. Fig. 4 shows the isolated FCGs of the medium speed wheel supportive assembly product. In contrast, a combined FCG is a subset of components that are fitting with each other in such as way that a single feature of a component fits with all the remaining components in that subset (i.e., a one-to-many relationship). For example, as shown in Fig. 5, the shaft, spacer1, buffet1, bearing1, wheel, bearing2, buffet2 and spacer2 form a combined FCG because the shaft fits with all the remaining components in the subset through a single feature $mf2$ of the shaft. The constituent components of a combined FCG form a star in graph/network theory. The distinction of isolated and combined FCG is important from a variant design perspective since the isolated FCG caused local changes while combined FCG cause dimensional changes to many components, which might be more costly and time consuming. These observation provides the insight as how to identify the combined FCGs from the corresponding fit AMG. It is also noted that not all these classifications need to be present simultaneously in the assembly at hand.

ACG is a group of components constrained by "against" mating conditions. The ACGs have the following characteristics.
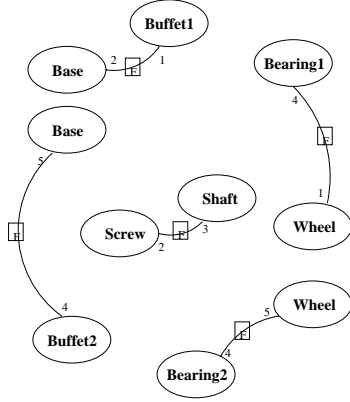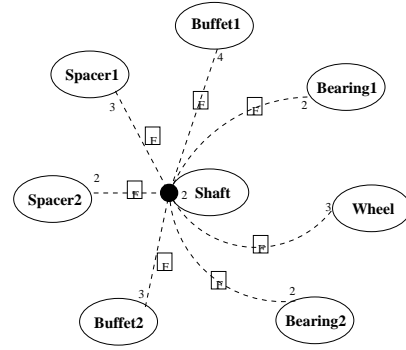
8

Figure 4: Isolated Fit Constraint Groups

Figure 5: Combined Fit Constraint Groups

- The number of components involved might be large for complex mechanical assemblies.

- The "against" mating relationships between these components create dimensional constraints that can be used to guide the variant design of these components.

The ACGs are useful to the variant design and can be classified into two categories: cyclic ACG and acyclic ACG. For each cyclic ACG, the constituent components form a loop connected by the "against" mating pairs. The physical loop corresponds to a cycle in the against AMG. For instance, the base, buffet1, bearing1, wheel, bearing2 and buffet2 form a loop which is a cyclic ACG as shown in Fig. 6 for the medium speed wheel supportive assembly. The cyclic ACGs can be identified automatically from the corresponding against AMG using graph theoretic tools such as the Floyd-Warshall Algorithm [1] which obtains a matrix of shortest path distances within $O(n^3)$ given a matrix of distances $d[i,j]$. An acyclic ACG is a group of components which are connected with each other through "against" mating pairs without forming a loop. The dimensional relationships of the group of components are constrained by extra dimensional specifications. For example, Fig. 7 shows an acyclic ACG of the medium speed wheel supportive assembly. The extra dimensional specification of the acyclic ACG is the maximum width of the shaft and screw coordination. For complex assembly products, there may be many acyclic ACGs. Once again, appropriate graph theoretic tools can be applied to identify these [1]. The designer might be asked to help identify the meaningful acyclic ACGs if necessary.

The CG identification procedure can be outlined as follows:

- **Input:** The AMG.

- **Step 1:** Decompose the AMG into fit AMG and against AMG.

- **Step 2:** For the fit AMG: (a) Identify combined FCGs and save the results. (b) Identify isolated FCGs and save the results.
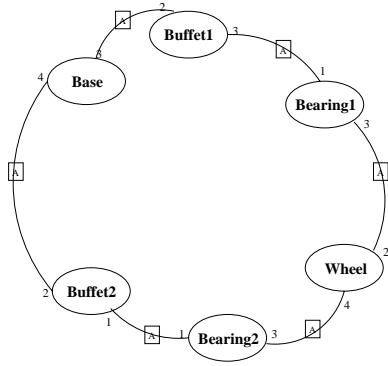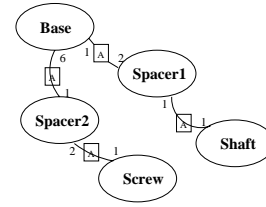
Figure 6: Cyclic Against Constraint Group

Figure 7: Acyclic Against Constraint Group

- **Step 3:** For the against AMG: (a) Identify cyclic ACGs (using Floyd-Warshall algorithm) and save the results. (b) Identify acyclic ACGs interactively and save the results.

- **Output:** The CGs identified.

To this point, the matching components are retrieved and stored in a format similar to the AVM and the constraint groups are also identified. However, the relationships between the key component design characteristics and the mating features needs to be setup and the redesign cost and time are to be defined before the details of the mixed integer programming problem can be formulated.

3.4 Relationships between the Key Component Design Characteristics and the Mating Features

To facilitate the cost and time related evaluation, the mating relationships between assembly constituent components are modeled at the form (geometric) feature level. Though a component is composed of features, people tend to interpret it using dimensions, tolerance and other attributes, together with the type of features. Therefore, we need to setup the relationships between the mating features and the dimensions of each component.

The Key Component Design Characteristics (KCDC) of a component are defined as parameters which specify the fundamental component features such as geometry, materials, and dimensions. Researchers use different terms to represent the same concept. For example, Minis *et al.* [20] use the term "critical design attributes" and STEP AP224 Protocol uses "form features".

AP224 is the STEP application protocol [6] which specifies the requirements for the representation and exchange of information needed to define product data necessary for manufacturing single piece mechanical parts. The product data is based on existing part designs that have their shapes represented by form features.

AP224 covers the following:

10

- Product data that defines a single piece machined part to be manufactured.

- Product data that covers parts manufactured by milling or turning.

- Product data that is necessary to track down the customer order in the shop floor.

- Product data necessary to identify the status of a part in the manufacturing process.

- Product data necessary to track raw stock certification.

- Product data necessary for tracking of a part design deficiency.

- Form features that are necessary for defining shapes necessary for manufacturing.

The KCDCs may represent the shape and material of the component. However, this research focuses on the KCDCs which specify the dimensions of the component and the KCDC schema used is a subset of the STEP AP224 protocol. These KCDCs are also concerned in the direct and indirect relationships of the AMG, which is not automated at this point (see Part I).

3.4.1 KCDC Naming Schema and Examples

To remain consistent with the naming taxonomy of high level computer languages such as C++ and Java, the naming schema of the KCDCs is designed as follows.

The name of each KCDC is composed of three parts. The first part is the name of the component with the first letter capitalized. The second part is the dimension type. In this research, the dimension types are classified into the following categories: Length, Height, Width, InnerDiameter and OuterDiameter. The categories can be expanded if necessary. The third part is the sequence number of the same kind of dimensional type of the same component. For example, the first length KCDC of component Base is BaseLength1 as shown in Table. 1.

