INCORPORATING USER DESIGN PREFERENCES INTO MULTI-OBJECTIVE ROOF TRUSS OPTIMIZATION

A Dissertation

by

BREANNA MICHELLE WEIR BAILEY

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

May 2006

Major Subject: Civil Engineering

INCORPORATING USER DESIGN PREFERENCES INTO MULTI-OBJECTIVE ROOF TRUSS OPTIMIZATION

A Dissertation

by

BREANNA MICHELLE WEIR BAILEY

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Approved by:

Chair of Committee,	Anne Raich
Committee Members,	Gary Fry
	Harry Jones
	Make McDermott, Jr.
Head of Department,	David Rosowsky

May 2006

Major Subject: Civil Engineering

ABSTRACT

Incorporating User Design Preferences into

Multi-Objective Roof Truss Optimization. (May 2006)

Breanna Michelle Weir Bailey, B.S., Texas A&M University;

M.S., University of Illinois Urbana-Champaign

Chair of Advisory Committee: Dr. Anne Raich

Automated systems for large-span roof truss optimization provide engineers with the flexibility to consider multiple alternatives during conceptual design. This investigation extends previous work on multi-objective roof truss optimization to include the design preferences of a human user. The incorporation of user preferences into the optimization process required creation of a mechanism to identify and model preferences as well as discovery of an appropriate location within the algorithm for preference application.

The first stage of this investigation developed a characteristic feature vector to describe the physical appearance of an individual truss. The feature vector translates visual elements of a truss into quantifiable properties transparent to the computer algorithm. The nine elements in the feature vector were selected from an assortment of geometrical and behavioral factors and describe truss simplicity, general shape, and chord shape.

Using individual feature vectors, a truss population may be divided into groups of similar design. Partitioning the population simplifies the feedback process by

allowing users to identify groups that best suit their design preferences. Several unsupervised clustering mechanisms were evaluated for their ability to generate truss classifications that matched human judgment and minimized intra-group deviation. A one-dimensional Kohonen self-organizing map was selected.

The characteristic feature vectors of truss designs within user-selected groups provided a basis for determining whether or not a user would like a new design. After analyzing user inputs, prediction algorithm trials sought to reproduce these inputs and apply them to the prediction of acceptable designs. This investigation developed a hybrid method combining rough set reduct techniques and a back-propagation neural network.

This hybrid prediction mechanism was embedded into the operations of an Implicit Redundant Representation Genetic Algorithm. Locations within the ranking and selection processes of this algorithm formed the basis of a study to investigate the effect of user preference on truss optimization.

Final results for this investigation prove that incorporating a user's aesthetic design preferences into the optimization project generates more design alternatives for the user to examine; that these alternatives are more in line with a user's conceptual perception of the project; and that these alternatives remain structurally optimal.

ACKNOWLEDGEMENTS

I would like to thank Dr. Anne Raich for her guidance and support throughout this process. I am especially grateful for the freedom she gave me to explore this topic while helping me to maintain focus and overcome difficulties.

I am also grateful to Dr. Make McDermott, Jr., Dr. Harry Jones, and Dr. Gary Fry for serving on my committee and for their comments and suggestions to improve my research. I particularly want to thank Dr. Fry for his long-standing mentorship and the many years of advice and encouragement.

Additionally, I wish to acknowledge the contributions of my professors, fellow graduate students, friends and family who completed surveys as part of the external verification of my research. Thank you for taking the time to offer your opinions.

Finally, I would like to thank my parents, who taught me the value of education and hard work, and my husband, Steven, whose patience and confidence in my abilities kept me going even when I doubted myself.

TABLE OF CONTENTS

CHAPTER Page

LIST OF FIGURES

FIGURE Page

FIGURE Page

FIGURE Page

FIGURE Page

FIGURE Page

FIGURE Page

FIGURE Page

LIST OF TABLES

TABLE Page

CHAPTER I

INTRODUCTION TO THE CONCEPTUAL DESIGN OF LARGE SPAN ROOF TRUSSES

The truss is one of the most basic, yet powerful, structures in the civil engineer's portfolio. First developed by the Romans, wooden trusses were modernized by A. Pallidio in the 16th century (West and Geschwindner 2002) and quickly found their way into roofing applications. Architects introduced the earliest designs for trussed roofs in order to create new architectural forms (Yeomans 1992). However, as knowledge of this structural device spread, carpenters took on the burden of design (Yeomans 1992), reflecting an early separation between architectural conception and structural realization.

With the advent of steel, modern roof trusses became useful in a variety of applications. Roof trusses are often selected for structural systems with large-span requirements, such as industrial warehouses, airplane hangers, and athletic stadiums (Hibbeler 2006). These long spans create large, column-free spaces that can improve visibility and storage flexibility. Trusses meet these needs by using slender members to convert applied bending loads into tensile and compressive forces, thus reducing the amount of material a beam might require for the same span (Hibbeler 2006). In fact, the use of trusses for roofing systems has become so ubiquitous that the most common truss configurations are referred to by the names of their designers (Howe, Pratt, etc.) or dominant features (bowstring, sawtooth, etc.) (Hibbeler 2006).

This dissertation follows the style of the *Journal of Structural Engineering.*

Trusses are often selected for roofing applications because they offer significant material and cost savings over beamed roofs. In this setting, where extremely efficient designs are desirable for economic feasibility, it seems logical that a designer would further refine this advantage by creating the highest performing structural truss system possible. To this end, a method for optimizing roof trusses in an efficient, structurally rigorous manner is highly desirable. Such an optimization method would not only ensure financially beneficial designs but would also reduce the engineer's design burden, especially if the method could incorporate the larger issue of conceptual design.

In generating initial structural designs, engineers may examine a limited number of potential configurations, with other design options "...implicitly considered and rejected" (Sisk et al. 2003). "At the conceptual stage of the design process, there is usually very little time to consider all possible (feasible) alternatives before decisions have to be made and resources committed" (Rafiq et al. 2003). Often, these same time restrictions will prevent designers from exploring alternative design solutions unless a problem arises with the initial proposal (Sisk et al. 2003). For roof truss applications, the most fundamental decision made in the conceptual design stage will concern the selection of a suitable truss topology. Given the wide array of potential truss configurations available, a tool that would allow designers to simultaneously consider multiple truss topologies could potentially lead to more optimal structural systems.

One tool that is particularly well suited to this type of optimization is the genetic algorithm (GA) because it is able to rapidly generate and evaluate many design alternatives. The GA is a search algorithm based on the theory of natural selection and

survival-of-the-fittest (Goldberg 1989). GAs consider multiple solutions to a problem and use "payoff" information, i.e., satisfaction of the objective function, to determine which of these potential solutions is optimal (Goldberg 1989). For example, a simple GA might determine optimal member sizing for a predefined truss topology; the potential solutions in this case would consist of a string of numbers that correlated to area selections for each member. For the more sophisticated roof truss application considered here, the Implicit Redundant Representation Genetic Algorithm (IRR GA) provides additional flexibility and the ability for solutions to self organize (Raich and Ghaboussi 1997). The IRR GA, which will be discussed further in the next section, is a mechanism that allows for optimization of *all* aspects of a structural design, including both the geometric sizing suggested above as well as topological considerations (Raich and Ghaboussi 2000). Diverse design alternatives develop as an integral part of the IRR GA's evolution towards optimal structural solutions (Raich and Ghaboussi 2000).

However, creating a tool for the conceptual design of large span roof trusses requires something more than the ability to consider multiple topologies. An optimization program run independently of the designer will find solutions that perform well from a structural standpoint, but these solutions may or may not satisfy larger design considerations. Constraints such as conformance to the architectural program, constructability, and overall economic feasibility are often readily apparent based upon visual inspection of the design alternatives. Formulating these constraints in a way the GA can understand is mathematically intractable, yet human designers are exceptionally good at recognizing them. This observation reflects the basic fact that although

"...digital computers are now a million times faster than the original computers, there are still several problems in which humans can outperform computers. These include pattern recognition, fault tolerance, intuitive reasoning, and so on" (Ramasamy and Rajasekaran 1996).

Teaching GAs to recognize these "intuitive" constraints would establish their efficacy for use in conceptual design. Not only would GAs be able to aid engineers by generating and identifying optimal structural solutions, but they would also have the ability to autonomously assess the practicality of these solutions. Moreover, if feedback from a designer could simultaneously provide information about practicality and aesthetic desirability, the GA could use this information to perform a user-guided exploration of the search space. This exploration could encourage the generation of new, previously unexplored truss topologies that more closely align with a user's preferences, thus increasing the GA's ability to reunite the sometimes-conflicting goals of architectural conception and structural realization.

The goal of the present research is to design such a system for incorporating a user's aesthetic design preferences into the multi-objective optimization of large span roof trusses.

Multi-Objective Optimization of Large Span Roof Trusses

The design of large span roof trusses must account for three key elements: truss topology, nodal geometry, and member sizing. Truss topology refers to the number and placement of members and nodes defining a given truss configuration. As stated above, it is desirable for a conceptual design tool to consider multiple truss topologies in the

search for a truly global optimal design. To provide even more variety in the creation of truss alternatives, the joint locations within a given topology are not considered to be static. That is, during optimization of the nodal geometry, truss joints may be perturbed so as to alter the height, length, or inclination angle of a truss member. The selection of appropriate member sizes completes the truss optimization process. The current application is limited to steel truss designs, so member selection draws on a discrete set of rolled W-shapes to provide design alternatives. W-shapes were selected for this application since they are more frequently used in long-span trusses than double angles or channels.

The ability to encompass topology, geometry, and sizing represents a holistic approach to truss optimization made possible, in part, because of the flexibility of the IRR GA. As with simple genetic algorithms, the IRR GA operates on the principles of selection, crossover, and mutation. These operations are described in Goldberg (1989) and will be briefly summarized here.

Each generation of the GA creates a population of individuals. These individuals, or "chromosomes," take the form of number strings, generally binary in nature. A chromosome represents a potential solution to the optimization problem-- in this case, a potential truss design. An individual's fitness is evaluated according to how well it satisfies the problem objectives, and the relative fitness of each individual guides the selection process. During selection, above-average solutions in the current generation are chosen, through one of several processes involving a random component, to become parents of the next generation. When two parent solutions are selected for

crossover, a point within the chromosome will be randomly chosen, and the "head" of one parent will be attached to the "tail" of the second and vice versa, as illustrated in Figure 1.1. This process forms two children solutions, shown on the right of Figure 1.1, that combine attributes of both parents, which are shown on the left (Goldberg 1989).

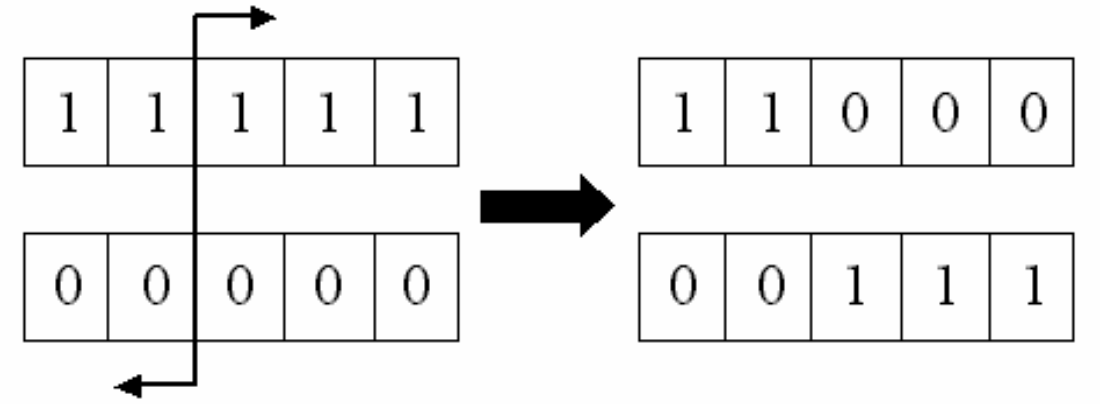


Figure 1.1. Schematic representation of the crossover operator.

Lastly, some of the solutions in the new generation will undergo mutation. Once again, a location within the chromosome string is selected at random, and the numerical value of the solution is changed at that point. For a binary representation, a "1" would change to a "0" or vice versa (Goldberg 1989). This process is illustrated in Figure 1.2.

Figure 1.2. Schematic representation of the mutation operator.

Individuals in succeeding generations of the GA are created through repetition of the selection, crossover, and mutation processes until the desired number of generations are complete. Selection and crossover are exploitive in nature, in that they seek to capitalize on highly performing solutions previously discovered. Mutation, on the other hand, is purely explorative and seeks to enhance the diversity of each generation by discovering new solutions. By combining two good solutions in the hopes of finding even better solutions, crossover also serves an explorative purpose. The GA balances these dual goals of exploration and exploitation by performing crossover and mutation on a percentage of the population, with mutation percentages generally smaller than crossover (Goldberg1989).

The IRR GA is unique not because of the way these operations are performed but because of the chromosome on which they are performed. According to the description presented in Raich and Ghaboussi (1997), an IRR GA solution chromosome contains genes, i.e., sequences of design variables, separated by redundant bits, as illustrated in Figure 1.3. A specific number sequence, such as "111", serves as a gene locator to flag the presence of each gene while the chromosome is being parsed. When a gene locator is encountered, the following bits will contain information about the current design. For a truss application, this information might be the proposed member area and the coordinates of the start and end nodes for that member. Once this information is read, the redundant section of the chromosome is ignored until another gene locator is found (Raich and Ghaboussi 1997).

The redundant sections of the chromosome provide locations for new gene

sequences to develop. In structural design applications, this process is equivalent to adding a member to a given topology. The processes of crossover and mutation may also interrupt an existing gene sequence, which effectively removes a member from the topology. Thus, the IRR GA encoding provides flexibility for topological forms to change dynamically over time, and it does so in an unconstrained search space without the use of heuristic information or pre-defined topologies (Raich and Ghaboussi 2000).

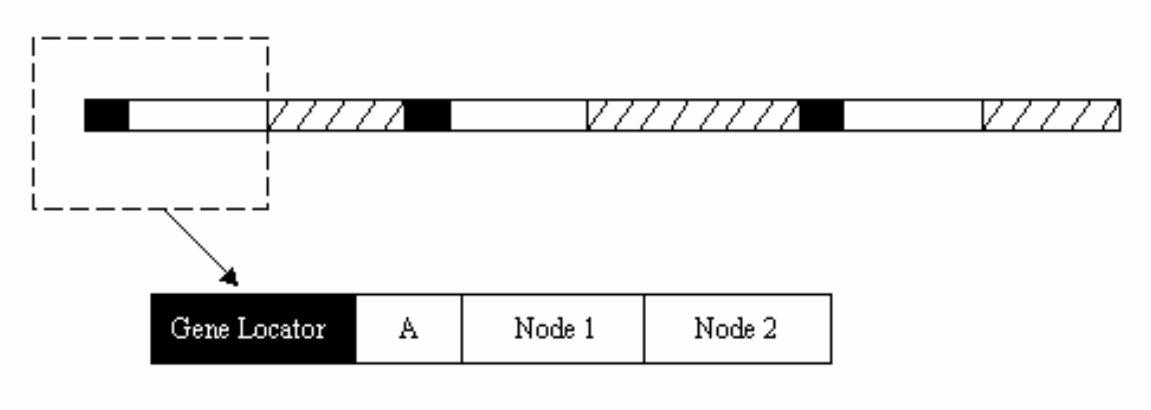


Figure 1.3. Schematic of the IRR GA chromosome with a sample truss encoding.

