

# IFEED: INTERACTIVE FEATURE EXTRACTION FOR ENGINEERING DESIGN

**Hyunseung Bang**  
Cornell University  
Ithaca, NY, USA

**Daniel Selva**  
Cornell University  
Ithaca, NY, USA

## ABSTRACT

One of the major challenges faced by the decision maker in the design of complex engineering systems is information overload. Effective exploration of the design space requires the generation and analysis of numerous design alternatives, each represented by multiple design variables and corresponding objective values. When the size and dimensionality of the data exceeds a certain level, a designer may become overwhelmed and no longer be able to perceive or understand the underlying dynamics of the design problem at hand, which can result in premature or poor design selection. There exist various visual analytic tools designed to relieve the information overload, such as BrickViz, Cloud Visualization, ATSV, and LIVE, to name a few. However, most of them do not explicitly support the discovery of key knowledge about the mapping between the design space and the objective space, such as the set of high-level design features that drive most of the trade-offs between objectives. In this paper, we introduce a new interactive method, called iFEED, that supports the designer in the process of high-level knowledge discovery in a large, multiobjective design space. The primary goal of the method is to iteratively mine the design space dataset for driving features, i.e., combinations of design variables that appear to consistently drive designs towards specific target regions in the design space set by the user. This is implemented using a data mining algorithm that mines interesting patterns in the form of association rules. The extracted patterns are then used to build a surrogate classification model based on a decision tree that predicts whether a design is likely to be located in the target region of the tradespace or not based on the driving features only. The classification performance and the compactness of this tree are used as figures of merit to evaluate the set of driving features, and are provided to the user to facilitate iteration. Initially, the extracted patterns that constitute test nodes of the classification tree are likely to be primitive (simple combinations of design variables). The proposed method enables the user to build and test more compact and higher-level driving features through an interactive process. Higher level driving features will result in more compact classification trees while improving classification performance, helping the user to gain insights and build a mental model of how design variables are mapped into objective values. The validity and effectiveness of such approach are tested on a controlled experiment with human subjects.

## 1. INTRODUCTION

Many modern engineering systems such as aircrafts, nuclear power plants, and satellite systems are extremely complex, and their scale and scope can be immense [1]. Interactions between subsystems, sometimes even across different levels of hierarchies, may give rise to emergent behaviors that are difficult to predict. Designing such a complex engineered system still remains a challenging task.

After the introduction of the concept of decision-based design [2], most research efforts in engineering design have focused on developing fully-automated optimization technologies to find a single point solution that provides maximum “value”. Under this approach, various design alternatives are rank-ordered through mathematical calculation of the “value” of each design, which can be defined in various ways [3,4]. This requires *a priori* articulation of preferences among different attributes. In some cases, this is relatively simple; for example, for a for-profit organization, value can be defined as the expected utility of the net present value of the project over its lifetime, which incorporates both revenues and costs, opportunity costs, inflation, uncertainty and risk aversion. In other cases however, especially in the case of early-phase design (e.g. conceptual design and system architecting) of unprecedented systems in not-for-profit organizations, the design problems are usually ill-defined and the stakeholders may not know exactly what they really want until they have some idea of what the design space looks like. For example, consider the case of architecting the next generation Earth observing satellite system for scientific purposes. In this case the primary stakeholder is the scientific community, whose preferences are hard to express with a single-dimensional value function, since such function would have to factor in many different measurement attributes for which it is very hard to establish combined preferences, such as spatial resolution, temporal resolution, accuracy, spectral sampling, and so forth.

More recently, a new paradigm of “design by shopping” [5] has emerged as an alternative. This has led to the development of various multi-objective optimization techniques, visual analytics, and tradespace exploration tools [6,7]. Under this paradigm, a designer explores multiple design alternatives and chooses a design that best matches his or her preference. Such practice gives the designer more control over the design and selection process. This is inherently an interactive process, where the human designer and the computer have to be able to

communicate with each other effectively.

Communications between the human and the computer can occur in both directions; from the computer to the human, and vice versa. For example, the computer should be able to show the visualization of the tradespace and provide all the relevant information in a compact form. After observing the provided information, the designer should be able to express preferences between possibly-conflicting objectives and select a set of designs among the vast number of possible design alternatives to examine it further in detail.

The capability of exchanging information between the human and the computer is becoming more critical, as the improvements in computational power and the developments of efficient multi-objective optimization algorithms [8,9] have enabled the simulation of a large number of design alternatives. If the communication framework is not carefully designed, a human designer can be overloaded with too much information.

To address this issue, much work has explored new data visualization and tradespace exploration techniques [10–16]. However, most of the existing tools lack explicit support for high-level knowledge discovery and testing. Here, the high-level knowledge refers to the knowledge that would require more logical operators and design attributes to express, compared to low-level knowledge. For example, a designer may be interested in learning why certain designs are mapped to the particular location in the objective space. However, such information may not be explicitly available in existing visualization methods.

One possible way of encoding such information is as a rule that has the form of an if-then association: if feature  $X$  (design space) is present in architecture  $A$ , then architecture  $A$  is likely to be good. Using such information, we define a driving feature as the combination of design variables that appear to drive designs to a desired (target) region of the tradespace [17]. Throughout this paper, we use the terms “feature” and “pattern” interchangeably to mean a combination of design variables and their corresponding values. While many automatic feature selection and extraction algorithms exist [18–20], they suffer from the curse of dimensionality, since the space of candidate features is possibly unbounded. Therefore, the success of these algorithms in practical design problems requires the intervention of the human in the feature selection process.

In this paper, we propose and test an interactive strategy called iFEED (Interactive Feature Extraction for Engineering Design) that can extract a compact set of driving features of a design problem, represent its structure and assess its predictive power using classification trees. One of the major advantages of this tool is that the human designer can actively participate in the knowledge discovery process, and can get new insights on the functional mapping between the design space and the objective space. Ultimately, this will result in better designs selected and more value to the decision maker.