Component Name	Dimensional Type	Number
Base	Length	1

Table 1: KCDC Naming Schema

For the purpose of demonstration and preparation for the variant design, the KCDCs for the components of the medium speed and load wheel supportive assembly are defined and shown from Fig. 8 through Fig. 14.

3.4.2 Relationships between the KCDCs

The KCDCs of each component are identified and are associated with the mating features. For the medium speed and load wheel supportive assembly, the mating features are shown in Fig. 6 of the first paper of this
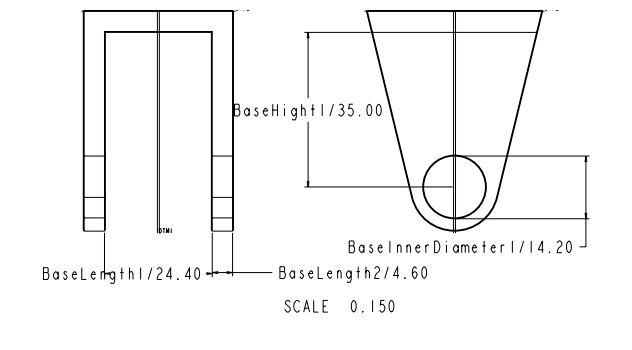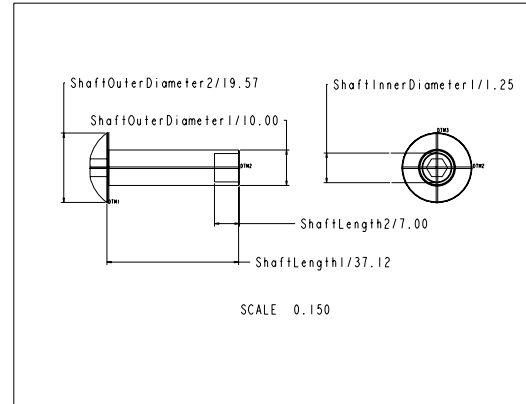
11

Figure 8: KCDCs for the Base
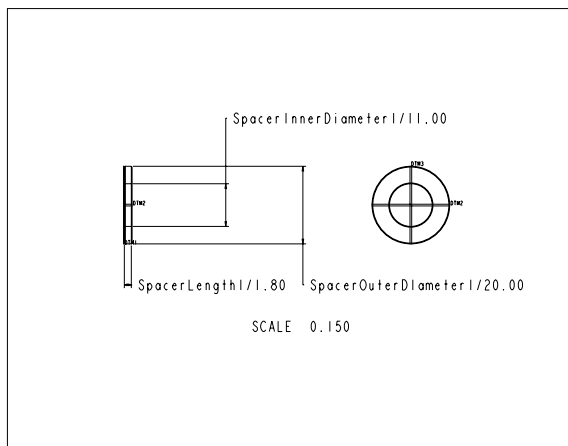

Figure 9: KCDCs for the Shaft
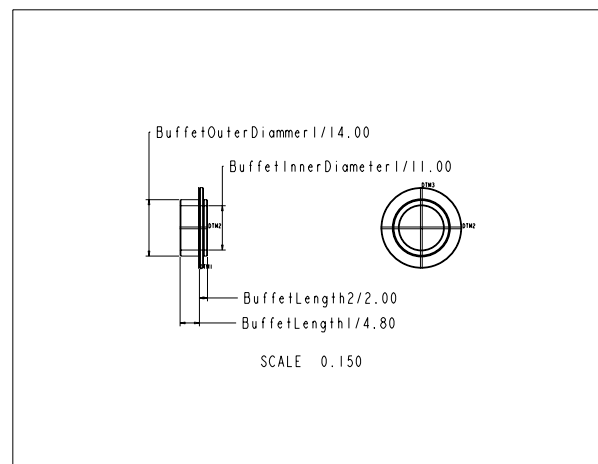

Figure 10: KCDCs for the Spacer

Figure 11: KCDCs for the Buffet

series [31] and the KCDCs for each constituent component are shown from Fig. 8 through Fig. 14. The relationships between the KCDCs of mating components can be classified into two categories: relationships captured by the constraint groups encoded in the assembly mating graph, and relationships not captured by the assembly mating graph.

KCDC relationships captured by the constraint groups belong to the following categories:

- **One Dimensional:** All the fit constraint groups and those against constraint groups in which all the mating surfaces are parallel to each other and are perpendicular to an axis of the coordinate system are termed as one dimensional. In this case, the relationships between KCDCs are straight forward. Figure 15 (a) demonstrates the one dimensional relationships between the KCDCs of an against constraint group, where $d1 = d2 + d3$. In the medium speed and load assembly supportive assembly, the against constraint group composed of the base, buffet1, bearing1, wheel, bearing2 and buffet2 is a one dimensional constraint group. The relationships between the involving KCDCs can be repre-
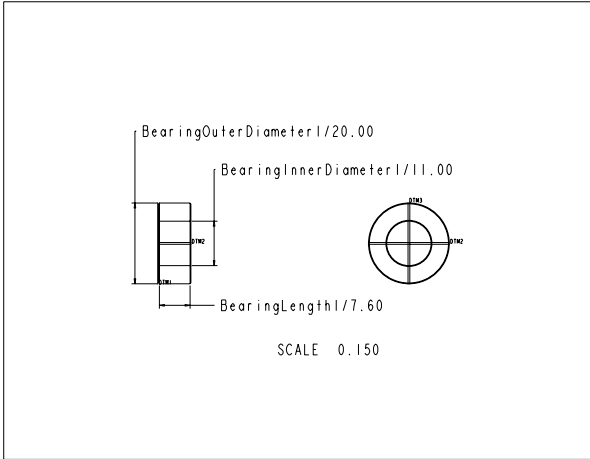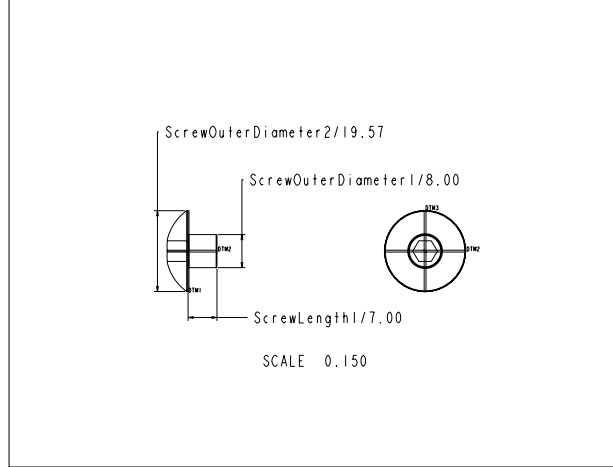
12

Figure 12: KCDCs for the Bearing

Figure 13: KCDCs for the Screw

sented as BaseLength1 = Buffet1Length2 + Bearing1Length1 + WheelLength2 + Bearing2Length1 + Buffet2Length2. Equations (13) or (17) of Section 3.6 can be used to represent these relationships.