The IRR GA is therefore a very powerful tool for conceptual design. It can accommodate variations in truss topology, geometry, and member sizing. It also has the flexibility to generate new and unusual trusses, thus providing the designer with alternatives that might not otherwise have been considered. The ability to produce a robust body of design alternatives is particularly important in this application, since the goal of the present roof truss optimization software is not to generate a single, good truss design but to offer the user a wide range of equally optimal designs from which to

choose.

In achieving this goal, an appropriate definition of "optimal" must be selected. Minimization of both the total truss weight and the maximum truss deflection are defined as primary objectives. Since these objectives conflict with each other, a trade-off, or Pareto, curve of optimal trusses will form. As shown in Figure 1.4, trusses along this curve will represent the "best" truss when different ratios of weight and deflection are used, so that trusses at one extreme of the curve will be light weight but very flexible, and other trusses will be much heavier but more rigid. The goal is to develop this curve sufficiently that many, equally viable truss alternatives are presented to the designer.

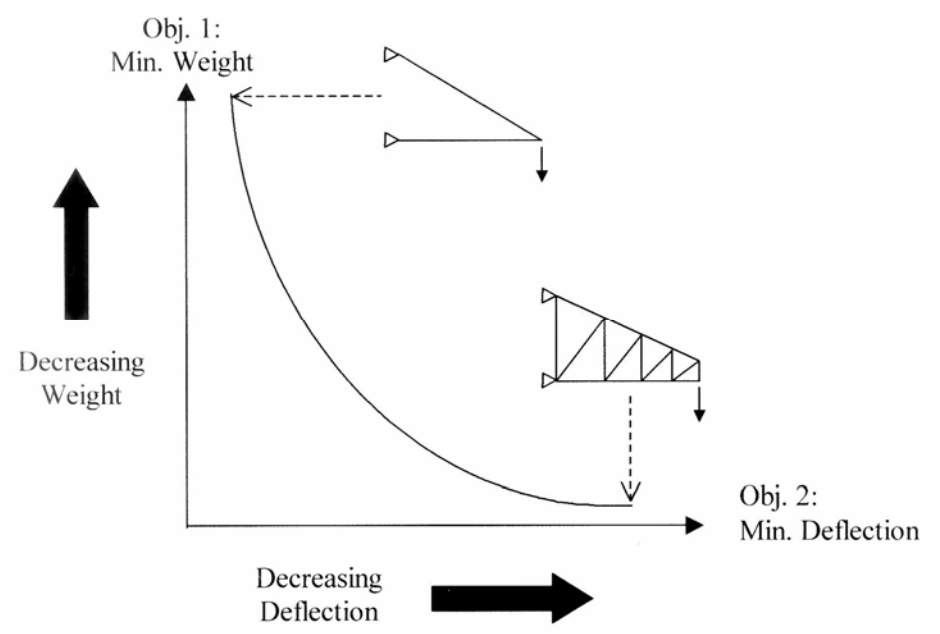


Figure 1.4. Example trade-off curve for minimizing truss weight and deflection.

Three additional constraints ensure that trusses generated by the IRR GA are practically feasible. First, member stresses are monitored to prevent violations of a member's buckling or tensile strength. For constructability reasons, all member lengths are limited to no less than eight and no more than twenty feet. Violations of kinematic stability are also penalized. Since this is a multi-objective formulation, each of these constraints is treated as a separate objective. Further details of the truss optimization project and multi-objective optimization are discussed in Chapter V.

Research Objectives and Scope

To the existing definition of truss optimality presented in the previous section, the present work adds one further objective: satisfaction of user design preferences. It is believed that the incorporation of the human user into the automated design process will improve the quality of the design alternatives in terms of both practicality and desirability. The current state-of-the-art relies upon the IRR GA to evaluate thousands of possible design alternatives based solely on the structural criteria outlined above, regardless of whether these alternatives are appropriate to the conceptual design of a given system. If a human user's perceptions of and goals for a design could be understood by the IRR GA, then a more tailored exploration of the search-space would be possible. This user-guided exploration would enrich the conceptual design process by centering on alternatives that are truly of interest to the designer.

In order to successfully integrate the user into the optimization of large span roof trusses using the IRR GA algorithm, four key tasks were identified. These tasks form the major objectives of this research effort and include the following:

- **Feature Identification:** What characteristic features physically distinguish individual trusses, and how may these features be quantified?
- **Classification Mechanism:** What classification mechanism(s) may be used to perceive similarities and differences between the characteristic features of a truss population?
- **Preference Detection:** How may user evaluations of classified truss populations form the basis for the extraction and prediction of aesthetic preferences?
- **Preference Implementation:** Once extracted, how may user preferences guide the IRR GA towards more pleasing design alternatives?

The first objective asks fundamental questions about perception of form. Humans have the ability to visually distinguish between trusses. These distinctions are made rapidly and efficiently and form the basis for judgments about the aesthetic appeal or practicality of a given design. The first step in making these judgments available to the genetic algorithm is to form a discretized view of each truss. A vector of characteristic features that fully describe the truss in terms of human perception will represent each individual truss design. This feature vector will serve as input for the classification and preference detection mechanisms and represents the most fundamental "picture" of the truss available to the computer program. In creating such a feature vector, it is therefore important to determine what features most strongly impact truss appearance and to effectively formulate mathematical measures of these features.

Once a methodology for describing individual trusses has been developed, it becomes possible to consider how trusses compare to each other. In this second stage,

the goal is to develop a classification scheme for grouping trusses that are "alike", that is, trusses with similarities in their characteristic feature vectors. Grouping trusses according to their similarities simplifies the amount of information that must be provided by the user. Users are asked to evaluate groups of similar designs rather than consider each generated design on an individual basis, a task that could be overwhelming to a designer presented with 100 unique truss topologies. Moreover, a truss classification scheme leads to a more robust pool of information from which to extract user preferences. If a user were asked to examine 100 trusses, yet identified only one truss as a preferred form, very little information would be gained about the user's likes and dislikes. However, asking a user to make judgments about the desirability of different truss groups, each exhibiting different characteristics, provides a better, and more robust, indicator of the overall design attributes a user prefers.

An efficient classification mechanism, therefore, will provide an important basis for extracting a user's design preferences. A user's evaluations of these truss groups will provide the necessary information to determine what particular designs a user likes or dislikes. In order to be effective in a design context, however, this information must be translated into a prediction of what a user will like in the future. The IRR GA will generate hundreds of design alternatives in each generation of the main and sub-processes, and directly querying a user for feedback on these new designs is impractical. It is therefore necessary to have a preference detection mechanism in place that can accurately predict whether or not a previously unexamined truss topology will coincide with the user's overall design concept. Developing such a mechanism is the crucial step

in teaching a genetic algorithm to autonomously evaluate the search space.

This research effort combines the three proceeding objectives to formulate a method for capturing a user's aesthetic design preferences. The method with which these preferences are incorporated into the optimization process also requires careful investigation. By implementing design preferences as an additional objective in the optimization process, the IRR GA should become more responsive to the needs of its user. Alternatives that closely align with a user's overall design concept should be encouraged to propagate through future generations. However, the preference objective must not outweigh the other objectives and constraints that define the optimization problem as a whole. That is, the trusses proposed by the IRR GA as good design alternatives should still provide structurally optimal, or near-optimal, solutions.

An optimization algorithm based solely on questions of truss weight and deflection would prove useful to a designer, but how much more useful would that same algorithm be if it could be made responsive to its user? Incorporating user preferences into the optimization software will allow human intuition to guide the design process, thus generating more reliably "good" truss forms while maintaining the genetic algorithm's ability to find new and unusual topologies. This goal may be accomplished by defining the characteristic features that best describe a truss, appropriately grouping similar trusses based on these features, converting a designer's group evaluations into appropriate design criteria, and applying these criteria to the optimization process. Chapters II through V will discuss each of these objectives in detail.

Review of Related Literature

Truss optimization is not a new idea; a large body of previous research attempts to provide solutions to the questions of optimal member sizing, geometry, or topology. Much of the research over the past fifteen years has focused on the use of genetic algorithms. Nor is the concept of aesthetically evaluating a product during the design process unexplored. The perception that engineered products should become more responsive to customers' aesthetic preferences has led to growing interest in this area. However, the current research effort is aimed at combining both of these ideas. The desired synthesis will lead to an optimization process that considers all aspects of truss design; this holistic process is believed to be unique.

Member optimization for a given truss design has been well studied (Cheng and Li 1997; Camp et al. 1998), and there is little doubt that a GA performs well in this simple application. However, researchers recognized fairly early on that "…the selection of an optimal topology is arguably among the most difficult structural optimization problems" (Hajela and Lee 1995) and attempted to use GAs in these applications. Hajela and Lee (1995) performed topology optimization in two stages. The first identified kinematically stable truss configurations; in the second phase, addition and removal of members was used to find an optimal truss weight. These researchers used a ground structure definition to consider topological optimization of cantilever and bridge trusses (Hajela and Lee 1995). Popular in topological optimization, ground structures define the most complex structure the researchers wish to consider, and the GA or other optimization tool will determine which members to

include in the final configuration. In truss optimization research, defining a ground structure may mean defining all possible nodal coordinates (as in Hajela and Lee's case) or defining all possible members.

The need to incorporate more than one facet of truss design into the optimization process has also been well recognized. "To make the GA solution attractive as a design methodology… would require that sizing, shape, and topology aspects of structural optimization be addressed simultaneously" (Rajan 1995). Grierson and Pak (1993) took the first step towards this goal. They made use of a ground structure and variations in member length and size in order to minimize the weight of simple frame structures (Grierson and Pak 1993). Studies applying topological, geometrical, and sizing optimization to trusses frequently use ground structures, as well, with weight being the primary objective and constraints placed on displacement, member stresses, and buckling (Rajan 1995; Rajeev and Krishnamoorthy 1997).

Ruy et al (2001) attempted to expand this earlier work into a multi-objective setting that more closely reflected the engineering design process. Using a ground structure base and keeping member areas constant, optimal topologies were determined using a modified structured genetic algorithm. Geometric and member sizing variations were then considered for each identified topology. This work focused on minimizing both weight and maximum deflection (Ruy et al 2001).

Ground structure approaches tend to limit the design search space and strongly influence the final truss design (Rajan 1995). The current work aims to create new and unusual designs, which dictates as unconstrained a search space as reasonably possible.

The IRR GA approach has proven effective in this regard for the optimization of building frames (Raich and Ghaboussi 2000). There appears to be a gap in the literature for a similar consideration of trusses. Moreover, although proof-of-concept work has been performed for trusses in multi-objective settings (Camp et al 1998; Ruy et al 2001), this topic also needs to be more fully explored.

Of course, in searching for new and interesting designs, the genetic algorithm needs to understand what the user considers "interesting." This optimization aspect is where the question of truss aesthetics comes into play. At present, there does not appear to be any other work where the aesthetic qualities of a truss are used in the optimization process, although it is well recognized that a structure's appearance may be integral to its purpose and should be considered in its design (Reekie 1972; Kulasuriya et al. 2002). In the structural arena, perception of good design is often correlated to proportion, simple geometrical forms, and repetition (Reekie 1972). In one instance, researchers asked a group of civil engineers to consider parallel-chorded trusses with differing base to height ratios in an attempt to study the effect of proportion on aesthetic preference. Although results for trusses were less conclusive than for other geometries, they concluded that "... there may be some correspondence between aesthetic sense and optimal solutions" (Kulasuriya et al. 2002).

In recent years, researchers in varied fields have made a concerted effort to bring the question of product aesthetics into the design process. While this effort may be as simple as manually iterating design in response to a series of guided questions about aesthetic appeal (Wagner 2001), others focus on automated methods for incorporating

user preferences. In one such study, GAs are used for the conceptual design of bridges, and the fitness function relies solely on the aesthetic preferences of the designer (Furuta et. al. 1995). This fitness function comprises terms relating to the formative beauty, balance, and slenderness of the truss, and the user makes a-priori weight assignments to reflect the relative importance of these ideals (Furuta et. al. 1995).

Many other studies incorporate user preference in a more interactive way using evolutionary computing methods. A good summary of methods and applications is provided in Takagi (1998). Previous applications include computer graphics (Sims 1993; Graf and Banzhaf 1995) and automobile design (Petiot and Grognet 2002; Yanagisawa and Fukada 2004). In some cases, a CAD, or CAD-like, environment is used to create a range of products, from simple geometric shapes like cups or vases, to more complex entities, such as automobiles (Adelson 1998; Smyth and Wallace 2000).

Despite their differences in scope and application, these research programs share the goal of embedding "KANSEI" into the design process. KANSEI is a Japanese term used to summarize "...the total concept of intuition, preference, subjectivity, sensation, perception, cognition, and other psychological processing functions" (Takagi 1998). The interactive GA is a popular method of accomplishing this objective. Interactive GAs replace the mathematical fitness function with human judgment (Takagi 1998). In most cases, this means users directly select the phenotypes to become parents of the next generation (Sims 1993; Smyth and Wallace 2000; Graf and Banzhaf 1995). The evolution is considered complete when the user is satisfied with the current design.

Other methods make an attempt to learn a user's design preferences and therefore

speed convergence towards an acceptable solution. One researcher has created an expert system with rule-values based upon geometric factors; new products are designed in accordance with the current rules, which are updated based upon the user's evaluation of the product (Adelson 1998). Another approach uses rough set theory to draw comparisons between designs rated "good" or "bad" (Yanagisawa and Fukada 2003; Yanagisawa and Fukada 2004; Yanagisawa and Fukada 2005). Features that appear to be favored by the user are held constant while others are allowed to vary at random (Yanagisawa and Fukada 2003) or adapted using a multi-objective GA (Yanagisawa and Fukada 2004). This process is shown to converge to a solution more quickly than a simple interactive GA and has the added advantage that preferred features are not obscured by crossover (Yanagisawa and Fukada 2005).

Additionally, neural networks have been used to estimate a user's design preference. One group of researchers trained a neural network to predict users' evaluations of the aesthetic appeal of concrete retaining walls, thus creating a tool to capture these subjective judgments for later use (Chikata et al 1998). In some instances, neural networks have been used only in order to reduce the burden on the human evaluator, which is considered to be the biggest barrier to the implementation of interactive computational systems (Ohsaki and Takagi 1998). Both Euclidean distances and a trained neural network were used to compare similarities between facial expressions and sort them according to predicted user preferences, thus making the evaluation process simpler for a user (Ohsaki and Takagi 1998; Takagi and Ohya 1996).