Through a controlled experiment, we compare the effectiveness of this strategy in helping the user learn the driving features of the problem compared with the current standards: advanced visual analytic techniques (scatter plots, dynamic filters) and automatic feature extraction techniques.

The rest of the paper is structured as follows: Section 2 provides a review on current visual analytic tools. Section 3 describes the research gap and the main goal of our research. Section 4 introduces iFEED, and describes what the major steps are in the method. Section 5 describes how preliminary experiment is designed to test the effectiveness of iFEED. Section 6 presents the result. Section 7 discusses the conclusions and Section 8 provides the limitations of this research and opportunities for future works.

## 2. LITERATURE REVIEW

With the emergence of the “design by shopping” paradigm [5], multi-objective optimization tools have been developed to generate a large number of design alternatives [21]. Each of these design alternatives are represented by a set of design variables and corresponding objective values. Because of the large size and the high dimensionality of the data, a human designer faces the challenge of information overload [22], which occurs when the amount and complexity of information available exceeds the human’s cognitive capability. Therefore, the need for powerful analytical tools to aid in tradespace exploration has grown in the past decade.

Various interactive visualization techniques have been developed in this regard. For instance, BrickViz [10] aims to reduce the designer’s cognitive load by representing groups of designs using boxes. Many other researches have focused on providing techniques to visualize high dimensional space. Cloud visualization [11] enables visualization of both design space and objective space simultaneously, and the link between them. It also accommodates multi-dimensional view using cascading windows. ARL Tradespace Visualizer [12,13] enables multi-dimensional analysis of the tradespace by creating glyph plots, linked-views, histogram, and parallel coordinate plots. Hyper-space Diagonal Counting (HSDC) method [15] visualizes the Pareto Frontier for an  $n$ -dimensional objective space. However, the neighborhood points are not maintained during the visualization process. Hyper-Radial Visualization (HRV) [16] suggests an improved method to visualize the Pareto Frontier for an  $n$ -dimensional objective space. The major function of most of these tools is visualizing the objective space to see the trade-offs.

Subsequent developments to some of these visualization tools have been to add new functions that incorporate more human inputs reflecting users’ preferences. A visual design steering method called Graph Morphing [23,24] enables the designer to pause the optimization process and “steer” it into certain direction. Visual steering commands implemented in [25] make use of user guided samplers to redirect or narrow down the search during the tradespace exploration. Chiu and Bloebaum also implemented a technique that generates new designs within the region of interest identified by the designer [26].

Another line of effort is being put on combining interactive visualization and automated analytic tools such as data mining [22,27–29]. A justification for this approach is that humans and computers are complementary in the knowledge discovery process: humans are good at thinking holistically and

recognizing patterns, whereas computers are good at introducing exhaustiveness, rigor, and speed in analyzing data.

Some researchers have taken this approach to help the human designers learn about the design problem better. The framework developed by Zhang et al. utilizes clustering algorithms to draw nested boxes, each of which represents a cluster at a different dimension or scale [30]. The user can interactively select different clustering dimensions and their order to provide a view of how objective values may vary depending on the changes of design variables.

Another tool called LIVE developed by Yan et al. combines the interactive visualization with data-oriented analytic tools such as decision trees and clustering [31]. In this tool, the user can interactively choose a group of designs as target designs and construct a decision tree. The decision tree will show the user what combination of design values may lead a design into the user-specified target area in the objective space. The user can also run clustering algorithms within the design space to see how different clusters of designs are mapped in the objective space.

### 3. RESEARCH GAP AND GOALS

In 2010, Bertini and Lalanne did a literature review on recent approaches that combine advanced technologies in information visualization and data mining to facilitate the process of knowledge discovery [29]. They argue that there still are unexplored opportunities in the way visualization and data mining are integrated. One of their suggestions is to make it possible for the user to build high-level hypotheses and complex models using patterns mined by computational methods. By building such high-level models, the user can gain some insights into the relationship between objective values and input design variables. The limitation that exists in most of the current visualization and visual analytic tools is that it is difficult, if not impossible, to build high-level models of the design problem in an explicit manner. The user may observe visualization outputs of different tools and deduce high-level models, but this process is still limited by the human cognitive capacity and the complexity of the visualized content. Moreover, it also requires some effort to test and evaluate these models, as they may need to be transformed into the form compatible with the analysis tool.

The framework introduced in [31] is a notable exception, as it has the capability of generating surrogate models based on classification trees. However, it does not go beyond low-level pattern detection, with each of the test nodes in the decision tree having only a single design variable and its corresponding range. As low-level features constitute each test node, the constructed decision tree is usually large in size, and it is difficult to draw a meaningful insight on the user’s part. Furthermore, the tool does not provide any platform that the user can use to input high-level features and test them.

### 4. IFEED

#### 4.1 Overview of iFEED

In this paper, we propose an interactive strategy called iFEED that can be used to extract a compact set of driving features and build a surrogate model with them. Here, driving

features are defined as sets of design variables that play major roles in defining the location of designs in a user-selected region of the objective space. iFEED is based on a dialogue (two-way communication) between the human designer and the computer analytic tool. The key idea is to start from a set of low-level driving features mined from an automated algorithm, and to progressively identify and build up higher level features by incorporating the human knowledge and problem-solving capability. The surrogate model constructed using low-level features will likely be complex and difficult for the user to draw any meaningful insight from it. On the other hand, the model constructed using high-level features may have a more compact form, and therefore easier to understand.

The immediate goal of iFEED is to let the user gain new insights on how design variables affect the objective values as he or she interacts with the tool. It has been demonstrated that the knowledge gained in this process can later be used in making optimization and tradespace exploration more efficient [17]. The process consists of 5 steps, as shown in Figure 1. The 5 main steps are: 1) formulation of candidate driving features; 2) Selection of a region of interest in the tradespace (e.g. high performance, low cost, and low Pareto ranking designs) and defining a label based on it; 3) Identifying a set of driving features from the set of candidate features that lead to the region of interest; 4) Building a compact form of classification using the driving features as predictors and the region of interest as the label; 5) Evaluation of the classifier. At this point, the user goes back to Step 1 to refine the candidate features, or terminates if he or she is satisfied with the results.