- **Two or Three Dimensional:** Many against constraint groups (ACGs) can be two or even three dimensional. In this case, the two or three dimensional ACG has to be decomposed into two or three one dimensional ACGs and the same method developed above can be applied. The basic approach to decompose two and three dimensional ACGs into the corresponding one dimensional ACGs is straightforward and can be briefly outlined as follows.

Corresponding to each component, there is a local coordinate system. Therefore, for each component, the dimension involved in the ACG can be decomposed into three subsets along the three coordinate axes in its own local coordinate system. When the components are put together to form the assembly, their local coordinate systems (location and orientation) are placed the global coordinate system at the assembly level. Ultimately, a two or three dimensional ACG of the assembly can be decomposed into (respectively, two or three) one dimensional ACGs in the global coordinate system. Fig. 15 (b) shows a two-dimensional ACG and also demonstrates how it is composed into two constraint groups along the global X and Y axes. Now, the two obtained one dimensional constraints can be represented as: $\Delta X_{d1} = \Delta X_{d2} + \Delta X_{d3}$ and $\Delta Y_{d3} = \Delta Y_{d1} + \Delta Y_{d2}$.

In addition to the KCDC relationships captured by the assembly mating graph in the form of constraint groups, there are KCDC relationships which are established due to the relative positions of a pair of mating features of the components in the global coordinate system. Fig. 15 (c) shows two components that are suppose to mate. The relationships between $d11$ and $d21$, $d12$ and $d22$ are captured by the two fit mating pairs. However, the relationship between $d13$ and $d23$ is not captured yet. This kind of KCDC relationships also needs to be collected. In the variant design methodology, this category of KCDC relationships is

WheelInnerDiameter2/20.00
WheelHight1/34.00
WheelInnerDiameter1/11.00
DTM3
DTM2
DTM2
DTM1
WheelLength2/8.00
WheelLength1/20.00
SCALE 0.150
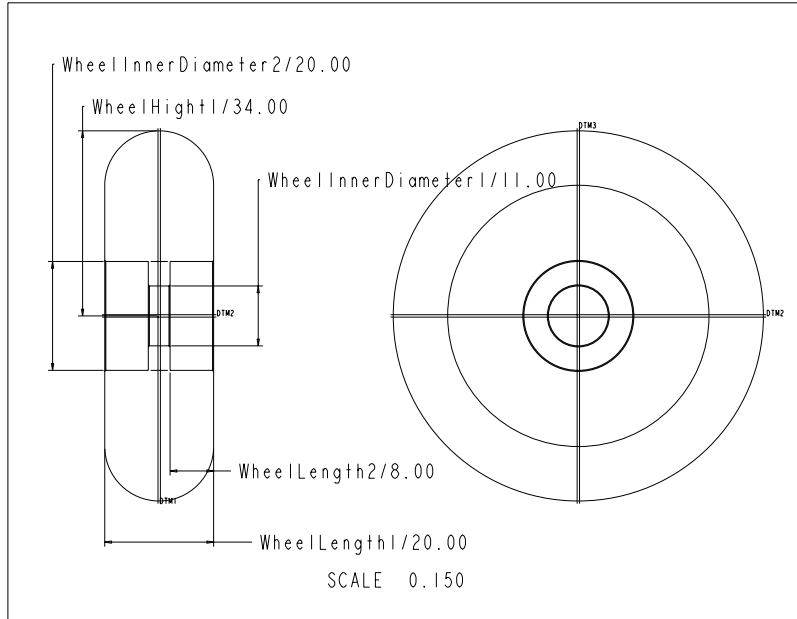
Figure 14: KCDCs for the Wheel

(a) One Dimensional

(b) Two Dimensional

(c) Extra Relationship
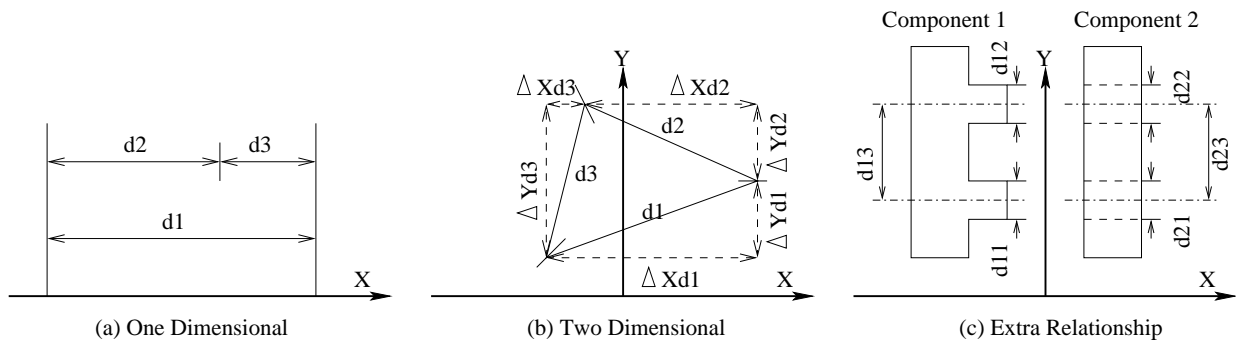
Figure 15: Relationships between KCDCs

identified and represented as extra constraints. The identification process is amenable for automation, but is designer identified in this paper for sake of simplicity.

3.5   Redesign Cost and Time

Assembly variant design is based on the modification of existing mature component designs and variant design schemes are evaluated by considering the redesign cost and time incurred by the modification. Therefore, the cost and time related information for each component in the AVM also needs to be collected.

For each similar component alternative, the original design and manufacturing cost and time and other cost and time related information (such as material cost, assembly time and cost, etc.) are taken as nominal

values and are stored in a database. To evaluate the variant design schemes, there is a need to estimate the design time and cost changes incurred by the modification of existing component designs. The following factors are necessary to be taken into consideration in order to leverage the effects of design changes in assembly variant design.

1. Cost and time for changing existing component design schemes in order to satisfy new customer needs.

2. Cost and time for changing the process plan of the components.

3. Actual manufacturing cost and time changes between the new variant design and the original design incurred due to the modification.

4. Cost and time for design and process plan verification.

5. Miscellaneous cost and time.

Traditionally, cost and time estimation methods are usually called cost estimation and they can be broadly divided into two categories according to their applicable conditions and targets.

The first category is employed when the product has been fully detailed and production is fully planned. It is a body of theory and practice that provides a measure of the economic want of the design [25] and is generally used by commercial offices to foresee the appropriate price of the product. Methodologies of this category are mature relatively and similar courses such as Engineering Cost Estimating, Cost Analysis, Cost Engineering, Cost Estimating, Engineering Economy, Industrial Analysis, and Manufacturing Estimating are offered in many universities. This category is usually referred to as traditional cost estimating.

The second category is used in early product design stage when the product is not fully defined and is targeted at evaluating design features by estimating the manufacturing costs expected to be incurred by them. Literature review shows that cost estimation methods during design stage are basically feature-based. Design features are evaluated by their direct effects on downstream activities and the effects are usually calculated as costs to be incurred. Though quantitative indices can be obtained, the accuracy of the estimation is hard to predict until detailed design is finished, since the product is never completely defined until then.