On a larger scale, efforts have also been directed towards developing conceptual

design tools for structural applications. A structured GA selected the shape of curves defining a concrete arch dam, with optimization focused on minimizing the total volume (Parmee 1993). Although not allowing for direct user evaluation of the proposed shapes, researchers made an effort to include the designer in this process by allowing users to extract preferred designs, make modifications to them outside of the GA, and then use them to "seed" future GA solutions (Parmee 1993).

Conceptual design tools for building layouts have also used structured GAs to search for design alternatives (Sisk et al 2003; Rafiq et al 2003). The goal here has been to "…support rather than replace the designer" (Sisk et al 2003). In one of these studies, users were allowed to select fitness function weights or to directly manipulate solutions (Sisk et al 2003). Other researchers focused on providing users with information about the GA's search progress and how different selections (material, geometry, etc.) would affect the overall design (Rafiq et al 2003). The aim of this was to add "…transparency to the otherwise black box operation of a GA" (Rafiq et al 2003).

Organization of the Dissertation

The remaining chapters of this dissertation discuss each of the research objectives in turn. Chapter II presents the feature identification investigation, which develops a method for describing individual trusses. Chapter III considers different classification mechanisms for perceiving similarities between trusses. In Chapter IV, preference detection methods are explored for their ability to capture and predict a user's aesthetic design preferences. Chapter V combines these efforts into an implementation of the preference criterion in the IRR GA. Results are summarized in Chapter VI.

CHAPTER II

FEATURE IDENTIFICATION

When an engineer looks at a truss, what characteristics distinguish it from other trusses common to his or her experience? The number of members in a truss, its height, and the shape of its top or bottom chord are all static quantities, yet different observers may place different levels of importance on one or the other of these quantities depending on the overall design envisioned for a particular project. In order to capture these types of judgments, a characteristic feature vector is developed as part of the current research effort. This chapter discusses the process applied and results obtained for this task. The feature vector defined here will contain the information most relevant to the physical description of a truss and serve as input for the classification and preference detection methods discussed in Chapters III and IV.

In determining the characteristics that have the most significant visual impact on a truss's appearance, a total of twenty potential features were explored. Eleven of these features were originally proposed in a proof-of-concept experiment to determine if trusses could be successfully grouped by appearance using a one-dimensional Kohonen self-organizing map (Wong and Sneed 2004). These eleven characteristics were expanded, and in some cases redefined, in the current work. The characteristics proposed for inclusion in the feature vector varied from simple, geometric properties to more abstract measurements intended to capture truss behavior. They include the following:

- Maximum truss height
- Maximum joint connectivity
- Average joint connectivity
- Number of top chord nodes (joints)
- Total length of all truss members
- Ratio of compression to tension members
- Mid-span clearance
- Number of joints
- Number of members
- Top chord concavity
- Presence of top chord dip(s)
- Top chord flatness (multiple measures)
- Number of bottom chord nodes (joints)
- Ratio of top to bottom chord nodes (joints)
- Bottom chord flatness
- Top chord angularity
- Bottom chord angularity
- Number of top chord direction changes
- Number of bottom chord direction changes
- Truss depth

Methodology for Selecting Effective Features

The following section will describe the procedures used to determine which of

the twenty truss characteristics proposed above should be included in the final form of the truss feature vector. In order to judge how effective these characteristics were in physically distinguishing trusses, a series of trials were performed using several different predefined truss populations. These trials isolated each proposed truss characteristic and attempted to create feature maps, that is, groupings of similar trusses, based only on the given characteristic. Both visual and numerical evaluation criteria were used to determine whether or not the selected characteristic was successful as a truss descriptor.

Test Populations

Each generation of the IRR GA creates as many as 300 new truss topologies. These trusses arise through the random processes of the genetic algorithm, and the topology of an individual design may vary substantially from other topologies in the population. The population as a whole will also differ from previously explored populations, especially if user preferences are used to guide the GA's exploration.

Therefore, in creating a feature vector, it was important to isolate truss characteristics that were invariant to population changes. To this effect, a series of test populations were created for the feature investigations. These test populations varied both in size and complexity and were created using the topology generator described in Agarwal and Raich (2005). In order to keep a constant search space, all trusses had a 75-foot span, a maximum height of 25 feet, and a second node located five feet horizontally, zero feet vertically from the support.

Each of the proposed characteristics was isolated and used to generate a series of feature maps for five of the test populations. The variations in population size were

important to testing the reliability of the neural network used to create these maps as well as for studying the effects of population size and complexity on the proposed characteristics. It was discovered halfway through the feature investigation that populations containing 100 trusses did not exhibit significant changes in clustering behavior when compared to populations of 50 trusses, and the 100-truss populations were eliminated from further trials. However, truss complexity, in the form of the maximum number of members per truss, did affect the clustering behavior. Accordingly, three additional populations were created for the second phase of the feature trials; these populations examined more variations in truss complexity and replaced the 100-structure populations. Table 2.1 summarizes the characteristics of each test population, and the truss topologies present in each population are displayed in Figures 2.1 through 2.8.

Table 2.1. Summary of Test Populations for Feature Identification.

Name	Population Size	Maximum Number of Members	Phase
S25M50	25 trusses	50 members	1, 2
S50M50	50 trusses	50 members	1, 2
S100M50	100 trusses	50 members	1
S100M25	100 trusses	25 members	1
S100M100	100 trusses	100 members	1
S25M35	25 trusses	35 members	2
S50M25	50 trusses	25 members	2
S50M35	50 trusses	35 members	2

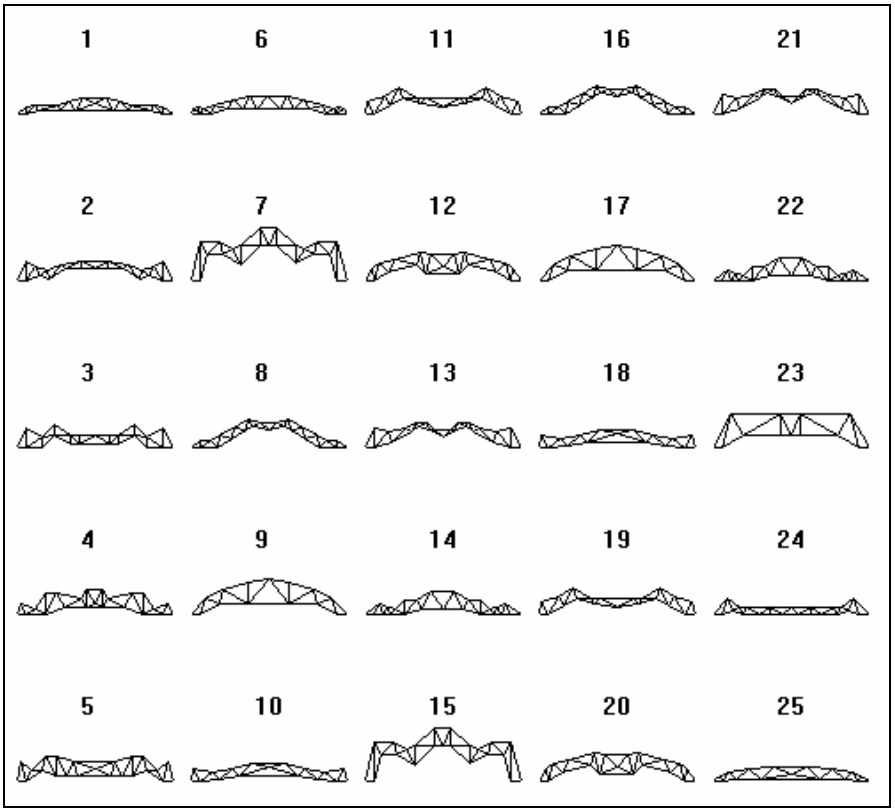


Figure 2.1. Population S25M50 has 25 trusses with 50 members per truss.

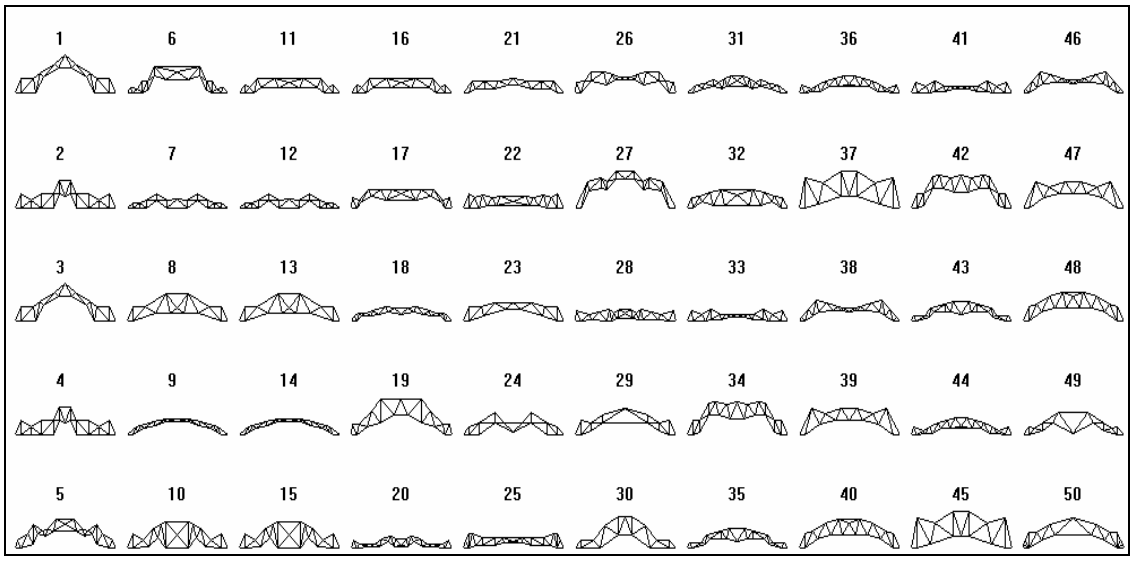


Figure 2.2. Population S50M50 has 50 trusses with 50 members per truss.

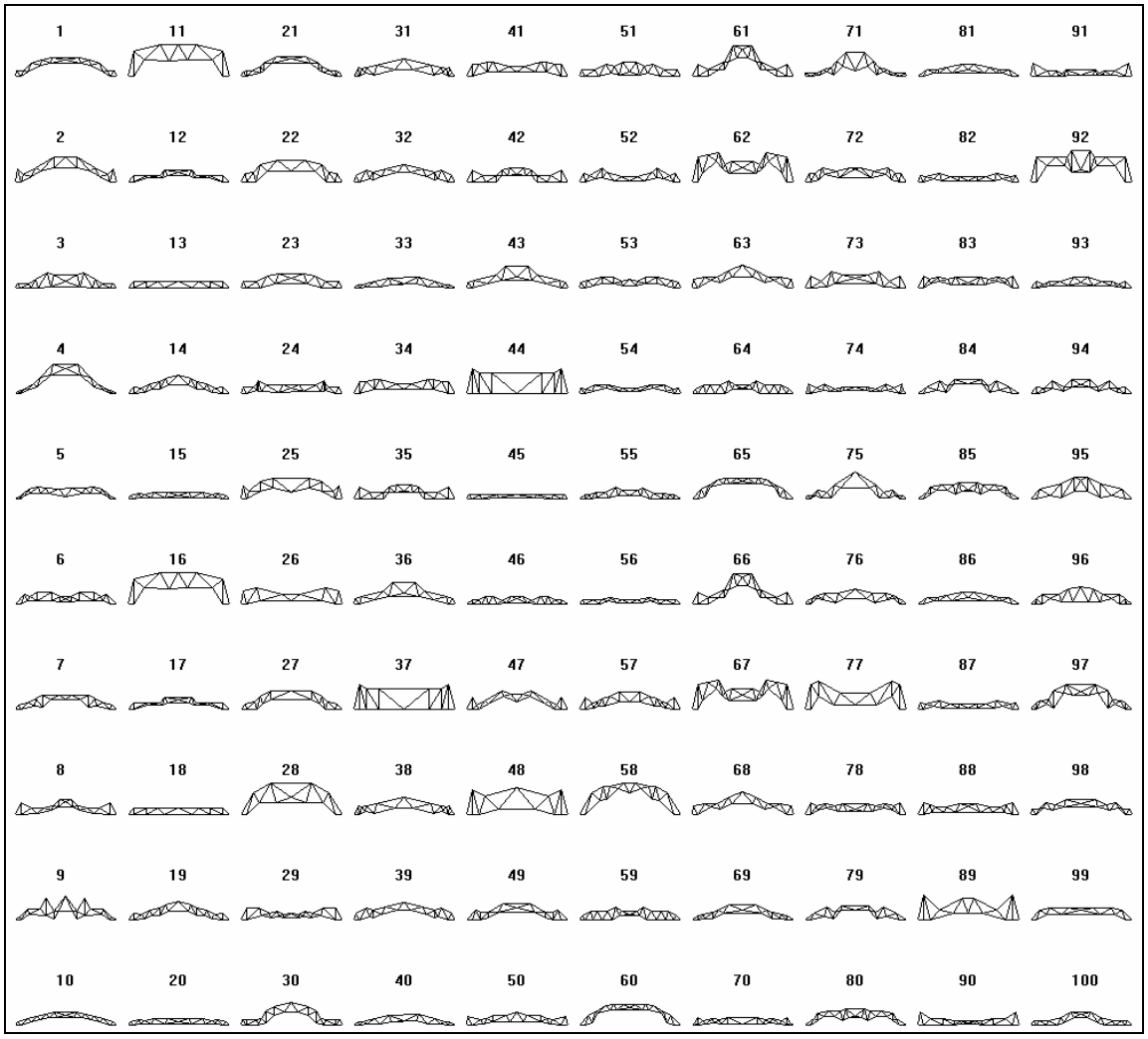


Figure 2.3. Population S100M50 has 100 trusses with 50 members per truss.