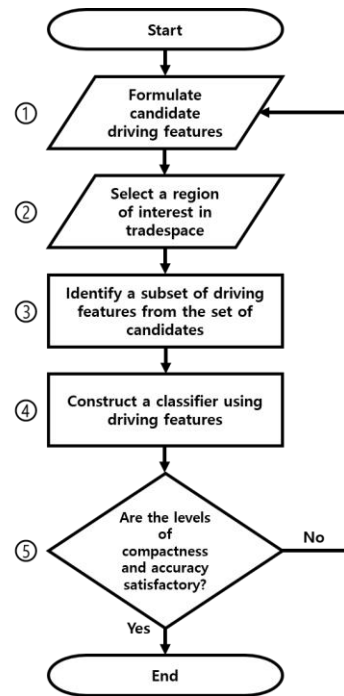
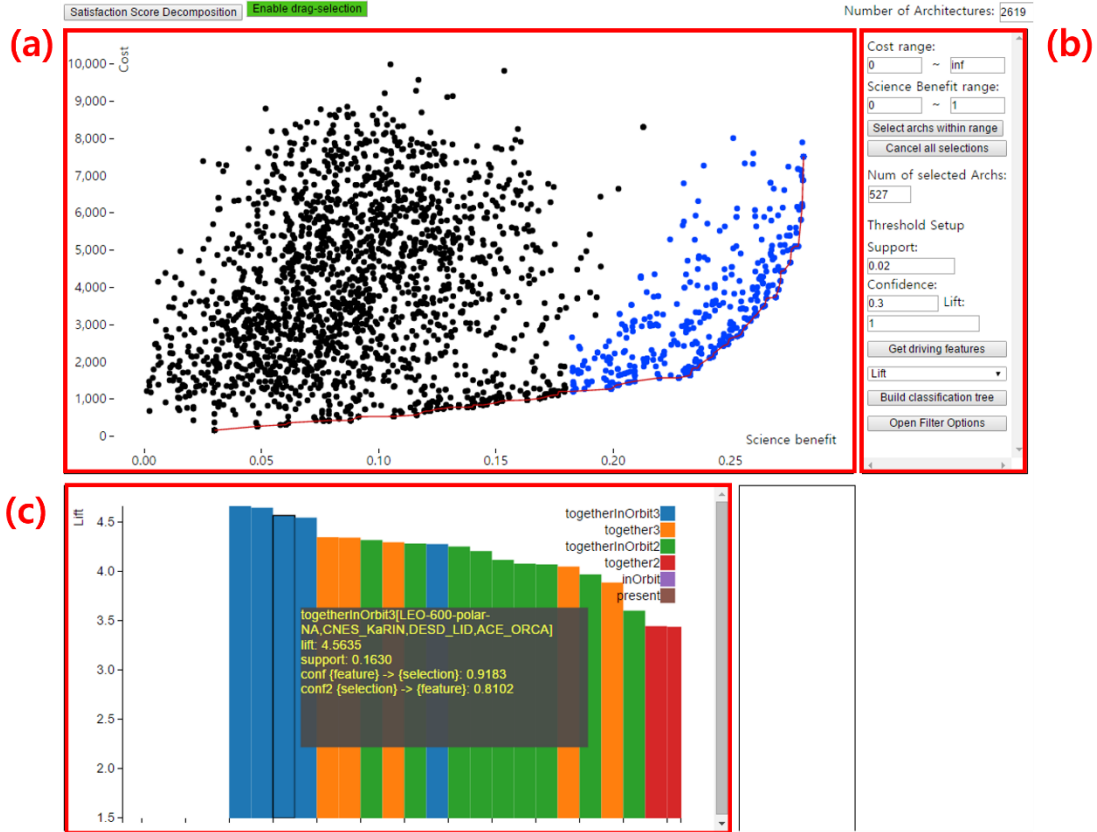


Figure 1. The flow chart showing major steps taken in iFEED for identifying and testing driving features.



**Figure 2.** The main interface of iFEED. (a) The scatter plot with interactive functions to select designs (b) Different menus for constructing driving features and setting their threshold values. (c) Bar charts displaying the predictive power of driving features. Each bar corresponds to one driving feature

Note that step 1 and step 2 rely on the human input to the computer, and step 3 and 4 are carried out by automated processes. Step 5 involves both components. To support such collaborative effort, both interactive visualization and data mining techniques are used to implement iFEED. The main interface of iFEED is shown in Figure 2. The following subsections will explain each of the steps in more detail.

## 4.2 Interactive Steps of iFEED

### 4.2.1 Formulation of candidate driving features

The first step in iFEED is to formulate candidate driving features using domain-specific knowledge and insights obtained by observing the structure of the design problem. The human designer pre-specifies the structure of all patterns to be searched, thereby limiting the feature space that the computer has to search to a tractable size. This step is justified, as the space of all possible design features is essentially unbounded in size and therefore too large to be searched exhaustively.

Some may point out that there exist various strategies in conventional data mining algorithms [18,19] that automate the process of reducing the feature space. While this is true, there is an advantage in having the user set the candidate driving features. Patterns specified by the human can be higher-level

compared to patterns mined from conventional data mining algorithms.