The brief literature review inspires this research to evaluate the cost and time changes at the form (geometric) feature level. Non-feature related factors such as factor 4 mentioned above, can be treated as fixed cost and time factors and hence one can employ traditional cost estimation methods (standard time and cost sheets) to estimate them. For the actual manufacturing cost and time changes, feature based methods can be established to deal with them. This is one of the reasons that this research models assembly mating relationships at the form (geometric) feature level. In this research, the cost and time changes are associated with form
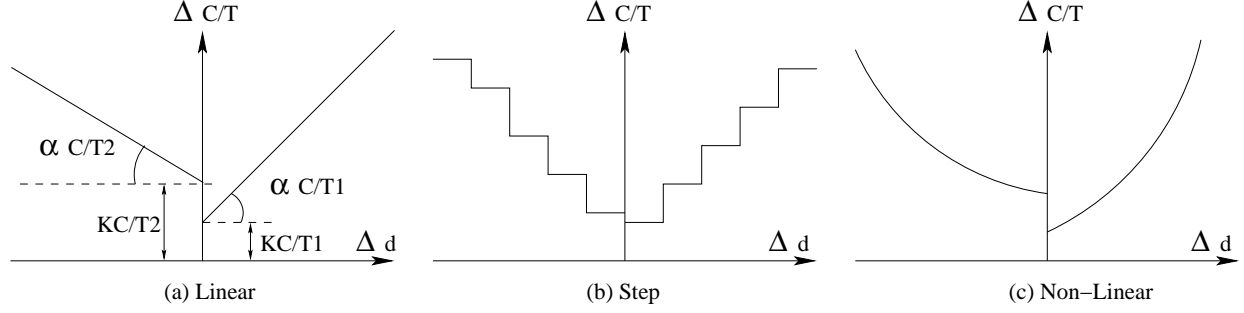
Figure 16: Relationship between dimensional change and cost/time change

(geometric) features and are classified into two categories: fixed and variable. The fixed redesign cost/time is the "setup" cost/time for undertaking a dimensional modification. It is invariant in the magnitude of the dimensional modification. If a perfect match is found, the fixed cost/time is zero. The variable redesign cost/time is a function of the amount of the corresponding dimensional change of associated feature(s). The fixed cost (or time) for increasing a dimension is represented by $KC1$ (or $KT1$), and for decreasing it is represented by $KC2$ (or $KT2$).

For the variable part, there are many kinds of functional relationships between the dimensional change and the cost/time change. For example, let d be the nominal dimension of a feature; δd be the dimensional change and δc and δt be the cost and time changes, respectively. Fig. 16 shows three kinds of relationships: linear, step and non-linear. In this section, for the sake of simplicity, it is assumed that the time and cost changes are linear to the dimensional changes of the form (geometric) features. If it is not linear, other non-linear optimization methods can be readily used but the basic idea of this research remains in place. Section 4 considers a detailed example with non-linear cost/time changes for illustrating this. Returning to the linear case for ease of understanding, for each KCDC of each component, let $\alpha C1$ (or $\alpha T1$) represent the proportional cost (or time) for increasing the dimension, and $\alpha C2$ (or $\alpha T2$) represent the proportional cost (or time) for decreasing the dimension. Combining the fixed and variable parts:

$$\delta C = \begin{cases} KC1 + \alpha C1 \times \delta d & if \quad \delta d > 0; \\ 0 & if \quad \delta d = 0; \\ KC2 + \alpha C2 \times \delta d & if \quad \delta d < 0. \end{cases} \tag{1}$$

$$\delta T = \begin{cases} KT1 + \alpha T1 \times \delta d & if \quad \delta d > 0; \\ 0 & if \quad \delta d = 0; \\ KT2 + \alpha T2 \times \delta d & if \quad \delta d < 0. \end{cases} \tag{2}$$

Equations 1 and 2, distinguish dimensional changes into three categories: dimensional increase, perfect match and dimensional decrease. For perfect match, there are no redesign cost and time. For dimensional increase and decrease, there are different time and cost coefficients.

16

Therefore, data about the coefficients $KC1, KC2, \alpha C1, \alpha C2, KT1, KT2, \alpha T1$, and $\alpha T2$ needs to be collected for each feature of each component. Since it is not our focus to develop methods to get reasonable values for these coefficients, these values are assumed to be readily available and are stored in the company database. Certainly, the values may vary from company to company and from component to component. The database can also be extended in the dimension of the nominal value, such that modification coefficients are available for each nominal dimension. This research recognizes that this data building is non-trivial in the amount of effort initially, nevertheless, it would be recovered by the automated evaluation power of the variant design methodology on repeated use.

3.6 Mixed Integer Programming (MIP) Formulation

The basic approach to perform the variant design is to formulate the variant design of each CG as an MIP (mixed integer linear or non-linear programming) problem and then combine them as a comprehensive optimization problem for the variant design of the entire assembly product.

3.6.1 Formulating the Variant Design of a FCG as an MIP

The variant design of an FCG can be formulated as the following MIP.

Indices:

i : Components. $i = 1 \ldots n$.
j : Alternatives of component i. $j = 1 \ldots a_i$.
k : Mating features of component i. $k = 1 \ldots f_i$.
k_i : The mating feature of component i involved in the FCG.
l : Types of dimensional changes. $l = 1(increase), 2(decrease)$.

Constants:

n : Number of components in the FCG.
a_i : Number of alternatives of component i.
f_i : Number of mating features of component i.
$\omega, 1 - \omega$: Weights of redesign cost and time ($0 \leq \omega \leq 1$).
$KC_{ijk_il}, \alpha C_{ijk_il}$: Fixed and variable cost coefficients incurred by dimensional change l for the jth alternative of component i, which is associated with feature k_i.
$KT_{ijk_il}, \alpha T_{ijk_il}$: Fixed and variable time coefficients incurred by dimensional change l for the jth alternative of component i, which is associated with feature k_i.
d_{ijk_i} : The current dimension of the jth alternative of component i, which is associated with feature k_i.
M : A large constant, greater than the largest KCDC dimension.