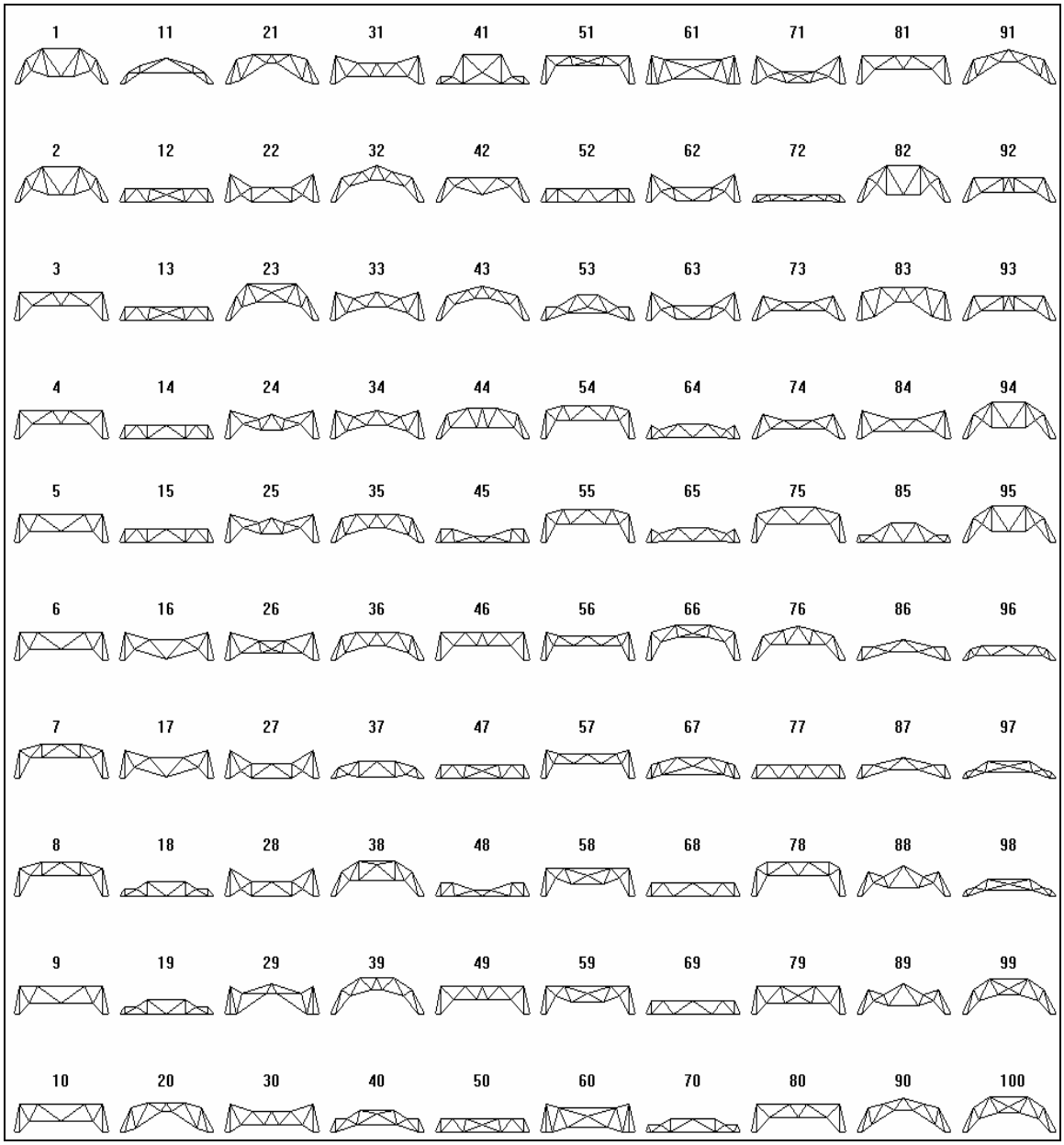


Figure 2.4. Population S100M25 has 100 trusses with 25 members per truss.

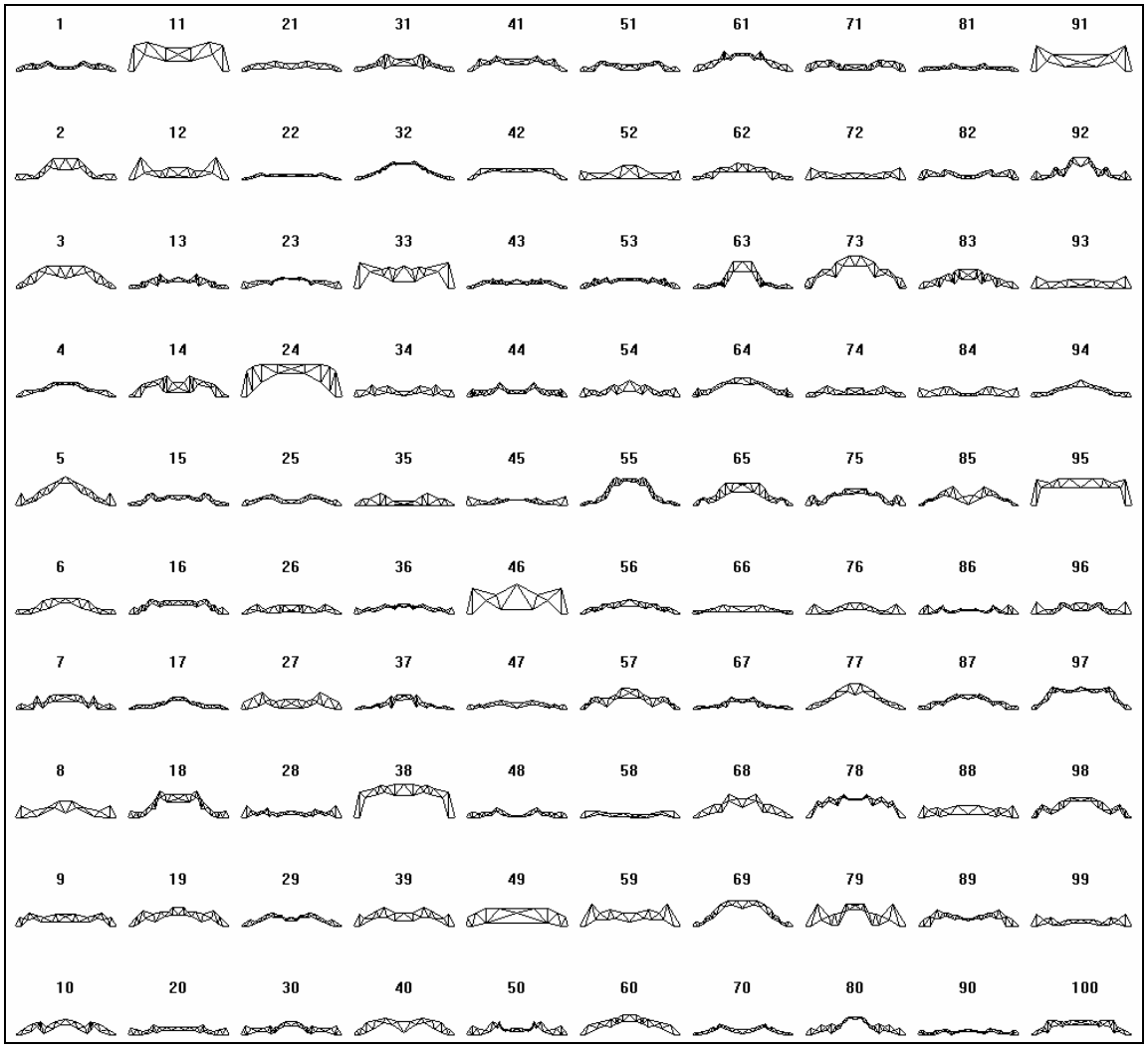


Figure 2.5. Population S100M100 has 100 trusses with 100 members per truss.

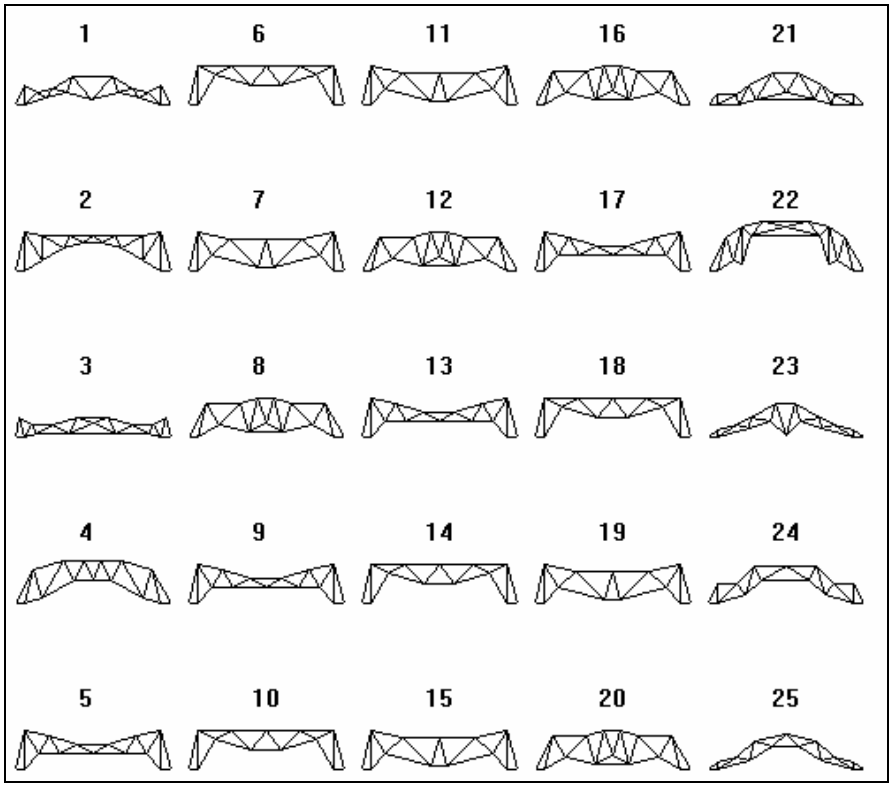


Figure 2.6. Population S25M35 has 25 trusses with 35 members per truss.

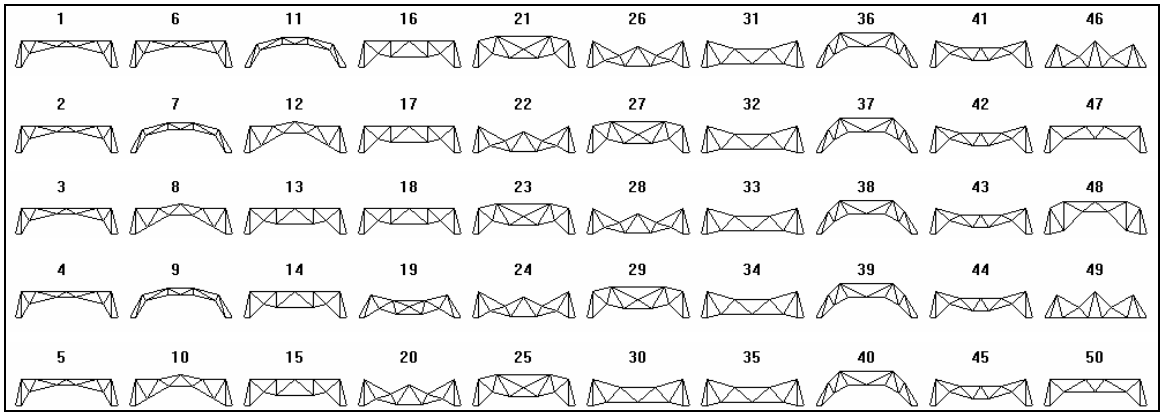


Figure 2.7. Population S50M25 has 50 trusses with 25 members per truss.

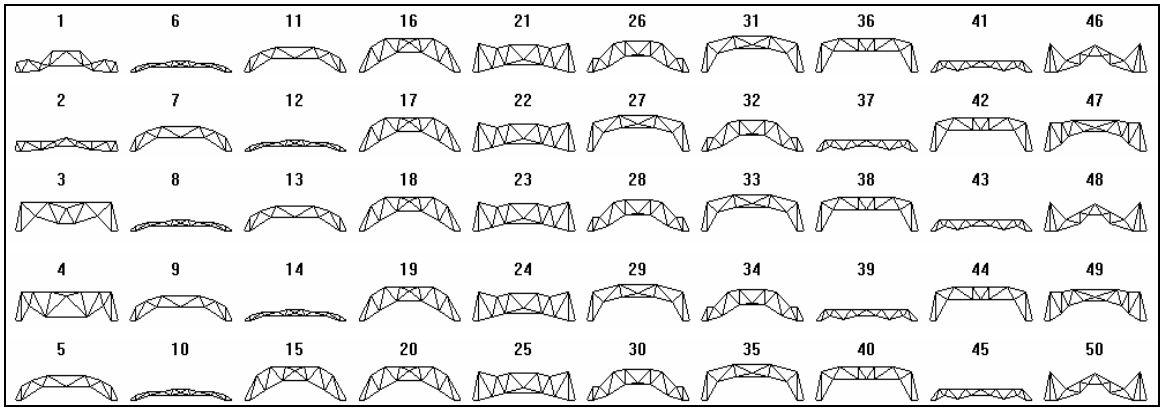


Figure 2.8. Population S50M35 has 50 trusses with 35 members per truss.

Creation of Feature Maps

The one-dimensional (1D) Kohonen self-organizing map (KSOM) was selected as the major tool for creating feature maps of the test populations. The KSOM is an unsupervised neural network that projects an incoming signal onto a discrete, topographic map, which is usually one- or two-dimensional (Haykin 1999). These networks find an "almost optimal" spatial ordering of input information with regard to the underlying statistical features; this ordering may then be used to describe relationships between input data (Kohonen 1988b). The basic premise guiding the self-organizing behavior of a Kohonen map is that "...many originally random local interactions... can coalesce into states of global order and ultimately lead to coherent behavior" (Haykin 1999).

More details on the KSOM implementation will be provided in Chapter III, but a brief description, based largely on Haykin (1999), is presented here for the sake of clarity. As in most neural networks, the KSOM makes use of a layer of input neurons

and a layer of output neurons; these neurons are linked to each other by connections of varying strength, or "weight." These input neurons represent vectors of the information to be mapped. For each input, the closest output neuron (as determined by the Euclidean distance between the input and the weights connecting it to the output neuron) is chosen as the "winner." The weights of this output neuron, and those of a selected portion of its neighbors, are adapted to more closely align with the input (Haykin 1999).

The effect of this adaptation is to shift the weight vectors of the winning output neuron and its neighbors towards the input vector, with the degree of shift dependent on the distance to the winning neuron (Ritter et. al. 1992). Over time, weights begin to respond to particular signal patterns and the network becomes "...tuned to different inputs in an ordered fashion" (Kohonen 1988a). This attenuation, which is the basis of the map's self-organizing behavior, occurs even though the network is randomly initialized.

To put this process into context for the present application, consider the S25M35 test population. This population is comprised of twenty-five trusses. In order to create a feature map of this population for the proposed characteristic of maximum truss height, a vector of the maximum height of each truss is created. One at a time, these twenty-five heights are fed into the KSOM's single input neuron and associated with the closest output neuron. Once the neural network has been adequately trained, the KSOM will be able to place each truss in a group based solely on the relative distribution of maximum heights within population S25M35. Figure 2.9 schematically illustrates this process.

In this simple example, only one truss descriptor was used. Isolating potential

truss characteristics in this manner made it easier to initially determine the characteristic's effectiveness. The goal of the feature vector, however, is to provide several characteristics with which to simultaneously define the truss. In that case, the number of KSOM inputs will correlate to the number of features contained in the vector. In order to eliminate dimensionality problems, the values of all feature characteristics were normalized to lie between 0 and 1 before being used as KSOM inputs.