For example, consider the task of architecting an Earth observing satellite system, where the key design decision is to assign sets of remote sensing instruments to be used in different spacecraft. If we use automated data mining algorithms, the mined patterns will be simple combinations of design variables such as: ‘instrument A and B are assigned to spacecraft X’, or ‘instrument C is not assigned to spacecraft Y’. However, these patterns are low-level and therefore difficult to make useful interpretations. In contrast, the human designers with expert domain knowledge can come up with higher level patterns such as: ‘instrument A and B are not assigned to spacecraft flying in Sun-synchronous orbits with dawn-dusk local time of ascending node’, or ‘No more than 2 heavy instruments are assigned to each spacecraft’. These patterns are considered higher level, as they require more design variables and logical operators to express. Such high-level, compact patterns not only make the search process more efficient, but also make it easier for human designer to understand the result in later steps.

### 4.2.2 Selection of a region of interest in the tradespace

The second step is to select a region of interest in the objective space in order to mine patterns that appear more often in the selected region compared to the unselected region. The

selection can be made in two different ways. The first approach is to define an analytical function that differentiates the designs of interest from the rest. This function can be continuous (e.g., expected utility, or NPV), discrete (e.g., Pareto ranking), or binary (e.g., being on the Pareto front, objective values being within certain range). The second approach is to manually select a region or regions based on interactive visualization of the tradespace.

Both methods require human input, which is entered through the GUI shown in Figure 2. The main interface contains a scatterplot that displays design alternatives in objective space. The user can make selection of designs by setting target Pareto rankings, or range of objective values to be satisfied with each design. Another option is to make selection visually, by dragging mouse on to specific regions on the scatterplot.

#### 4.2.3 Identifying a set of driving features

The third step is to identify driving features using association rule mining [18], which is a method used to reveal interesting relations between variables in a database. A typical example is shown using market baskets. Given a database containing a large number of transaction records in a supermarket, we can discover that some items are often bought together. For instance, {peanut butter, jelly, bread}. From this set, we can generate an implication rule such as {peanut butter and jelly}  $\rightarrow$  {bread}. Likewise, we can define an association rule  $X \rightarrow Y$ , where  $X$  and  $Y$  are both combinations of binary attributes of a design. This rule suggests that the presence of a feature  $X$  implies  $Y$ . The major advantage of mining association rules is that the knowledge is encoded in a human-learnable way. Newell and Simon have shown that the way human experts reason can be modeled using logical rules [32].

Classification association rule mining extends this idea, by introducing class labels and treating them as additional binary attributes [33]. Then rules of form  $X \rightarrow c$  are only mined, where  $X$  is a set of attribute values and  $c$  is a class label. In iFEED,  $X$  represents features in the design space, and  $c$  indicates whether a certain design belongs to the targeted region in the objective space or not. Therefore, each association rule provides a link between the design space and the objective space.

Based on this structure, we can generate a large number of association rules. The subsequent mining process filters out rules that are statistically insignificant or not useful. In conventional association rule mining, two measures are often used: support and confidence. The support of a feature  $X$  is defined as the proportion of instances that contain  $X$  among all instances of design in the database. This can be written as:

$$\text{supp}(X) = \frac{|\{X \in t, t \in D\}|}{|D|} = P(X)$$

where  $X$  is a feature of interest,  $t$  is one particular design, and  $D$  is the dataset of multiple designs. A high support for  $X$  means that  $X$  appears frequently in the designs of the database. A minimum support threshold is used to filter out features that are statistically insignificant. The confidence of an association rule

$X \rightarrow Y$  is defined as the probability of a design containing feature  $Y$ , given that it contains  $X$ :

$$\text{conf}(X \rightarrow Y) = \frac{\text{supp}(X \rightarrow Y)}{\text{supp}(X)} = \frac{\text{supp}(X \cup Y)}{\text{supp}(X)} = P(Y|X)$$

Confidence is used as a level of strength of an association rule. For formal definition and detailed explanations, refer to [18].

In this study, we use an additional metric called lift (also called interest) [34] defined as following:

$$\text{lift}(X \rightarrow Y) = \frac{P(X \wedge Y)}{P(X)P(Y)}$$

This value is related to the level of statistical dependency between  $X$  and  $Y$ . If these two sets were independent, lift would be equal to 1.

Each classification association rule has a corresponding support, confidence, and lift. Rules that have all three of these values higher than their corresponding thresholds are accepted. Then the antecedent, or the left-hand-side of all classification rules become the driving features and are provided to the user through the GUI as shown in Figure 2c. The predictive power of driving features is presented using bar charts, of which the height can represent either support, confidence, or lift. The user can hover over each of the bar and relevant information is presented in the data tip. The thresholds for support, confidence, and lift can also be set by the user through the GUI.

#### 4.2.4 Building a classifier using driving features

Once the set of driving features has been identified, it can be used as a training set to train a classifier that classifies whether a certain design belongs to the target region or not. In iFEED, a classification tree is trained using the C4.5 algorithm [35]. A classification tree model is used, because it is known as one of the most human understandable classification models and it introduces hierarchical structure among different features. C4.5 follows a divide-and-conquer approach. At each node, it selects the feature that provides the largest information gain when a set of designs traced to that node are divided according to that particular feature. All features considered as the splitting criteria come from driving features identified in the previous step. If all of the designs in the remaining set are in the same class at particular node, it becomes a leaf node. The algorithm runs recursively until each path is connected to a leaf node. After the classification tree is constructed, it is displayed to the user through the GUI. The user can check the feature used at each test node, and see the classification statistics (the number of designs traced to that node, and the percentage of correctly classified designs) at each leaf node. The user can also interactively expand and collapse each node to adjust the pruning level of the tree.

The purpose of the classification tree is not to predict a certain metric precisely, but to provide hierarchical structure and priority information about the driving features, so that the user can gain insights about the relative importance of a feature compared to others. This may aid in identifying a small number

of features that are present in designs within a desired location in the objective space.