Decision Variables:

$$X_{ij} = \begin{cases} 1 & \text{if the } j\text{th alternative of component } i \text{ is selected;} \\ 0 & \text{Otherwise.} \end{cases} \tag{3}$$

$$d_{ik_i} : \text{The final matching dimension for the } k_i\text{th feature of component } i. \tag{4}$$

$$\Delta_{ijk_il} : \begin{array}{l} \text{Dimensional changes of the } j\text{th alternative of component } i, \\ \text{which is associated with feature } k_i.; \end{array} \tag{5}$$

$$Z_{ijk_il} = \begin{cases} 1 & \text{For feature } k_i, \text{ if the } j\text{th alternative of component } i \text{ is modified in mode } l \\ 0 & \text{Otherwise.} \end{cases} \tag{6}$$

Objective:

$$Min \sum_{i=1}^{n} \sum_{j=1}^{a_i} \sum_{l=1}^{2} ((\omega \times KC_{ijk_il} + (1-\omega) \times KT_{ijk_il}) \times Z_{ijk_il} \tag{7}$$

$$+ (\omega \times \alpha C_{ijk_il} + (1-\omega) \times \alpha T_{ijk_il}) \times \Delta_{ijk_il})$$

Subject to:

$$\sum_{j=1}^{a_i} X_{ij} = 1 \qquad\qquad for \ i = 1 \ldots n \tag{8}$$

$$\Delta_{ijk_i1} \geq d_{ik_i} - d_{ijk_i} - M(1 - X_{ij}) \qquad\qquad for \ i = 1 \ldots n, j = 1 \ldots a_i \tag{9}$$

$$\Delta_{ijk_i2} \geq d_{ijk_i} - d_{ik_i} - M(1 - X_{ij}) \qquad\qquad for \ i = 1 \ldots n, j = 1 \ldots a_i \tag{10}$$

$$Z_{ijk_il} \geq \Delta_{ijk_il}/M \qquad\qquad for \ i = 1 \ldots n, j = 1 \ldots a_i, l = 1, 2 \tag{11}$$

$$\sum_{j=1}^{a_i} \sum_{l=1}^{2} Z_{ijk_il} \leq 1 \qquad\qquad for \ i = 1 \ldots n \tag{12}$$

$$d_{1k_1} = d_{2k_2} = \cdots = d_{nk_n} \tag{13}$$

Binary:

$$X_{ij}, Z_{ijk_il} \in \{0, 1\} \qquad\qquad for \ i = 1 \ldots n, j = 1 \ldots a_i, l = 1, 2 \tag{14}$$

Bounds:

$$\Delta_{ijk_il} \geq 0 \qquad for \ \ i = 1 \ldots n, j = 1 \ldots a_i, k = 1 \ldots f_i, l = 1, 2 \qquad (15)$$

$$d_{ik} > 0 \qquad for \ \ i = 1 \ldots n, k = 1 \ldots f_i \qquad (16)$$

In this formulation, Equation 7 is the objective function which minimizes the weighted redesign costs and times incurred by the dimensional changes. When $\omega = 0$, the entire emphasis is on redesign time, while when $\omega = 1$, the entire emphasis is on redesign cost. The designer can select an appropriate value $0 \leq \omega \leq 1$ to appropriately weight both criteria (see also Section 4.2). Constraint 8 guarantees that one and only one alternative of each component is selected. Constraints 9 and 10 guarantee that the dimensional changes no matter it is increased or decreased are non-negative. Constraints 11 and 12 ensure that, for the dimension associated with each feature of a component, it is either modified or is a perfect match. Constrain 13 sets up the dimensional relationships within each constraint group. Equation 14 identifies the binary variables. Finally, Equations 15 and 16 claim the bounds for those real value decision variables.

3.6.2 Formulating the Variant Design of an ACG as an MIP

No matter how many components are involved in an FCG, the structure of it can be taken as one dimension because the center lines of all the mating features are aligned. However, the structure of an ACG may be very complex as discussed in Section 3.3. In this paper, let us constrain our scope to one dimensional ACG. Two and three dimensional ACGs can be decomposed into one dimensional ACGs and the then the same method can be employed.

The procedure of formulating the variant design of an ACG as an optimization program is similar to that of an FCG. The only difference lies in the representation of the relationships between involving dimensions. For an FCG, the involving dimensions should be equal to each other as shown in Equation 13. In an ACG, the involving dimensions form a loop. As a result, the sum of some of the involving dimensions equals to the sum of the remaining dimensions which is represented as the following equation in which d_{k_i} is the dimension of the ith component which is associated with feature k_i.

$$\sum_{i=1}^{n-m} d_{ik_i} = \sum_{i=n-m+1}^{n} d_{ik_i} \qquad for \ \ i = 1 \ldots n \qquad (17)$$

3.7 Summary of System Design and Use

This section summarizes the modules involved in building a system that embodies the assembly variant design system methodology. This is presented in the flowchart of Fig. 17. Once the system has been built, it can be used repeatedly to conduct variant design activities in response to varying customer specifications. During use, the system can benefit from diverse components available in distributed databases of the agile manufacturing environment. Repeated use of the system is summarized in the flowchart of Fig. 18. It is important to indicate that the building the system is a one-time activity, whose efforts can be recovered easily by repeated use of the system for proliferating multiple variants rapidly.
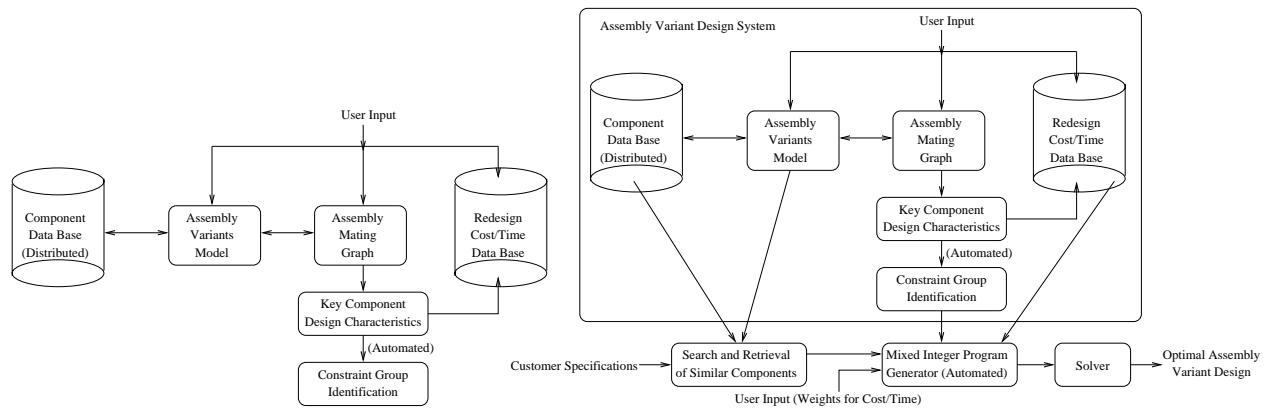
Figure 17: Building the System   Figure 18: Using the System

4 Software Implementation and Results

4.1 Software Development

The assembly variant design methodology developed in Section 3 has been implemented in C++ on the PC/Linux platform. The similar components and corresponding assembly mating graphs are taken as input. Then the constraint groups are identified and finally the variant design process is formulated as a mixed integer (non)linear programming problem which is then solved using CPLEX [15]. Since the weighted sum optimization method has been used to leverage the impact of redesign cost and time, different combinations of the weight of redesign cost and time have been tested.