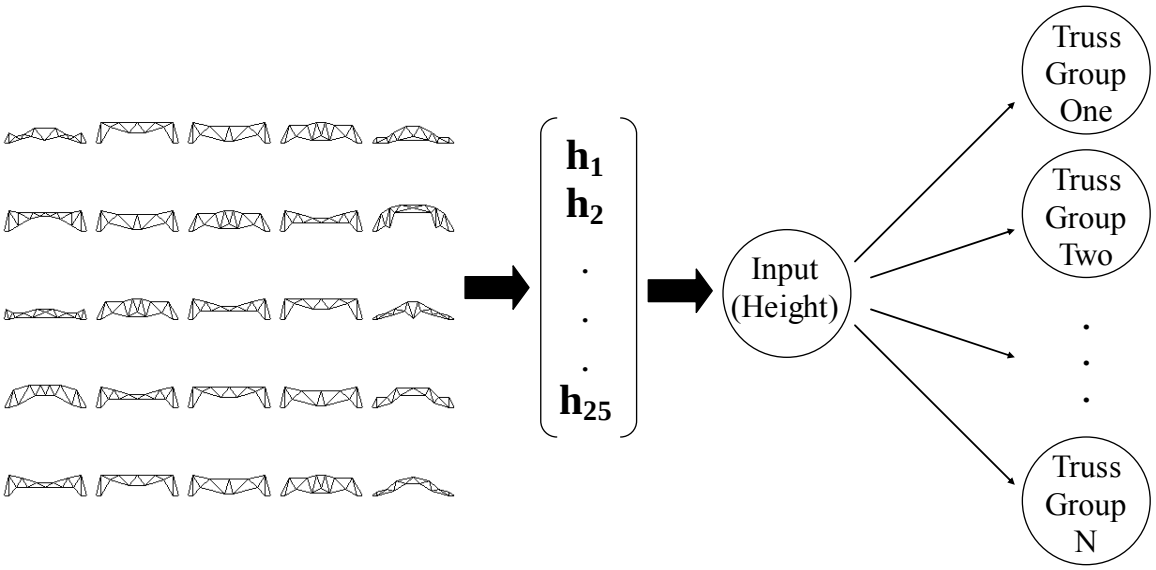


Figure 2.9. KSOM operations for S25M35 with maximum height as input.

The KSOM used in this study underwent two training stages, as suggested in Haykin (1999). The first "ordering" stage, consisting of 1,000 steps, varied the size of the neighborhood surrounding the winning neuron. Initially, the neighborhood included all output neurons, so that all weight vectors were adapted to reflect their relationship to a given input. During ordering of the KSOM, neighborhood size was linearly reduced

until weight adaptation occurred for only the winning neuron. This linear reduction was outlined in Kohonen (2001) as being appropriate for small networks; it is relatively simple to implement, yet provides global ordering of the neurons. The second, 5,000-step training stage was used to ensure that the KSOM converged to a stable solution. These separate ordering and convergence stages were deemed necessary after a shorter training process produced unstable maps; that is, the feature maps generated by the KSOM would change when the random starting positions of the weight vectors changed.

The number of training steps required was determined, in part, by a suggestion in Kohonen (2001) to use 500 steps for each output neuron in the network. During the first phase of this investigation, it was believed that the number of output neurons affected the stability of the KSOM maps. Therefore, smaller populations containing less than 50 truss members used a KSOM with only five neurons, whereas larger populations used KSOMs with nine neurons. These map sizes were determined experimentally.

Once it was discovered that the number of output neurons did *not* affect the stability of an adequately trained KSOM, however, the decision was made to continue using five output neurons for small populations and nine output neurons for larger populations. Although there was no numerical reason not to increase the number of output neurons once the KSOM had stabilized, increasing the number of output neurons increases the number of truss groups that may be defined. In most design problems, asking users to evaluate more than nine groups of trusses would be tedious and unnecessary. Moreover, the KSOM frequently left at least one group empty; that is, the information contained within the input could be expressed using fewer output neurons

than already provided in the existing maps.

Selection Criteria

Information had to be extracted from the feature maps created by the 1D KSOM for each of these twenty potential characteristics to evaluate the quality of the proposed classifications. Several criteria were defined to help make decisions concerning which characteristics would be maintained in the feature vector.

First, and most important from a human user's perspective, the following question was asked: Does this characteristic form a visual partition of the truss population? It was important to determine whether the groups created by the KSOM in response to the feature inputs made significant distinctions between trusses. If the reason why trusses were grouped together was not easily discernible to the human eye, then the characteristic was considered to be a poor descriptor of truss appearance.

The averages and standard deviations for the KSOM groups were also calculated. These statistics were examined for numerical consistency. Characteristics with large standard deviations or closely spaced averages were undesirable, since they indicated that no real order within the characteristic was being mapped by the KSOM. Similarly, the distribution of group averages was considered to ensure that no single truss group dominated the map. Such "super groups" would have illustrated an insufficient amount of information in the input.

Lastly, three-dimensional plots were created to study the clustering behavior of the KSOM. These plots served two purposes. First, they helped determine whether or not enough information was being provided to the KSOM to form rational groups.

Groups were considered rational if their members were closely spaced and the group could be easily distinguished from others. Secondly, the plots helped to illustrate the features' distribution in space, which again indicated how much variation existed within the truss populations. Clearly, features resulting in little or no variation within a population would be unsuccessful at distinguishing differences between trusses.

To create these plots, feature sets of successfully performing characteristics were outlined. Three characteristics were assigned to each feature set, and each of these characteristics formed an ordinate by which an individual truss could be assigned a place in three-dimensional feature space.

Presentation and Discussion of Selected Results

The following section presents the 1D KSOM feature maps created using each of the twenty proposed truss characteristics. For the sake of clarity, detailed results will be shown for population S50M50 only. This population was selected because it was used in both phases of the feature trials and with a consistent number of output neurons. Results for all other populations considered in this investigation are available in Appendix A.

When reading these feature maps, it is important to note that the 1D KSOM map was used to perceive similarities between individual trusses based upon a given feature. Therefore, similar trusses, in terms of values for the current feature, were grouped together. In the KSOM maps, the truss topologies associated with each group are placed

in columns. In addition to the maps, pie charts show the averages and standard deviations for these groups, as well as the relative proportion of trusses in each group.

Maximum Truss Height

During the generation of individual truss topologies described in Agarwal and Raich (2005), the user defines a design space that consists of a maximum horizontal span and vertical height. The trusses topologies vary in the degree to which they vertically fill this search space, which means that some trusses remain very short and compact while others can grow quite tall. An obvious question to ask designers is whether they prefer taller or shorter trusses. To that end, the first feature investigated is the maximum height of a given truss. Since all trusses are referenced from a y-coordinate of 0, the maximum truss height was measured simply by identifying the maximum y-coordinate generated for a given truss topology. Figure 2.10 presents the 1D KSOM results obtained for population S50M50.

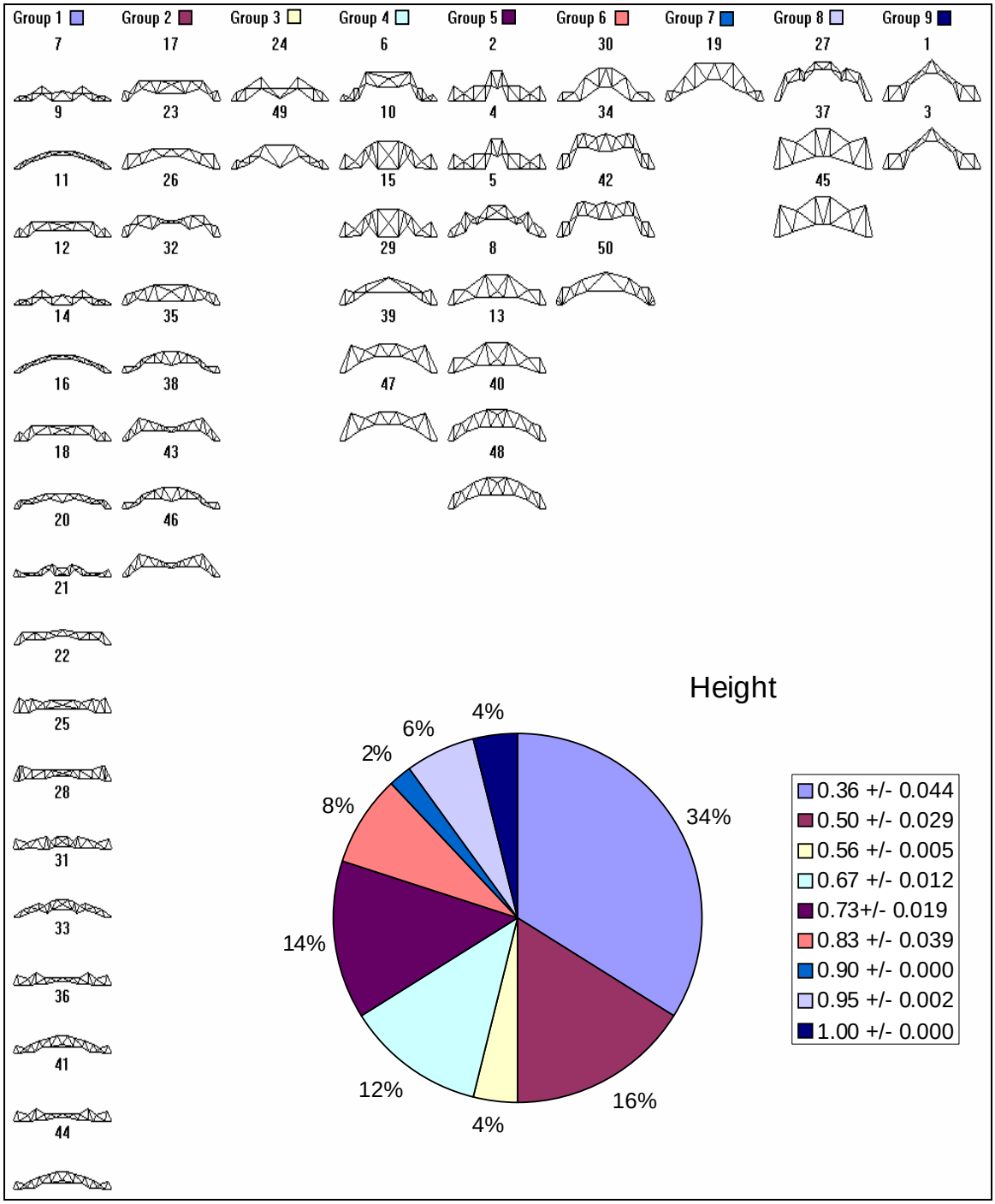


Figure 2.10. Feature map created for S50M50 with maximum height as input.

There is clear height delineation between groups. The KSOM has subdivided the population into a progression from short to tall trusses. The height ranges appear to be numerically significant, and truss groupings make sense from an aesthetic viewpoint. Therefore, maximum truss height appears to be a good characteristic to promote for classification.

Maximum Joint Connectivity

The number of members framing into a joint directly affects the construction cost of a given truss. For this reason, designers may prefer trusses with simple joints. From an aesthetic standpoint, as well, the complexity or simplicity of a joint can lead to an appearance of complexity in the truss as a whole. Therefore, it was proposed that a truss's maximum joint connectivity would be a useful feature to examine.

The maximum joint connectivity for a truss was determined by counting the number of times each joint appeared as an end node for a truss member. The highest number of connections at any joint was then used as input for the KSOM. Figure 2.11 summarizes the results of this analysis for population S50M50 using the maximum joint connectivity feature.

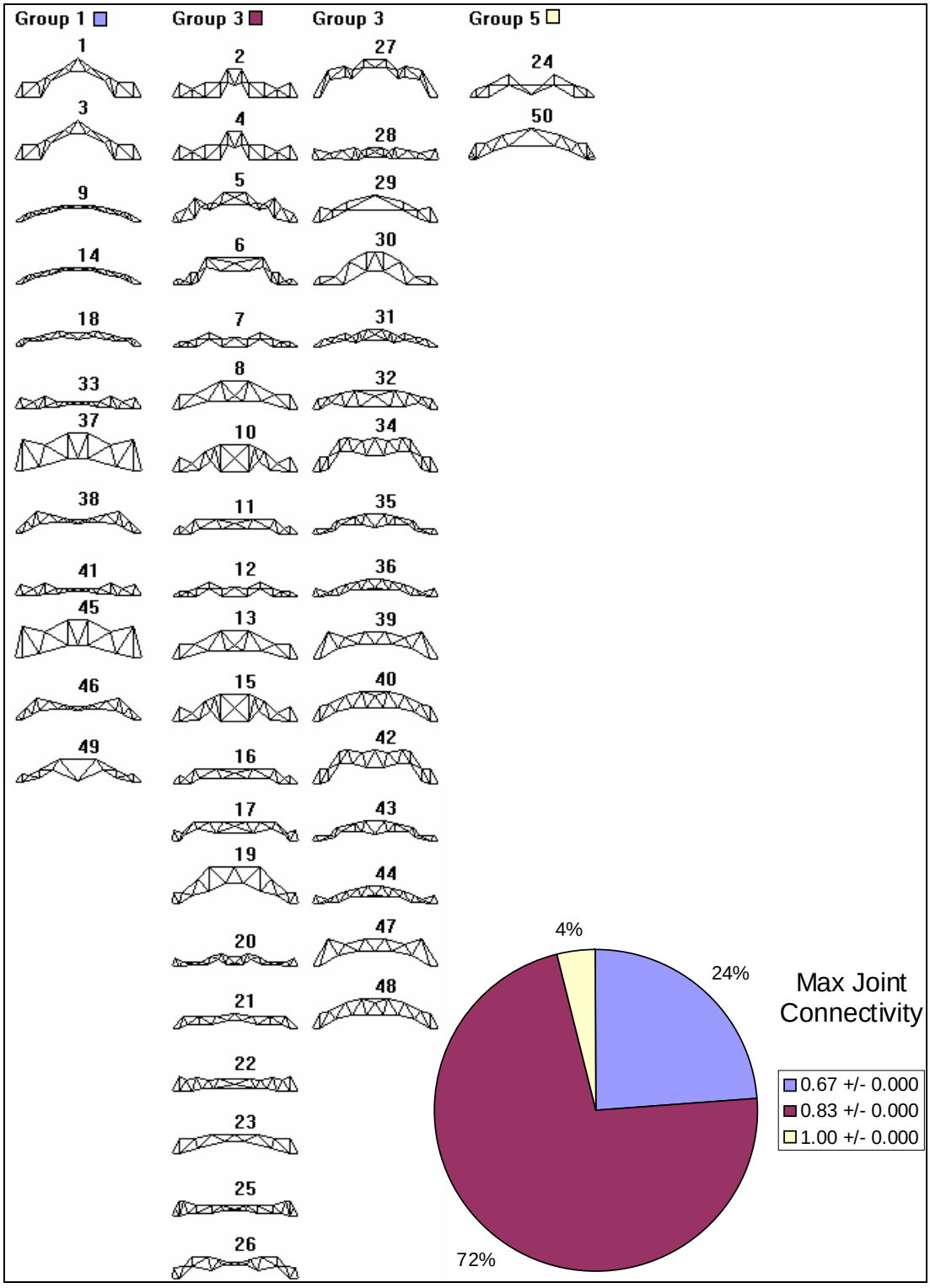


Figure 2.11. Feature map created for S50M50 with maximum joint connectivity as input.