#### 4.2.5 Evaluation of the classifier

After the classifier is presented to the user, he or she can evaluate them based on the classification accuracy and the level of compactness as figures of merit. The confusion matrix can be used to define the classification accuracy of the tree. Both accuracy and compactness are dependent on the set of driving features used to train the classifier. For example, when the driving features only have small confidence and lift values, the classifier is likely to have low classification accuracy. This is because driving features with low confidence and lift are not prominent features that separates desirable designs from other designs well. In addition, when the set of driving features only contain low-level design features, the classification tree will have more nodes and therefore, will be larger in size. In contrast, high-level driving features will result in a decreased number of nodes and a smaller tree. Thus, the level of compactness can be used to check if driving features are high-level or not.

Based on these observations, the user can evaluate the driving features and their corresponding classification tree. If the classification accuracy or the level of compactness are too low, the user must go back to Step 1 and define new higher level candidate driving features. Otherwise, the user can decide to terminate the whole process.

#### 4.3 Implementation of iFEED

The iFEED tool is implemented following a client-server model. The client receives user inputs and visualize data, while the server does all the data handling and analysis. The client side includes the GUI, which is built using HTML, CSS, and JavaScript. Visualizations of data and interactive interface are made using a JavaScript library called D3. As the GUI is built on web technologies, it can be displayed using any web browser on any device regardless of hardware and operating system. The server side contains algorithms that generate the driving features using association rule mining, and construct classification trees. These algorithms are implemented using Java.

## 5. EXPERIMENT METHODS

The primary aim of iFEED is to help designers gain new insights on what combination of design features drives the trade-offs in the objective space. In order to test the effectiveness of iFEED, we designed a series of controlled experiments with human subjects. The experiment consists of two major steps called the learning session and the problem solving session. First, in the learning session, the human subjects are provided with iFEED or other variant tools to analyze thousands of design alternatives generated for a real world design problem. Then in the following problem solving session, a problem set is given to the human subjects to test how much knowledge and insights the subjects have gained during their interactions with the tool. After the problem set has been graded, the scores are used to check if the users have benefited in using iFEED compared to other tools. In this paper, we conduct a preliminary experiment with a limited

number of participants. The following sections provide details of the experimental setup.

#### 5.1 Hypotheses

The experiment aims to examine the efficacy of iFEED compared to pre-existing design and analysis tools. The effectiveness of iFEED can be measured in several aspects: the amount of factual knowledge that the user gains about the design space, the improvements in the user's ability to generate a point design that is considered to be good based on the user-defined value function, and the increase in user's trust on the tool. Based on the above, three hypotheses were developed.

- H1: iFEED enhances the users' understanding of the design problem and major trade-offs compared to pre-existing tools
- H2: iFEED enhances the users' ability to come up with a new, high-value design, compared to pre-existing tools
- H3: iFEED increases the users' trust on the tool, compared to pre-existing tools

The first hypothesis comes directly from the fact that iFEED is intended to aid users in analyzing and discovering new knowledge about the design problem. Using iFEED, the user can select the target region in the objective space and mine patterns that are frequent among the designs in the region. The user can further identify the structure and relative importance of these features using a classification tree. Then the user can derive and test higher level driving features based on this information. During this interactive process, the user will have chances to learn what design variables play major roles in locating a design into a certain region of the objective space.

The second hypothesis is developed from the assumption that, if the user has gained enough factual knowledge and is able to synthesize it, then he or she should be able to propose a design that is likely to be good according to the value function. It is assumed that a designer who has gained more knowledge will be able to propose designs with higher value.

The third hypothesis is based on another assumption that a user will trust the output of the tool more if he or she understands how it was obtained. In iFEED, the user plays an active role in the knowledge discovery process, making it clearer to the user as to how the resulting driving features are obtained.

In addition to these three hypotheses, one more hypothesis was added:

- H4: Introducing a hierarchical structure and the notion of relative importance among the set of driving features supports the knowledge discovery process

This hypothesis is added because it is not obvious that constructing a classification tree will add any value in presenting and testing driving features.

#### 5.2 Experiment conditions

In order to test the hypotheses, we divided all human subjects into four groups. Each group is given different set of tools to analyze data during the learning session, as summarized in Figure 3.

Conditions	Scatter plot	Filter	Interactive DF mining	Classification tree	Automated DF mining (apriori)
Baseline	Green	Green			
Interactive DF	Green	Green	Green		
iFEED	Green	Green	Green	Green	
Automated DF	Green	Green			Green

**Figure 3. Different set of tools provided to each group of test subjects. The green boxes represent that corresponding function is implemented in the given tool.**

The first group of participants are provided with a conventional analytic tool which includes a basic scatter plot and filters. The scatter plot displays a large number of design alternatives in the objective space. The test problem for this study has two objectives, so the scatter plot is displayed on a 2D plane. When there are more objectives, the scatter plot can be obtained either by forcing the user to choose 2 to 3 objective values or by implementing dimension reduction to project the objective space in 2D or 3D.

Filters allow the user to selectively highlight designs on the scatter plot based on their design variables. For example, if a user sets the range of values for a particular design variable, then the filter highlights all designs whose design variable has a value within the specified range. This helps the user to concentrate on a specific set of designs or learn how designs with the similar range of design inputs are distributed in the objective space.

The second group is provided with interactive driving feature mining along with a scatter plot and filters. The user can interact with the GUI and extract driving features using association rule mining. The user can also generate and test new driving features. The mined driving features can be sorted based on their lift, confidence, or support.

The next group is provided with the same tools as the previous group with the additional capability of generating classification trees, which adds hierarchical structure to the driving features. This group gets the full functionality of the iFEED method proposed in this paper. The user is instructed to try to find a compact form of classification tree by introducing high-level driving features.

The last group is provided with the tool that also consists of scatter plot and filters. In addition, the given tool is capable of extracting driving features automatically. In this case, the driving features are extracted with minimal interaction with the user. The target region in the objective space is automatically fixed and driving features are mined using apriori algorithm, which is one of the classical methods used in association rule mining [18]. The apriori algorithm includes automatic candidate generation steps and pruning method. No user input is used for setting the target region in the objective space or for generating candidate driving features.