4.2 Examples and Results

In this section, the variant design of the medium speed wheel supportive assembly has been formulated as a Mixed Integer Quadratic Program (MIQP). The process begin with specifying the time and cost coefficients as well as nominal dimensions of each component. For simplicity, these are generated randomly. For the time and cost coefficients, the nominal values are different from component to component, similar to

the nominal dimensions of each component. Let nv be the nominal values for either the time and cost coefficients or the nominal dimensions, the range of the random numbers generated can be expressed as $U(nv - a, nv + b)$. Since there are too many tables of the time and cost coefficients for each alternative of each component, they are omitted in this paper. Details may be found in [30]. Table 2 presents the representative values for the Buffet1. Further to illustrate the methodology on non-linear cost and time relationship, a quadratic relationship is considered for the changes in outer diameters of the Shaft, Buffet1 and Buffet2. All other changes are assumed linear.

In the demonstration, it is assumed that the number of alternatives for the Shaft, Spacer1, Base, Buffet1, Bearing1, Wheel, Bearing2, Buffet2, Spacer2 and Screw are 3,2,3,2,2,3,2,2,2 and 2, respectively. Though the number of alternatives for each component is quite small, there are 1728 possible assembly design schemes.

Tables 3 through 12 specify the nominal dimensions of related KCDCs of each alternative of the components. In these tables, the first column indicates the alternative while the remaining columns indicate the dimensions of related KCDCs.

The optimization model obtained was a mixed integer quadratic program (MIQP), which was solved using the MIQP solver of CPLEX 9.0 using the "optimize" function. CPLEX MIQP solver can be used in cases when the coefficient matrix of quadratic terms is positive semi-definite for minimization problems (i.e., convex quadratic minimization problems) or negative semi-definite for maximization problems (i.e., concave quadratic maximization problems). The results are shown in the following sections.

When ω is set to 0.5, the optimal solution is obtained in 4198.42 seconds with the objective function of 69.475. The alternative component selection and the final dimensions for each component are shown in Table 13 in which the perfect original matching dimensions are marked with underlines.

When ω is set to 1.0 and 0.0 respectively, the optimal solutions were obtained in 3720.78 and 2112.97 seconds with the objective functions of 66.379 and 65.69 respectively. The alternative selection and the final dimensions for each component are shown in Tables 14 and 15 in which the perfect original matching dimensions are marked with underlines.

4.3 Analysis

In this research, the weighted sum method has been used to generate the optimal assembly design. In the above numeric examples, three scenarios have been implemented. When $\omega = 0.5$ such that both the time and cost factors are taken into consideration with equal weights. When $\omega = 1.0$, only redesign cost is considered. And when $\omega = 0.0$, only redesign time is considered. The examples demonstrate that the optimal design scheme can be obtained in reasonable time (ranging from 35 to 70 minutes) for this prototypical problem.

When the redesign cost and time factors are set to different weights, the obtained optimal design schemes are possibly different. This provides the flexibility to the designer or the product manager.

Now, let us interpret the numerical results by answering the following questions.

- For each component, why does one alternative get selected over the other?

- When the ω weight is changed, why something else get selected?

4.3.1 Why does one alternative get selected over the other?

Redesign cost/time are distinguished into two categories: fixed (setup) and proportional cost/time which are expressed in Equations 1 and 2. For each key component design characteristic associated with the mating feature(s), the redesign cost/time equals to zero if it is a perfect match which means that the original KCDC value happens to be the same as the optimal value in the corresponding assembly design. However, each dimensional modification is penalized by a setup cost/time coefficient and a proportional coefficient. As a result, the optimizer invoked to solve the compounded MIQP drives the optimal assembly design scheme to the original KCDC values of constituent components as much as it can since the objective is to minimize the overall weighted redesign cost and time incurred by the dimensional changes. As a result, the alternative with matching dimension(s) is selected if there is perfect matching dimension(s). Otherwise, the alternative with less weighed dimensional changes is selected.

Let us take the variant design of the medium speed wheel supportive assembly with $\omega = 0.5$ as an example. Six out of ten components have some matching dimensions in the optimal assembly design. Therefore, the alternative selection tends to favor those components. For example, the first dimension of alternative 1 of the Screw is a perfect match, hence alternative 1 is selected. For the Shaft, the first and third dimension of alternative 2 are matching dimensions, hence alternative 2 is selected over the other two alternatives.

For the two alternatives of Bearing1, they have no matching dimensions. However the overall dimensional changes for the first alternative $(8.843 - 7.416 + 10.942 - 10.402 + 20.858 - 20.623 = 2.2)$ is less than that of the second alternative $(8.997 - 7.416 + 10.402 - 10.053 + 20.858 - 20.563 = 2.22)$. Therefore, the first alternative is selected. This is only a simple way to explain the decision making process when there are no matching dimensions. Actually, these dimensional changes should be multiplied by corresponding cost and time coefficients and then should be weighted before the decision can be made.

4.3.2 When the ω weight is changed, why something else get selected?

Let us take the Buffet1 as an example to explain why the alternative selection has been swapped when $\omega = 1.0$ and $\omega = 0.0$. The time and cost coefficients for the alternatives of Buffet1 are shown in Table 2.

When $\omega = 1.0$, only redesign cost is considered. The incurred redesign costs of the two alternatives are calculated as follows:

$$\Delta C1 = 4.227 + (14.680 - 14.440) \times 0.482 + 4.635 + (2.508 - 2.462) \times 0.566$$
$$+ 4.738 + (10.606 - 10.402) \times 0.581 = 13.860$$

$$\Delta C2 = 2.351 + (14.759 - 14.680) \times 0.208 + 2.805 + (2.462 - 2.401) \times 0.258$$
$$+ 2.668 + (10.402 - 10.35) \times 0.232 = 7.868$$

Since the overall redesign cost of the second alternative is less than that of the first one, the second alternative is selected.

Similarly, when $\omega = 0.0$, only redesign time is considered. The incurred redesign times of the two alternatives are calculated as follows:

$$\Delta T1 = 2.628 + (14.680 - 14.440) \times 0.245 + 2.625 + (2.569 - 2.508) \times 0.257$$
$$+ 2.189 + (10.606 - 10.402) \times 0.270 = 7.572$$

$$\Delta T2 = 4.612 + (14.759 - 14.680) \times 0.538 + 4.912 + (2.569 - 2.401) \times 0.546$$
$$+ 4.306 + (10.402 - 10.35) \times 0.422 = 13.986$$

Since the overall redesign time the first alternative is less than that of the second, the first alternative is selected.

This numerical example shows that different alternatives of a component are selected when the weights of redesign cost and time change because the weighted redesign effort is changed in different situations. The above example demonstrates how these decision choices are made. A closer observation of the situation allows us to quickly estimate the result. There are two specific features of the scenario shown in the above example:

- There are no perfect matching original dimensions.

- The redesign cost coefficients of the first alternative are much larger than those of the second alternative.

Based on these observations, we can easily conclude that the second alternative will be selected when only

the redesign costs are considered and the first alternative will be selected when only the redesign times are concerned without precise calculation.