In this population, as well as in the others examined, the maximum joint connectivity characteristic could distinguish only three groups. In each of these groups, the truss topologies bear little resemblance to each other. The physical explanation for this observation is simple. This feature looked only at the greatest number of connections a joint could have; it therefore conveyed information about only a single joint in the truss, rather than the truss as a whole. Therefore, trusses with varying degrees of complexity, but having a single joint with an equal number of members framing into it, would be classified together. For instance, in Figure 2.11, structure 37 in group 1 is one of the simplest trusses in the population. However, it is placed in the same group with structures 9 and 33, both of which are much more narrow and complex.

Overall, this characterization did not offer enough variation between trusses to form effective groups. Moreover, the information it conveyed did not make a clear visual distinction between trusses. It was therefore considered a poor characteristic and was not retained in further analyses.

Average Joint Connectivity

Average joint connectivity was proposed to offer a more global picture of joint complexity than the maximum joint connectivity characteristic discussed above. Average joint connectivity was calculated by summing the number of connections for each joint in the truss and dividing this number by the number of joints. This value would then give an idea of how many member connections an "average" joint would contain. Results for this investigation for population S50M50 are presented in Figure 2.12.

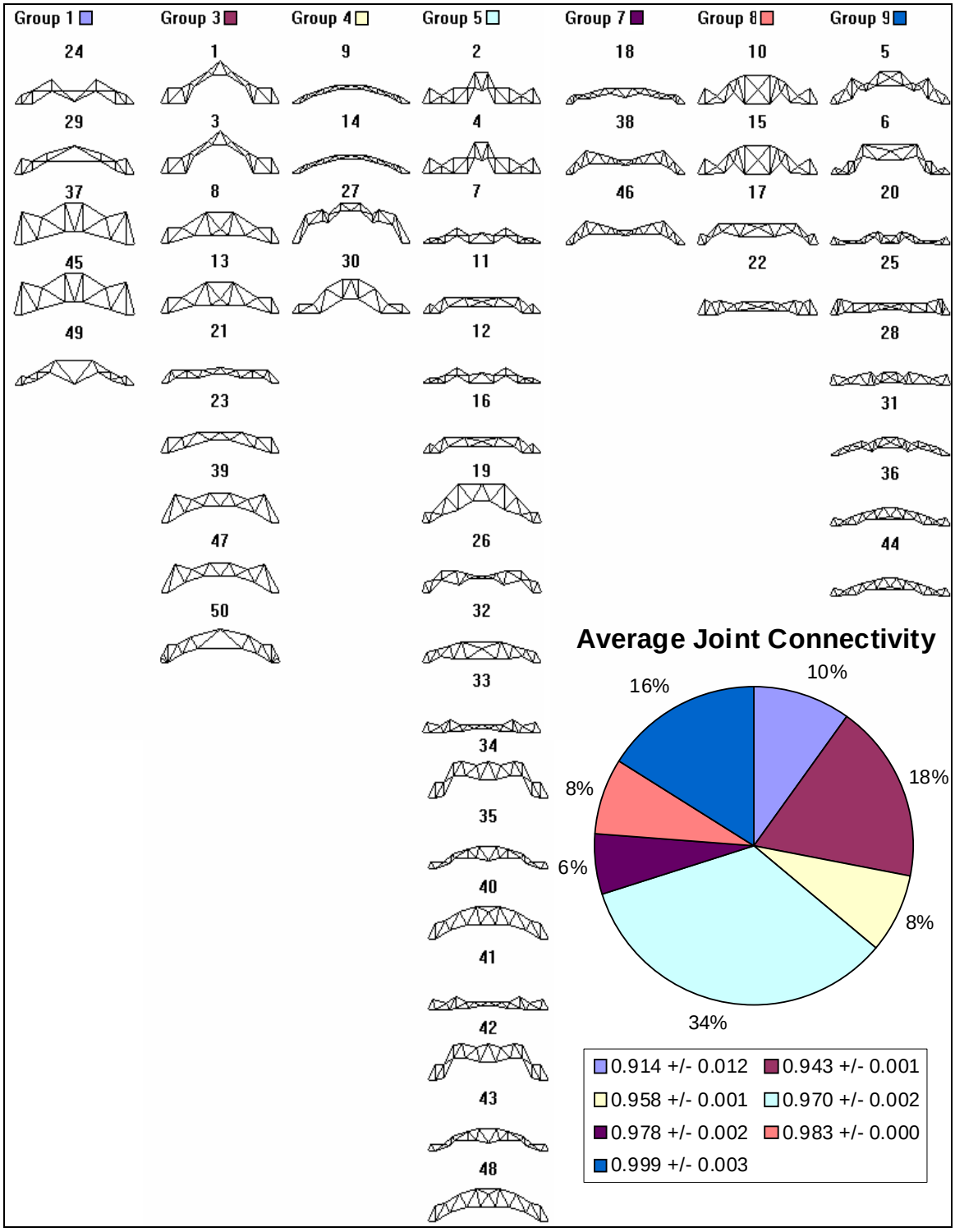


Figure 2.12. Feature map created for S50M50 with average joint connectivity as input.

Overall, average joint connectivity provided a more successful classification than maximum joint connectivity. The KSOM identified more groups with this characteristic, and these groups showed a visible progression from simple to more complex trusses. The only population (S100M25; see Appendix A) where the average connectivity produced only three groups still compares favorably with its maximum connectivity counterpart. Moreover, although limited in range, the average connectivity values within these groups are tightly clustered. This feature provides a good estimate of truss complexity as measured by connection behavior.

Averages for this feature range from 0.9 to 1.0. This limited range occurs partly because the total number of connections was multiplied by a large constant in order to prevent round off problems during type conversions. It also occurs because the first phase of this feature investigation normalized features to have a maximum of 1.0, but did not also enforce a minimum of 0.0. The data for this characteristic emphasized the need to apply a lower bound to feature normalization (implemented during phase two trials) but did not detract from the value of information gathered during phase one trials.

Top Chord Nodes

Top chord shape is a major factor in truss appearance. A first attempt to describe this shape counted the number of joints in the top chord. Top chord nodes were identified by comparing heights of all nodes at a given span location. Member end nodes were also used to determine if members crossed over any potential top chord nodes. The total number of top chord nodes thus identified for a truss was used as a classification feature, the effect of which is summarized in Figure 2.13 for S50M50.

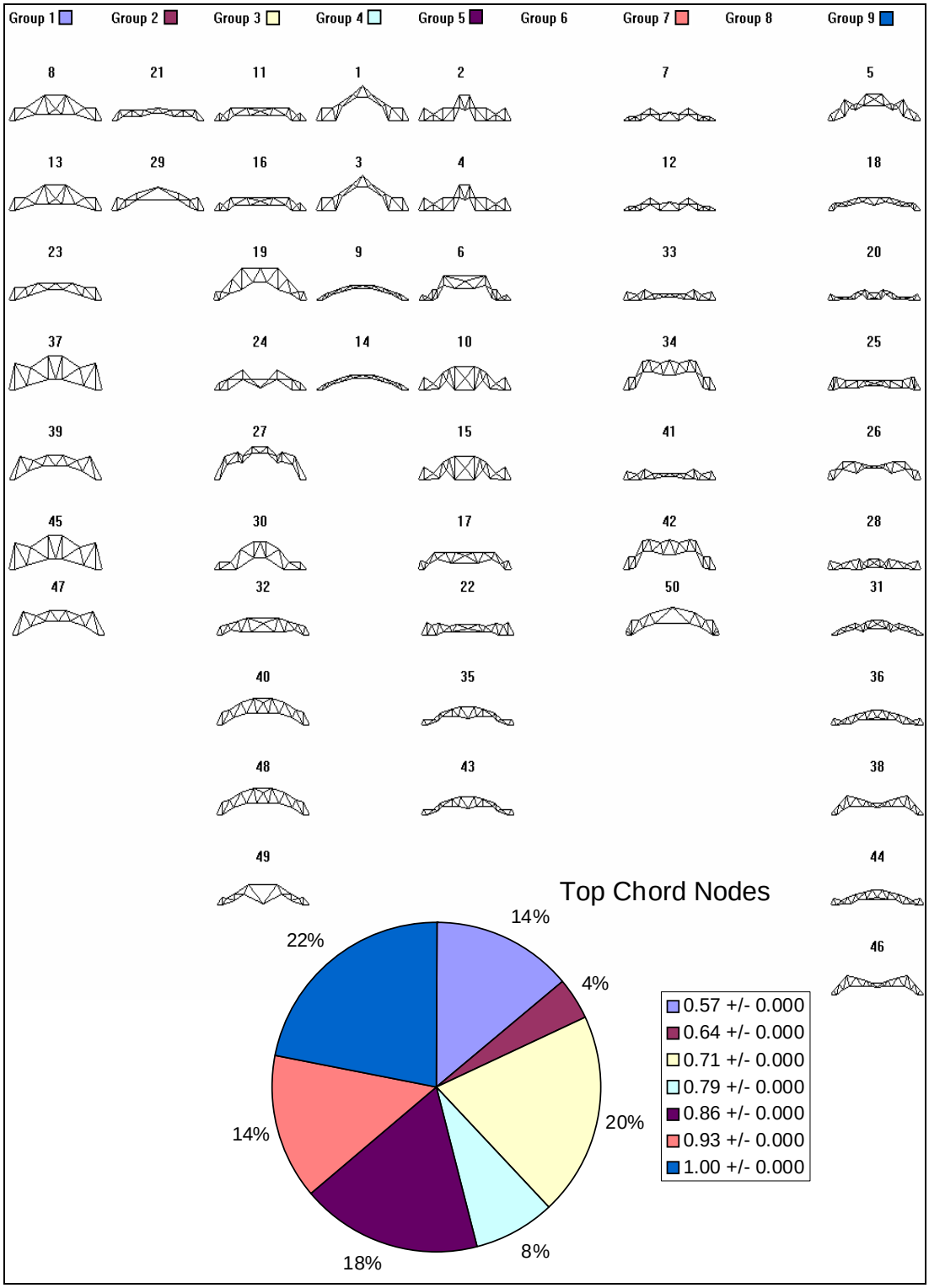


Figure 2.13. Feature map created for S50M50 with number of top chord nodes as input.

Using the number of top chord nodes as input, the KSOM was able to rationally distribute trusses in a given population. This characteristic seems to help indicate the general shape of the truss; for example, in Figure 2.13, the groups vary in both steepness and simplicity. However, the presence of small groups and a lack of a definitive visual phenomenon indicate that this feature may need refinement or further study. In its current form, this feature does not provide a strong basis for classification.

Total Member Length

Although all trusses should have a similar span length, it was thought that variation between the lengths of individual truss members would give an indicator of truss simplicity or complexity. To that end, the member lengths were read from one of the output files created during topology generation. These lengths were simply added for each truss and input into the KSOM. The results of this classification are shown in Figure 2.14 for population S50M50.

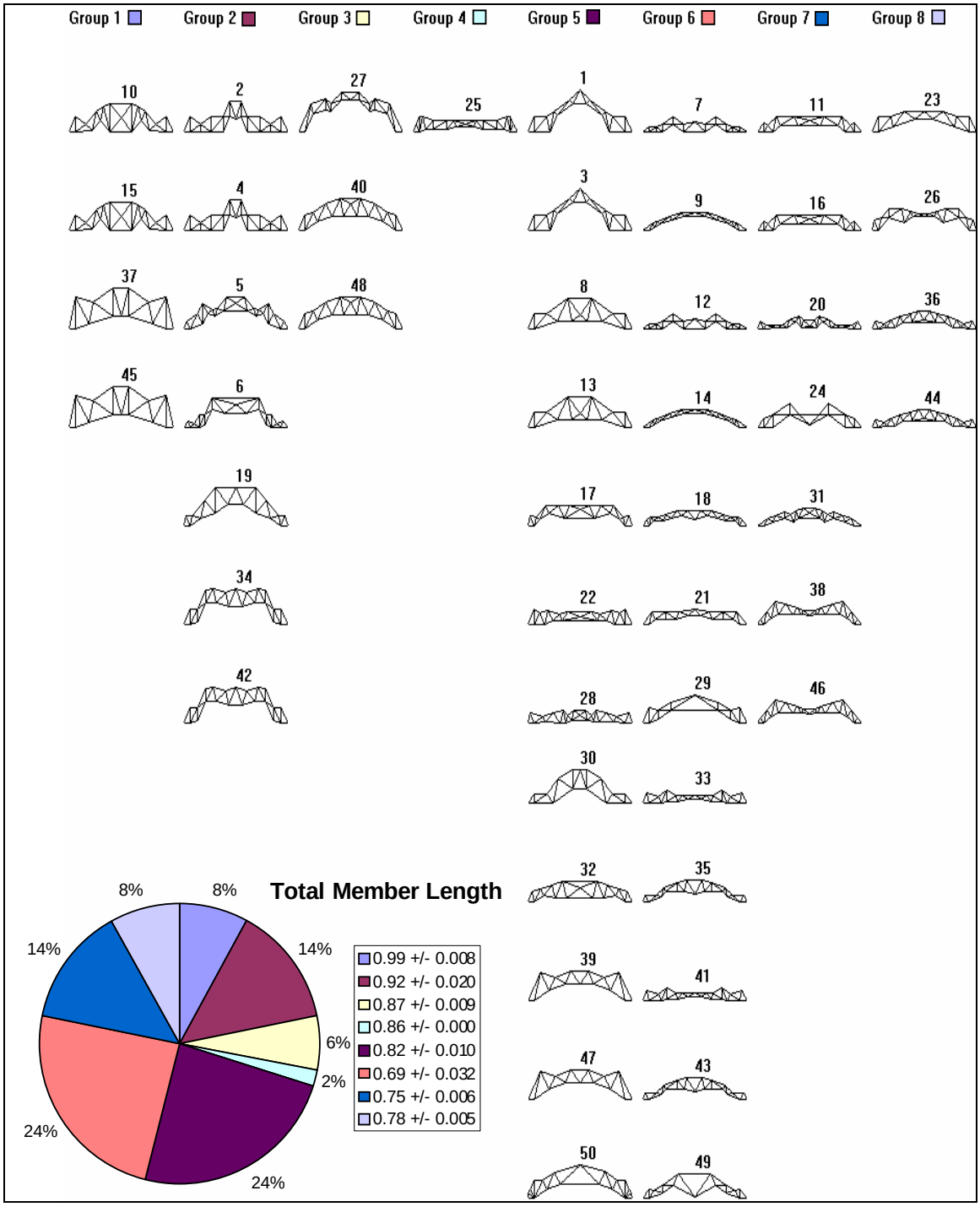


Figure 2.14. Feature map created for S50M50 with total member length as input.

The total member length characteristic failed to group structures according to complexity as originally desired. For example, group 5 of Figure 2.14 contains an array of both very simple and very complex trusses. Moreover, further inspection of the truss classifications did not reveal any other aesthetic reason for the groups selected by the KSOM. From a numerical standpoint, as well, this characteristic performs poorly. The group with the smallest feature average (group 3 with a value of 0.69) is embedded between two groups with higher averages. This placement indicates that the information provided by the characteristic was insufficient to properly train the KSOM.

Ratio of Compression to Tension Members

In some trusses, the arrangement of compression and tension members may be readily apparent by visual inspection, and an experienced designer might take this factor into effect when designing a truss. Therefore, it was thought that a method of quantifying how many compression and tension members were in a truss would be a useful classification technique.