### 5.3 Evaluation criteria

After each participant completes the learning session, they are given a problem set to solve. The problem set contains 5 questions asking for demographic information, 2 questions regarding the user’s trust towards the tool, and 15 questions asking about the design problem. Demographic information questions asked a participant’s age, gender, educational level, undergraduate major, and whether he or she had a prior engineering design experience in the same domain as that of the test problem (satellite system design). 15 questions that asked about the design problem have two different types: multiple-choice questions and design synthesis questions.

First, multiple-choice questions mostly test the user’s factual knowledge about the design problem. For example, some problems ask what range of objective values can be achieved when a constraint is applied to modify the boundaries of the design space. There are also problems that ask about the sensitivity of objective values to changes in design decisions. Other questions ask about the coupling effects between different design decisions, such as synergies or interference effects.

Another type of problems is the design synthesis task. Here, the user is asked either to propose a new “good” design or to improve the given design. It is assumed that, in order for the designer to successfully provide answers to synthesis type of questions, he or she needs to have enough of factual knowledge and be able to apply it to more complex conditions.

### 5.4 Test Problem

The experiment is carried out using a real-world problem previously studied by the authors [36]. The problem is to design a system architecture of an Earth Observing Satellite Systems that gives good value-to-cost to the climate scientific community. The system architecting problem has been simplified into a mathematical formulation of an assignment problem between a set of instruments and a set of orbits. Given a set  $P$  of candidate instruments and a set  $O$  of candidate orbits, the architectural space is defined as the mapping of  $P$  onto  $O$ .

Each instrument in  $P$  can be assigned to more than one orbit in  $O$ , or it may not be assigned to any orbit at all. The size of the architectural space is then defined as  $2^{|O||P|}$ , where  $|O|$  is the number of candidate orbits and  $|P|$  is the number of candidate instruments. In our test problem, we have  $|O| = 5$  and  $|P| = 12$ , and therefore  $2^{60}$  total number of possible architectures. Each architecture can be represented by a Boolean matrix  $M$  where  $M(o, p) = 1$  if instrument  $p$  is assigned to orbit  $o$ , and  $M(o, p) = 0$  otherwise.

A total of 2,619 architectures were generated and displayed in the scatterplot of the GUI. For each architecture, two objective values, a cost metric and a science benefit metric, were calculated using a simulation tool previously developed in [37].

**Table 1.** Preset Candidate Driving Features

Name of the feature	Arguments	Description
Present	$I_i$	Instrument $I_i$ is present in at least one of the orbits
Absent	$I_i$	Instrument $I_i$ is absent in all the orbits
InOrbit	$O_i, I_j$	Instrument $I_j$ is present in orbit $O_i$
NotInOrbit	$O_i, I_j$	Instrument $I_j$ is not present in orbit $O_i$
Together	$I_i, I_j, (I_k)$	Instruments $I_i, I_j$ (and $I_k$ ) are present together in any orbit
TogetherInOrbit	$O_i, I_j, I_k, (I_l)$	Instruments $I_j, I_k$ (and $I_l$ ) are present together in orbit $O_i$
Separate emptyOrbit	$I_i, I_j, (I_k)$ $O_i$	Instruments $I_i, I_j$ (and $I_k$ ) are not present together in any single orbit No instrument is present in orbit $O_i$
numOrbitUsed	n	The number of orbits that have at least one instrument assigned is n

Based on this problem formulation, we pre-specified the structure of candidate driving features to be searched as shown in Table 1. This task is problem-specific, as the domain knowledge, problem structure and formulation can all be used in introducing high-level features.

### 5.5 Experiment Protocol

Before the start of the learning session, each participant is briefed on the context of the design tasks. Then they are provided with verbal explanations on how to use the analytic tools. This includes explanations on how to retrieve information from the scatterplot, how to use filters, how to set the target region in the objective space, and how to obtain driving features. After all the explanations are provided, the participants are given 40 minutes to interact with the given tool and to analyze the given designs. During this learning session, the participants were encouraged to take notes. After the learning session is finished, a problem set is given to each participant to solve. While solving problems, the participants were allowed to refer only to their own notes. There was no time limit for problem solving session.

### 5.6 Participants

There were total 12 participants in this study. The average age of all participants was 24.75 with the standard deviation of 3.67. There were 9 males and 3 females, 7 majoring in engineering, 2 in science, and 3 in business or economics. There were 4 undergraduate students, 7 graduate students, and 1 holding a doctorate degree. 7 participants had some previous experience in satellite system design, but this factor did not have any statistically significant influence on the results.

## 6. RESULTS

The results of the experiment in this paper are meant to be preliminary. There were only 3 participants for each group. Note that, because the number of participants was very small, statistical testing could not be used to reject null hypotheses

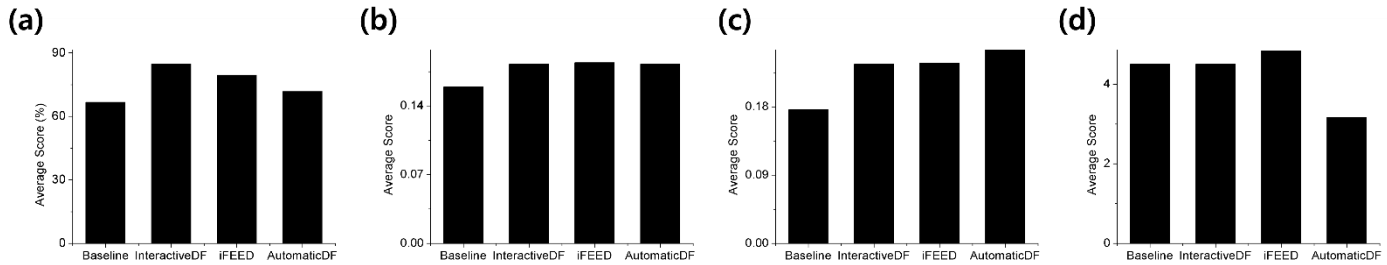
introduced in Section 5.1, and the error bars are not displayed in the figures.