4.4 Discussion

In this paper, only the incurred redesign cost and time are considered in the variant design process. This forms the backbone of the variant design methodology. Whenever necessary, it can be enriched by taking more aspects interested by the designer or customer into consideration by collecting more data and by modifying the objective function. For example, the normal cost and time of each component can be added to the objective function if they are important to the designer's decision making. Similarly, similarity match can also be added to the objective function if each of the component alternative represents a different degree of matching to customer specifications. The variant design methodology will serve as an engine into which the designer can feed such cost, similarity match or profitability information they are interested and get the optimal redesigns they desire.

In addition to various cost criteria, the customer might have some permissible "limits" on cost or redesign time. The mathematical programming model is quite flexible to incorporate such issues as constraints. Other design constraints such as parameter limits (e.g., minimal shaft diameter) can also be readily included in the model.

5 Conclusions and Further Work

In this research, a framework for generating new assembly variant design schemes based on mature component designs that might be dispersed at geographically distributed partner sites has been developed. The assembly variant design methodology elaborated in this research caters for the new demand proposed by the agile manufacturing paradigm.

The intention for this research is to advance the scope of variant design from a single component or a set of piece parts into complex assembly products. This demands for new variant design methodology. The assembly variant design methodology developed in this research is tightly coupled with the proposed assembly modeling concept proposed in [31]. The assembly variant design is initiated by the new customer needs which is decomposed to the requirements at the component level. The search and retrieval of matching components are performed by working on the AVM. Then the Constraint Groups (CG) are identified by manipulating the corresponding AMG. The variant design is finally formulated as a mixed integer linear or nonlinear programming problem. The generated MIP can be solved using a standard solver which demonstrates the efficiency of the methodology. For larger and more complex assemblies, math programming based heuristics can be employed if the computational times exceed permissible levels.

In conclusion, the theoretical contribution of this study can be outlined as follows:

- A framework that facilitates the generation of new assembly designs based on mature component designs which may be distributed at different partner sites is developed.

- Complementary assembly modeling concept is proposed which is finally instantiated as the hierarchical assembly model or Assembly Variants Model (AVM) and relational assembly model or Assembly Mating Graph (AMG). These models can be used to expedite the application in assembly variant design, assembly planning and disassembly planning.

- A variant design methodology that advances the variant design scope to complex assemblies is implemented.

The research can be furthered from the following perspectives.

- Although the complementary assembly modeling concept is proposed and AVM and AMG are developed, our attention is focused on how to organized assembly related information so that assembly applications such as assembly variant design, assembly planning and disassembly planning can be expedited. The research can be furthered by developing methods or procedures to generate the AVM and AMG from heterogeneous databases automatically.

- For simplicity, the scalarization method is used to generate the optimal assembly designs. In future research, the redesign cost and time can be taken as competitive factors so that the utility theory or game theory can be exploited.

- We are dealing with the variant design complex assemblies which includes lots of geometric information. It would be nice to have such a variant design environment in which the variant design process can be visualized. Virtual reality technology has the potential to achieve this end.

Acknowledgment

References

[1] Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin. *Network Flows*. Printice-Hall, 1990.

[2] O. Akin. *Psychology of architectural design*. Pion, London, 1986.

[3] D. Allen and P. Smith. Part classification and coding. Monograph no. 3, Brigham Young University, CAM Software Laboratory, 1980.

[4] D.K. Allen. Group technology. *The Journal of Applied Manufacturing Systems*, 6(2):37–46, 1994.

[5] R. Anderl and R. Mendgen. Parametric design and its impact on solid modeling applications. In *ACM Solid Modeling 1995, Salt Lake City, Utah*, pages 1–12, 1995.

[6] STEP AP224. Mechanical product definition for process planning using form features. *Document ISO TC184/WG3 N264(T7), National Institute of Standards and Technology, USA*, 1996.

[7] X. Chen and C.M. Hoffman. Design of feature-based and constraint-based cad. In *ACM Solid Modeling 1995, Salt Lake City, Utah*, pages 13–19, 1995.

[8] B.S. DSouza and T.W. Simpson. A genetic algorithm based method for product family design optimization. In *Proceedings of DETC02, Montreal Canada, September, 2002, CD#:DETC2002/DAC34106*, 2002.

[9] S. Finger and J.R. Dixon. A review of research in mechanical engineering design. part i: Descriptive, prescriptive, and computer-based models of design process. *Research in Engineering Design*, 1:51–67, 1989.

[10] J.E. Fowler. Variant design for mechanical artifacts: A state-of-the-art survey. *Engineering with Computers*, 12:1–15, 1996.

[11] R.L. Francis, L.F. McGinnis, and J.A. White. *Facilities Layout and Location*. 2nd ed. Prentice Hall, 1992.

[12] J.K. Allen G. Hernandez and F. Mistree. Design of hierarchic platforms for customizable products. In *Proceedings of DETC02, Montreal Canada, September, 2002, CD#:DETC2002/DAC34095*, 2002.

[13] S.L. Goldman, R.N. Nagel, and K. Preiss. *Agile competitors and virtual organizations: strategies for enriching the customer*. Van Nostrand Reinhold, New York, 1995.

[14] N.L. Hyer and U. Wemmerlov. Group technology oriented coding systems: Structures, applications, and implementation. *Production and Inventory Management*, 26:55–78, 1985.

[15] ILOG. *ILOG CPLEX 9.0, User's Manual*. www.ilog.com, 2003.

[16] S. Iyer and R. Nagi. Automated retrieval and ranking of similar parts in agile manufacturing. *IIE Transactions on Design and Manufacturing, special issue on Agile Manufacturing*, 29(10):859–876, 1997.

[17] A. Kinsey. Automated generation of group technology codes from a pdes product information model. M.S. thesis, Systems Research Center, University of Maryland, 1992.

[18] V. Krishnan, S.D. Eppinger, and D.E. Whitney. Simplifying iterations in cross-functional design decision making. *Journal of Mechanical Design*, 119(4):485–494, 1997.

[19] A. Messac, M. Martinez, and T. Simpson. Introduction of a product family penalty using physical programming. *ASME Journal of Mechanical Design*, 124:164–172, 2002.

[20] I. Minis and et. al. Optimal selection of partners in agile manufacturing. Technical report, University of Maryland, 1995.

[21] S.H. Mullins and D.C. Anderson. Automatic identification of geometric constraints in mechanical assemblies. *Computer Aided Design*, 30(9):715–726, 1998.

[22] R.U. Nayak, W. Chen, and T. Simpson. A variation-based method for product family design. *Engineering Optimization*, 34(1):65–81, 2002.

[23] OIR and Multi-M. *Code Book and Conventions*. Organization for Industrial Research, Waltham, MA, 1986.

[24] A. Opitz. *A Classification System to Describe Workpieces*. Pergamon Press, 1970.

[25] Phillip F. Ostwald. *Engineering Cost Estimating*. Prentice Hall, New Jersey, 1992.

[26] I. Prebil, S. Zupan, and P. Lučič. Adaptive and variant design of rotational connections. *Engineering with Computers*, 11:83–93, 1995.