The ratio of compression to tension members was determined using the matrix structural analysis code used by the genetic algorithm to help calculate member fitness. In order to use this program, all truss members were assigned a default cross-sectional area of 5.0 square-inches, and a truss spacing of 30 feet was selected. The analysis then returned the member forces, and separate tallies were kept of the number of tension and compression members in the structure. The ratio between compression and tension members could then be calculated. Figure 2.15 illustrates the KSOM feature map created with this ratio for population S50M50.

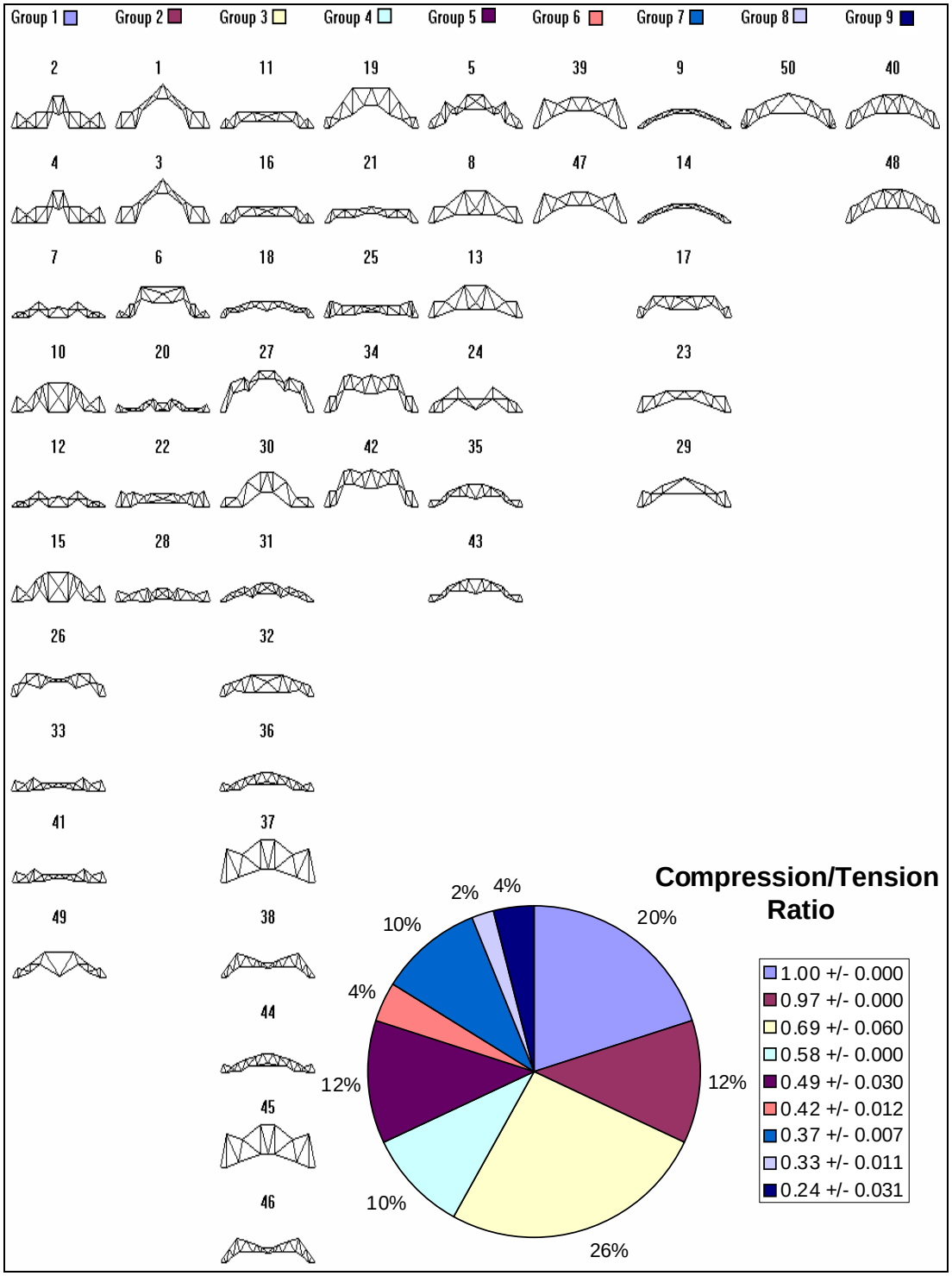


Figure 2.15. Feature map created for S50M50 with compression-tension member ratio as input.

The compression to tension member ratio was another feature that resulted in poor visual classifications. When examining the populations, there seemed little visual reason for the way trusses were grouped. In Figure 2.15, structure 37 and 45 bear a strong resemblance to structures 8 and 13, yet these structures have been placed in separate groups. While these structures are not always paired in other maps (for instance, they differ in average nodal connectivity), separations in that case may be attributed to other physical characteristics (for instance, structures 8 and 13 are slightly more complex). Truss groups in the maps of the remaining populations also lack an identifiable physical trend, such as shape or complexity. It must therefore be concluded that the compression to tension ratio does not offer an easily distinguishable aesthetic feature and should not be retained in further analyses.

Mid-Span Clearance

Truss clearance at mid-span was the next characteristic considered. This feature makes physical sense, since flat-bottomed trusses are easily distinguishable from arched or raised trusses. Bottom chord nodes were determined in the same way that top chord nodes were identified, with opposite criteria; this time, the goal was to identify nodes with the lowest y-coordinate at a set span location and to search for members that ran below these coordinates. This process applied only to those nodes defined as "mid-span" nodes, that is, one quarter of the total nodes in the truss, starting with the nodes closest to the truss's x-axis line of symmetry. Results are shown in Figure 2.16 for population S50M50.

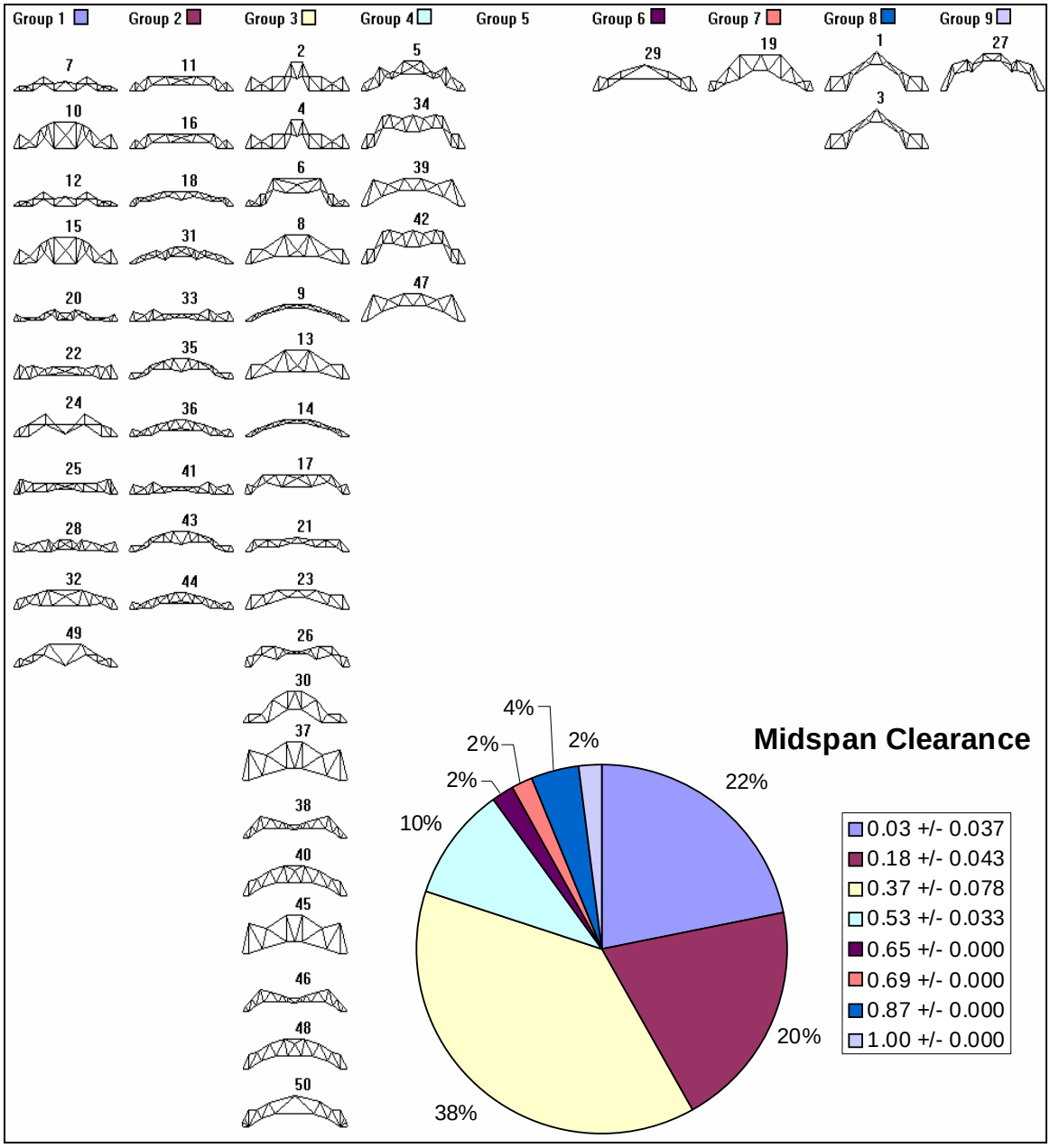


Figure 2.16. Feature map created for S50M50 with mid-span clearance as input.

As expected, mid-span clearance proved to be a valuable characteristic in assessing the form of the sample truss populations. Figure 2.16 reveals a clear progression of trusses with varying mid-span heights. This feature speaks strongly to overall truss shape. Further refinement might involve considering the shape of the bottom chord in more detail or whether mid-spans were flat or angular. Overall, however, mid-span clearance provides a consistent rubric for aesthetic classification.

This feature is also successful at grouping trusses from a numerical standpoint. Standard deviations are small, and the groups do not overlap. Additionally, the groups selected represent significant changes in mid-span clearance, rather than minor progressions.

Number of Nodes

The number of nodes (or joints) that a truss contains directly correlates to its complexity. For this reason, node numbers were taken from the topology input file and used as KSOM input. The effects of this feature on population S50M50 is shown in Figure 2.17.

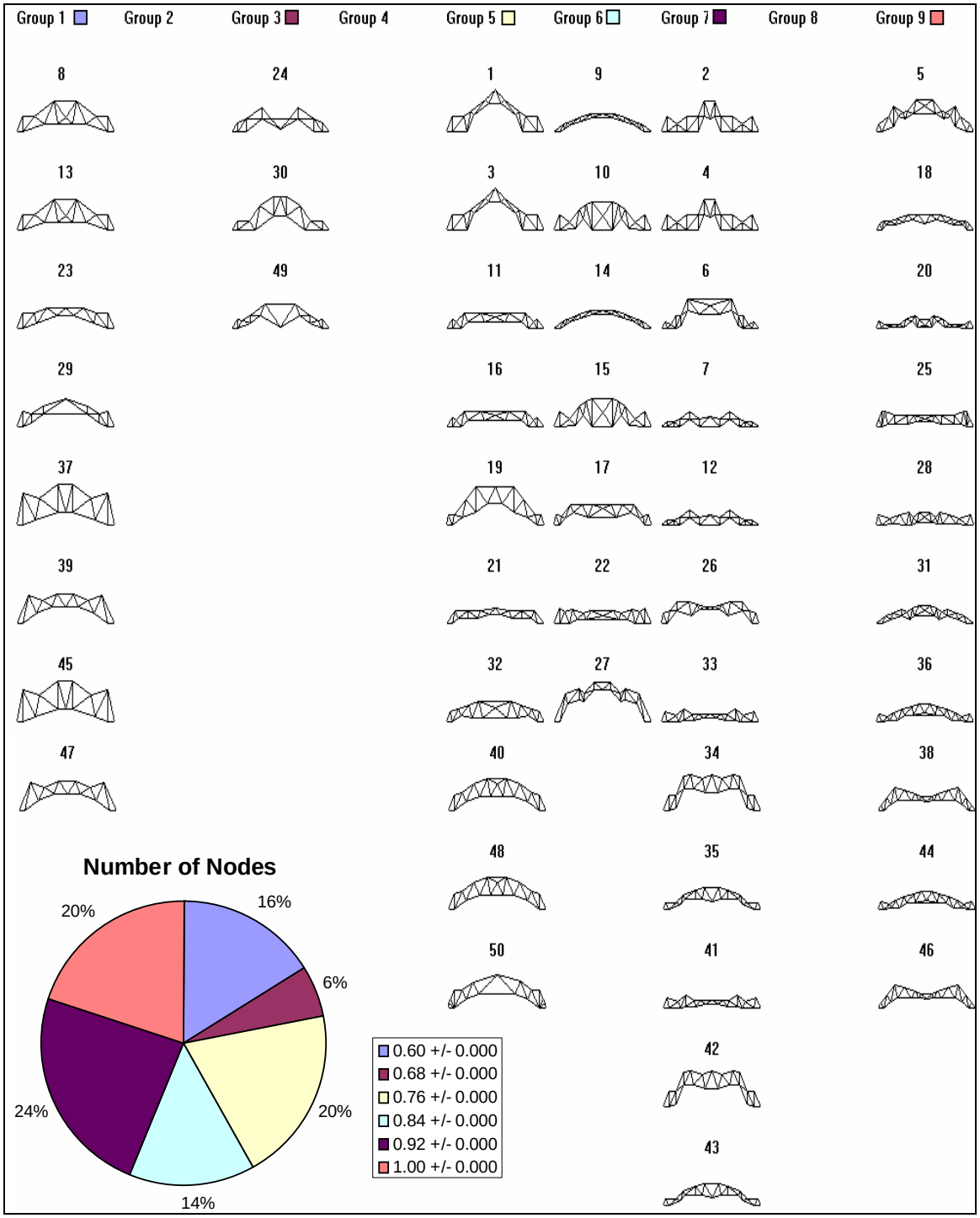


Figure 2.17. Feature map created for S50M50 with the number of nodes as input.

In each population, the number of nodes was able to successfully identify a progression from simple to complex structures. Because this feature was an integer quantity, the number of groups discovered by the KSOM was strictly limited by the available data. This limitation actually serves to make the number of nodes a powerful descriptor in that all trusses within a group are identical for this feature, and the discovered groups represent physical variations that exist within the population.

In population S100M25 (see Appendix A), this criterion created only two groups. While these groups show a clear dichotomy in terms of complexity, all structures in this population contain a maximum of twenty-five members and may be considered fairly simple. In fact, this population was specifically chosen to illustrate the effects of the KSOM features on basic trusses. Therefore, features intended to measure complexity would find little variation in the S100M25 population; results for average nodal complexity and other characteristics that measure simplicity support this conclusion.

Overall, the number of nodes in a truss was considered a good indicator of truss complexity from both a numerical and visual standpoint.