Figure 4 summarizes the result for each group of participants. Figure 4a shows the average scores of each group of participants for multiple choice questions. The multiple choice questions test the participant’s factual knowledge. The figure shows that those who used interactive DF and iFEED outperformed the ones who used the baseline and automated DF. This is in agreement with the first hypothesis H1. However, the result is contradictory to H4, as the subjects who were provided with classification trees performed more poorly compared to the ones who were not given classification trees. The difference in the scores were not statistically significant.

Figure 4b and Figure 4c shows the average scores for two different design synthesis problems. The first one was to improve a given design by making a limited number of changes in the design variables, and the second one was to come up with a new design from the scratch. For both tasks, the subjects who were given only the basic tools (scatterplot and filters) performed poorly compared to other groups. However, in the task where the users were asked to create a new design, the automatic DF group performed better than the groups that used interactive methods. This result does not agree with the second hypothesis H2.

Figure 4d shows the level of trust felt by each group of subjects, on the analytic tools they have used. The group that used iFEED felt slightly more trust towards the tool compared to other groups. Interestingly, the subjects who used automated driving feature function felt the least trust, even less compared to those who used only the basic functionalities. This may indicate that when the user does not know how certain information is generated, he or she may feel less trust towards that information even when it is useful. This is in partial agreement with the third hypothesis, H3.





**Figure 4.** The result of the pilot experiment. (a) The average scores for multiple choice questions. (b) The average scores for design synthesis task, where the user was asked to improve a given design by modifying design variables. (c) The average scores for design synthesis task. The users are asked to come up with a new design. (d) The level of trust felt by each group on the tool they have used.

## 7. CONCLUSION

In this paper, we presented a new interactive knowledge discovery method called iFEED. It is aimed to help the human designer analyze a large dataset containing design alternatives with corresponding objective values and design variables. The method includes having the designer select a target region in the objective space, and mining driving features, which are the combinations of design variables that consistently lead a design into the target region. The driving features are mined using concepts from association rule mining. Then hierarchical structure is introduced to the driving features. Based on the performance and the level of compactness of classification trees, the user can devise and test new higher level driving features. This interactive process is expected to facilitate the designer in gaining high-level knowledge about the design problem. A preliminary experiment was conducted in an effort to test the effectiveness of iFEED.

## 8. LIMITATIONS AND FUTURE WORK

The findings of this paper is primarily limited by the small number of participants. In order to observe statistically meaningful results, the experiment has to be carried out with a significantly greater number of participants. Moreover, iFEED has only been tested on a specific system architecting problem containing only Boolean design variables. In principle, however, the method can be applied to any general complex engineering design problem with the mix of continuous, discrete, and categorical design variables.

For future work, we plan to apply the knowledge discovered using iFEED to support the design optimization process. The if-then format of driving features mined from association rules makes it easy to incorporate driving features in heuristic optimization. There have been efforts to incorporate domain-specific knowledge to make optimization process more efficient [9,38]. Some preliminary result from [17] shows promises in applying driving features for this purpose.

## REFERENCES

- [1] Bloebaum, C. L., and McGowan, A.-M. R., 2012, "The Design of Large-Scale Complex Engineered Systems: Present Challenges and Future Promise," AIAA Aviation Technology, Integration, and Operations (ATIO) Conference and 14th AIAA/ISSM, pp. 1–19.
- [2] Hazelrigg, G. a., 1998, "A Framework for Decision-Based Engineering Design," *J. Mech. Des.*, **120**(4), p. 653.
- [3] Ross, A., O'Neill, G., Hastings, D., and Rhodes, D., 2010, "Aligning Perspectives and Methods for Value-Driven Design," *AIAA Sp. 2010 Conf. Expo.*, (September), pp. 1–30.
- [4] Collopy, P. D., 2011, "Value-Driven Design," *J. Aircr.*, **48**(3), pp. 749–759.
- [5] Balling, R., 1999, "Design by shopping: A new paradigm?," *Proceedings of the Third World Congress of structural and multidisciplinary optimization (WCSMO-3)*, pp. 295–297.
- [6] Stump, G. M., Simpson, T. W., Yukish, M., and Hara, J. J., 2004, "Trade space exploration of satellite datasets using a design by shopping paradigm," *IEEE Aerosp. Conf. Proc.*, **6**, pp. 3885–3894.
- [7] Ross, A. M., Hastings, D. E., Warmkessel, J. M., and Diller, N. P., 2004, "Multi-Attribute Tradespace Exploration as Front End for Effective Space System Design," *J. Spacecr. Rockets*, **41**(1), pp. 20–28.
- [8] Hadka, D., and Reed, P., 2012, "Borg: An Auto-Adaptive Many-Objective Evolutionary Computing Framework," *Evol. Comput.*, **21**(2), pp. 1–30.
- [9] Hitomi, N., and Selva, D., 2015, "The effect of credit definition and aggregation strategies on multi-objective hyper-heuristics," *Proceedings of the ASME 2015 International Design Engineering Technical Conferences & Computers and Information in Engineering Conferences*, Boston, MA, pp. 1–10.
- [10] Kanukolanu, D., and Winer, E. H., 2006, "A Multidimensional Visualization Interface to Aid in Trade-off Decisions During the Solution of Coupled Subsystems Under Uncertainty," *J. Comput. Inf. Sci. Eng.*, **6**(September 2006), pp. 288–299.
- [11] Eddy, J., and Lewis, K. E., 2002, "Visualization of Multidimensional Design and Optimization using Cloud Visualization," *Proc. DETC'02*, pp. 899–908.
- [12] Stump, G. M., Yukish, M., Simpson, T. W., and Harris, E. N., 2003, "Design space visualization and its