[27] Y. Shimomura, M. Yoshioka, H. Takeda, Y. Umeda, and T. Tomiyama. Representation of design object based on the functional evolution process model. *ASME Journal of Mechanical Design*, 120:221–229, 1998.

[28] J. Stutz and R.L. Kashyap. Improving variant design of mechanical systems through functional relationships. *ASME International Computers in Engineering Conference and Exposition*, 1:151–159, 1989.

[29] N.P. Suh. *The Principles of Design*. Oxford University Press, New York, New York, 1990.

[30] Aihu Wang. *Variant Design of Complex Assemblies in Agile Manufacturing*. Ph.D. thesis, SUNY at Buffalo, Buffalo, NY, USA, 2001.

[31] Aihu Wang, Bahattin Koc, and Rakesh Nagi. Complex assembly variant design in agile manufacturing. Part I: System architecture and assembly modeling methodology. *IIE Transactions on Design and Manufacturing*, 2004.

[32] I. Watson and S. Perera. Case-based design: A review and analysis of building design applications. *AIEDAM*, 11:59–87, 1997.

[33] D.E. Whitney. From geometric modeling to product data models: Collaboration between engineering, computer science, and industry at leeds university. World Wide Web, *http://web.mit.edu/ctpid/www/Whitney/Europe/LEEDS.pdf*, 1995.

Altern	KCDC	KC(1)	KC(2)	αC(1)	αC(2)	KT(1)	KT(2)	αT(1)	αT(2)
	OuterDiameter1	4.227	4.769	0.482	0.440	2.628	2.604	0.245	0.247
1	Length2	4.598	4.635	0.571	0.566	2.625	2.721	0.257	0.238
	InnerDiameter1	4.184	4.738	0.511	0.581	2.243	2.189	0.261	0.270
	OuterDiameter1	2.585	2.351	0.249	0.208	4.741	4.612	0.524	0.538
2	Length2	2.805	2.149	0.258	0.287	4.912	4.615	0.546	0.409
	InnerDiameter1	2.668	2.977	0.232	0.257	4.306	4.174	0.422	0.574

Note: (1) implies factor for increase in dimension and (2) implies factor for decrease in dimension.

Table 2: Time and Cost Coefficients of the Buffet1

Alternative	Length1	OuterDiameter1	InnerDiameter1
1	37.725	10.199	8.694
2	37.616	10.655	8.544
3	37.098	10.545	8.050

Table 3: Original KCDC Dimensions of the Shaft

Alternative	Length1	InnerDiameter1
1	2.975	10.464
2	2.970	10.728

Table 4: Original KCDC Dimensions of the Spacer1

Alternative	Length2	InnerDiameter1	Length1	InnerDiameter2	Height1
1	34.531	14.680	24.693	14.372	40.389
2	34.050	14.810	24.170	14.061	40.311
3	34.784	14.212	24.322	14.485	40.009

Table 5: Original KCDC Dimensions of the Base

Alternative	OuterDiameter1	Length2	InnerDiameter1
1	14.440	2.508	10.606
2	14.759	2.401	10.350

Table 6: Original KCDC Dimensions of the Buffet1

Alternative	Length1	InnerDiameter1	OuterDiameter1
1	8.843	10.942	20.623
2	8.997	10.053	20.563

Table 7: Original KCDC Dimensions of the Bearing1

Alternative	InnerDiameter2	Length3	InnerDiameter1	InnerDiameter2	Height1
1	20.864	4.387	10.306	20.284	38.027
2	20.613	4.391	10.019	20.031	38.456
3	20.858	4.937	10.402	20.727	38.612

Table 8: Original KCDC Dimensions of the Wheel

Alternative	Length1	InnerDiameter1	OuterDiameter1
1	8.808	10.875	20.583
2	8.595	10.079	20.793

Table 9: Original KCDC Dimensions of the Bearing2

Alternative	Length2	InnerDiameter1	OuterDiameter1
1	2.462	10.591	14.589
2	2.569	10.816	14.673

Table 10: Original KCDC Dimensions of the Buffet2

Alternative	Length1	InnerDiameter1
1	2.344	10.010
2	2.267	10.342

Table 11: Original KCDC Dimensions of the Spacer2

Alternative	Length1	OuterDiameter1
1	7.232	8.771
2	7.372	8.870

Table 12: Original KCDC Dimensions of the Screw

	Component	Selection	KCDC1	KCDC2	KCDC3	KCDC4	KCDC5
1	Shaft	2	<u>37.616</u>	10.402	<u>8.544</u>		
2	Spacer1	1	<u>2.975</u>	10.402			
3	Base	1	<u>34.531</u>	<u>14.680</u>	<u>24.693</u>	14.680	<u>40.389</u>
4	Buffet1	1	14.680	2.462	10.402		
5	Bearing1	1	7.416	10.402	20.858		
6	Wheel	3	<u>20.858</u>	<u>4.937</u>	<u>10.402</u>	20.858	<u>38.612</u>
7	Bearing2	2	7.416	10.402	20.858		
8	Buffet2	1	<u>2.462</u>	10.402	14.680		
9	Spacer2	2	2.975	10.402			
10	Screw	1	<u>7.232</u>	8.544			

Table 13: Alternative Selection and Optimal Dimensions ($\omega = 0.5$)

	Component	Selection	KCDC1	KCDC2	KCDC3	KCDC4	KCDC5
1	Shaft	2	<u>37.616</u>	10.402	<u>8.544</u>		
2	Spacer1	1	2.344	10.402			
3	Base	1	<u>34.531</u>	<u>14.680</u>	<u>24.693</u>	14.680	<u>40.389</u>
4	Buffet1	2	14.680	2.462	10.402		
5	Bearing1	2	7.416	10.402	20.858		
6	Wheel	3	<u>20.858</u>	<u>4.937</u>	<u>10.402</u>	20.858	<u>38.612</u>
7	Bearing2	1	7.416	10.402	20.858		
8	Buffet2	1	<u>2.462</u>	10.402	14.680		
9	Spacer2	1	<u>2.344</u>	10.402			
10	Screw	2	<u>7.372</u>	8.544			

Table 14: Alternative Selection and Optimal Dimensions ($\omega = 1.0$)

	Component	Selection	KCDC1	KCDC2	KCDC3	KCDC4	KCDC5
1	Shaft	1	<u>37.725</u>	10.402	<u>8.694</u>		
2	Spacer1	1	<u>2.975</u>	10.402			
3	Base	1	<u>34.531</u>	<u>14.680</u>	<u>24.693</u>	14.680	<u>40.389</u>
4	Buffet1	1	14.680	2.569	10.402		
5	Bearing1	1	7.308	10.402	20.858		
6	Wheel	3	<u>20.858</u>	<u>4.937</u>	<u>10.402</u>	20.858	<u>38.612</u>
7	Bearing2	2	7.309	10.402	20.858		
8	Buffet2	2	<u>2.569</u>	10.402	14.680		
9	Spacer2	2	2.975	10.402			
10	Screw	2	<u>7.372</u>	8.694			

Table 15: Alternative Selection and Optimal Dimensions ($\omega = 0.0$)