- application to a design by shopping paradigm,” Proceedings of the ASME 2003 Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Chicago, IL, pp. 1–10.
- [13] Stump, G. M., Yukish, M., Martin, J., and Simpson, T., 2004, “The ARL Trade Space Visualizer: An Engineering Decision-Making Tool,” 10th AIAA/ISSMO Multidiscip. Anal. Optim. Conf., (September).
- [14] Stump, G. M., Yukish, M., and Merenich, J. J., 2005, “Tracing interesting features in trade spaces to conceptual model design rules,” IEEE Aerosp. Conf. Proc., **2005**.
- [15] Agrawal, G., Bloebaum, C. L., and Lewis, K. E., 2005, “Intuitive Design Selection Using Visualized n-Dimensional Pareto Frontier,” 46th AIAA/ASME/ASCE/AHS/ASC Struct. Struct. Dyn. Mater. Conf. AIAA 2005-1813, (April), pp. 1–14.
- [16] Chiu, P.-W., and Bloebaum, C., 2008, “Hyper-Radial Visualization (HRV) for Decision-Making in Multi-Objective Optimization,” 46th AIAA Aerosp. Sci. Meet. Exhib. AIAA 2008-907, (January), pp. 1–16.
- [17] Selva, D., Abello, C., and Hitomi, N., 2015, “Preliminary Experiments with Learning Agents in an Interactive Multi-agent Systems Architecture Tradespace Exploration Tool Preliminary Experiments with Learning Agents in an Interactive Multi-agent Systems Architecture Tradespace Exploration Tool,” Systems Conference (SysCon), 2015 9th Annual IEEE International., IEEE, pp. 445–452.
- [18] Agrawal, R., Imielinski, T., and Swami, A., 1993, “Mining Association in Large Databases,” IBM Almaden Res. Cent., pp. 207–216.
- [19] Han, J., Pei, J., Yin, Y., and Mao, R., 2004, “Mining Frequent Patterns without Candidate Generation: A Frequent-Pattern Tree Approach,” Data Min. Knowl. Discov., **8**(1), pp. 53–87.
- [20] Peng, H. C., Long, F. H., and Ding, C., 2005, “Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy,” IEEE Trans. Pattern Anal. Mach. Intell., **27**(8), pp. 1226–1238.
- [21] Coello Coello, C. A., Van Veldhuizen, D. A., and Lamont, G. B., 2002, Evolutionary algorithms for solving multi-objective problems.
- [22] Keim, D., Kohlhammer, J., Ellis, G., and Mansmann, F., 2010, Mastering the Information Age Solving Problems with Visual Analytics, Eurographics Association.
- [23] Winer, E. H., and Bloebaum, C. L., 2002, “Development of visual design steering as an aid in large-scale multidisciplinary design optimization. Part I: method development,” Struct. Multidiscip. Optim., **23**(6), pp. 412–424.
- [24] Winer, E. H., and Bloebaum, C. L., 2002, “Development of visual design steering as an aid in large-scale multidisciplinary design optimization. Part II: method validation,” Struct. Multidiscip. Optim., **23**(6), pp. 425–435.
- [25] Stump, G., Lego, S., Yukish, M., Simpson, T. W., and Donndelinger, J. a., 2009, “Visual Steering Commands for Trade Space Exploration: User-Guided Sampling With Example,” J. Comput. Inf. Sci. Eng., **9**(4), p. 044501.
- [26] Chiu, P., and Bloebaum, C., 2009, “Visual Steering for Design Generation in Multi-objective Optimization Problems,” 47th AIAA Aerosp. Sci. Meet., (January), pp. 1–14.
- [27] Shneiderman, B., 2002, “Inventing Discovery Tools: Combining Information Visualization with Data Mining,” Inf. Vis., **1**(1), pp. 5–12.
- [28] Puolamäki, K., Papapetrou, P., and Lijffijt, J., 2010, “Visually controllable data mining methods,” Proc. - IEEE Int. Conf. Data Mining, ICDM, pp. 409–417.
- [29] Bertini, E., and Lalanne, D., 2010, “Investigating and reflecting on the integration of automatic data analysis and visualization in knowledge discovery,” ACM SIGKDD Explor. Newsl., **11**(2), p. 9.
- [30] (Luke) Zhang, X., Simpson, T., Frecker, M., and Lesieutre, G., 2012, “Supporting knowledge exploration and discovery in multi-dimensional data with interactive multiscale visualisation,” J. Eng. Des., **23**(1), pp. 23–47.
- [31] Yan, X., Qiao, M., Li, J., Simpson, T. W., Stump, G. M., and Zhang, X. (Luke), 2012, “A Work-Centered Visual Analytics Model to Support Engineering Design with Interactive Visualization and Data-Mining,” 2012 45th Hawaii International Conference on System Sciences, pp. 1845–1854.
- [32] Newell, A., and Simon, H. A., 1972, Human problem solving., Englewood Cliffs, NJ: Prentice-Hall.
- [33] Liu, B., Hsu, W., Ma, Y., and Ma, B., 1998, “Integrating Classification and Association Rule Mining,” Knowl. Discov. Data Min., pp. 80–86.
- [34] Brin, S., Motwani, R., Ullman, J. D., and Tsur, S., 1997, “Dynamic itemset counting and implication rules for market basket data,” ACM SIGMOD Rec., **26**(2), pp. 255–264.
- [35] Quinlan, J. R., 1993, C4. 5: programs for machine learning, Elsevier.
- [36] Selva, D., 2014, “Knowledge-intensive global optimization of Earth observing system architectures: a climate-centric case study,” SPIE Remote Sens., **9241**, p. 92411S–92411S.
- [37] Selva, D., and Crawley, E. F., 2013, “VASSAR: Value assessment of system architectures using rules,” IEEE Aerosp. Conf. Proc.
- [38] Selva, D., 2014, “Experiments in knowledge-intensive system architecting: Interactive architecture optimization,” IEEE Aerosp. Conf. Proc.