Number of Members

The number of truss members was also considered as a potential feature for truss classification. The number of members should indicate truss complexity while offering a different perspective than provided by the number of nodes. For instance, a truss may have few nodes with many member connections; to properly classify this design, both node and member quantities would prove useful. Figure 2.18 illustrates the map generated for population S50M50 using the number of members as input.

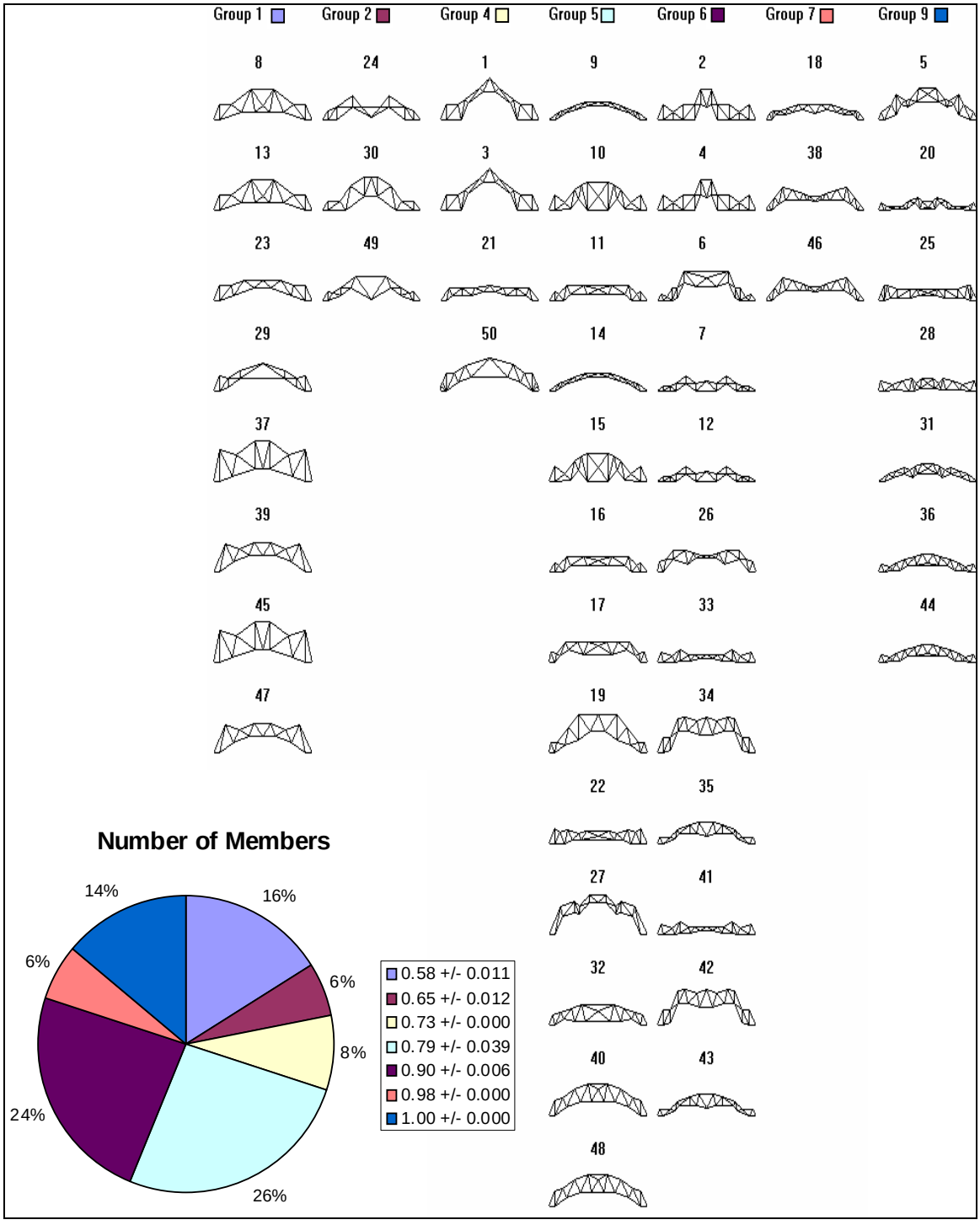


Figure 2.18. Feature map created for S50M50 with the number of members as input.

As with the number of nodes, the number of members presents a decided variation in truss complexity, yet this second feature does not duplicate information. The first two groups of Figure 2.18 echo the first two groups of Figure 2.17, but important differences exist between these two maps. One difference is that the number of members generated seven truss groups, whereas only six groups were discovered using the number of nodes. Moreover, this characteristic helps distinguish trusses with intricate member configurations. Therefore, it is suggested that using both node and member quantities will improve the KSOM simplicity classification.

Top Chord Concavity

As mentioned earlier, the shape of a truss's top chord has a significant impact on its appearance. A first attempt to capture this shape was made by enumerating the number of top chord nodes. Another characteristic examined to improve the KSOM's ability to distinguish top chord shapes is top chord concavity.

In this context, concavity was defined as whether the height at mid-span was equal to the maximum height of the truss. This property consisted of a simply, binary decision with a value of 1 if the truss was concave, 0 if not. If the truss had a node located on the line of symmetry, this node was considered mid-span; if not, the two nodes closest to this line were selected. In addition, a two-inch tolerance was used when considering whether a mid-span node was at maximum truss height. This tolerance was necessary to avoid misclassification due to small numerical differences along top chord node heights that would not be detectable to the human eye. Figure 2.19 depicts the concavity-based feature map for population S50M50.

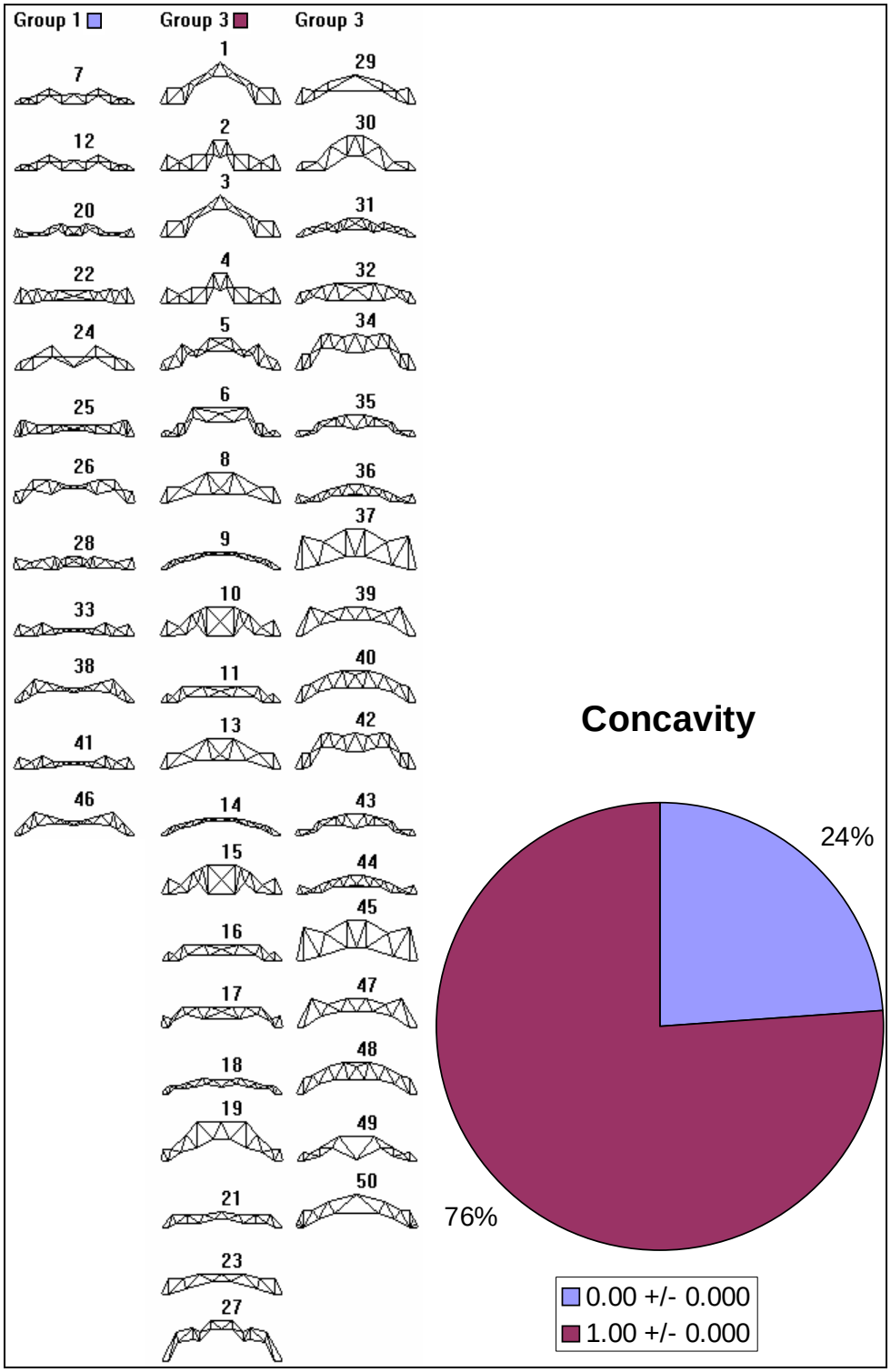


Figure 2.19. Feature map created for S50M50 with top chord concavity as input.

For all truss populations, only two groups were identified by the KSOM using the top chord concavity input. This result was expected and necessary, since concavity involved a binary classification. All trusses in these populations were correctly classified as concave or non-concave, based on the requirements previously explained. Concavity makes a quick, easily distinguishable decision about the aesthetic features of the truss and was therefore deemed an important characteristic that should be considered in more detail.

Top Chord Dip

Since top chord concavity focused only on truss height at mid-span, rather than whether or not that height was achieved by monotonic growth, it seemed advisable to keep track of whether or not a dip occurred in top chord height. This factor was determined by examining successive top chord nodes for negative height differentials. If a height drop occurred, then the truss was labeled as containing a dip. As with top chord concavity, this feature represented a binary value of 1 for trusses with a dip and 0 for trusses without a dip. Results are shown in Figure 2.20 for population S50M50.

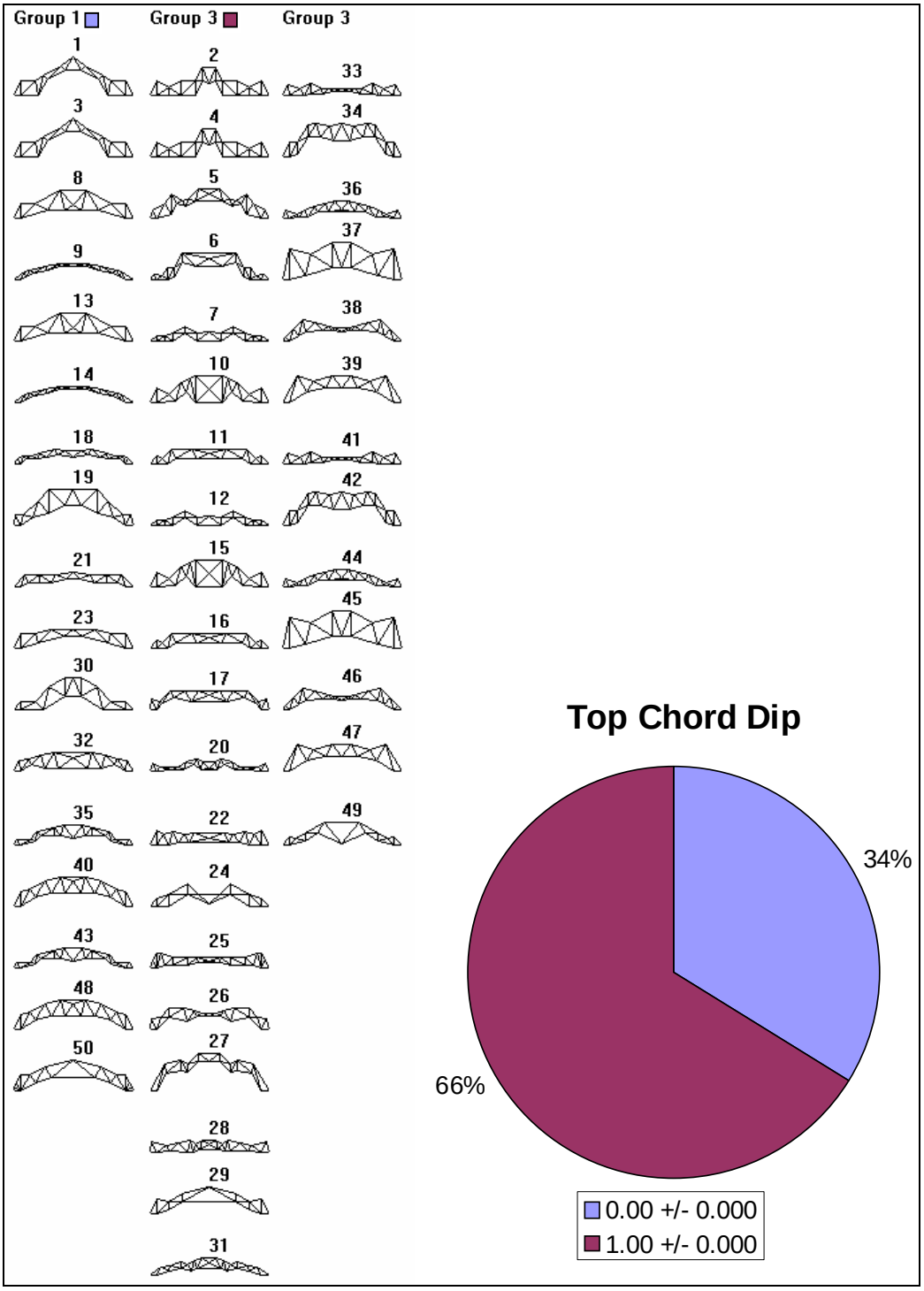


Figure 2.20. Feature map created for S50M50 with the presence of a top chord dip as input.

Like top chord concavity, the presence of a top chord dip creates two distinct groups in all five populations. This feature is a simple decision that makes a clear aesthetic decision. However, as with top chord concavity, the fact that this characteristic forms only two groups raises questions about its overall efficacy to classify trusses in conjunction with other aesthetic features.

Top Chord Flatness

The amount to which height varies along the top chord is also a significant factor in determining truss shape. Therefore, a method for determining relative differences in top chord flatness seemed appropriate. Both phases of the feature investigation considered measures of top chord flatness.

In the first phase, the heights of successive top chord nodes were compared and the absolute value of any height differentials summed. This total was then divided by the maximum truss height to provide a normalizing effect between tall and short trusses. Results for this measure were mixed. Overall trends in truss steepness and the number of direction changes were imperfectly captured.

In an effort to refine this characteristic during the second phase of the investigation, the decision was made to divide the total height differential, not by truss height, but by the number of top chord nodes. This alteration is intended to give a more accurate average measure of top chord flatness. The results of this revised measure for population S50M50 are shown in Figure 2.21.

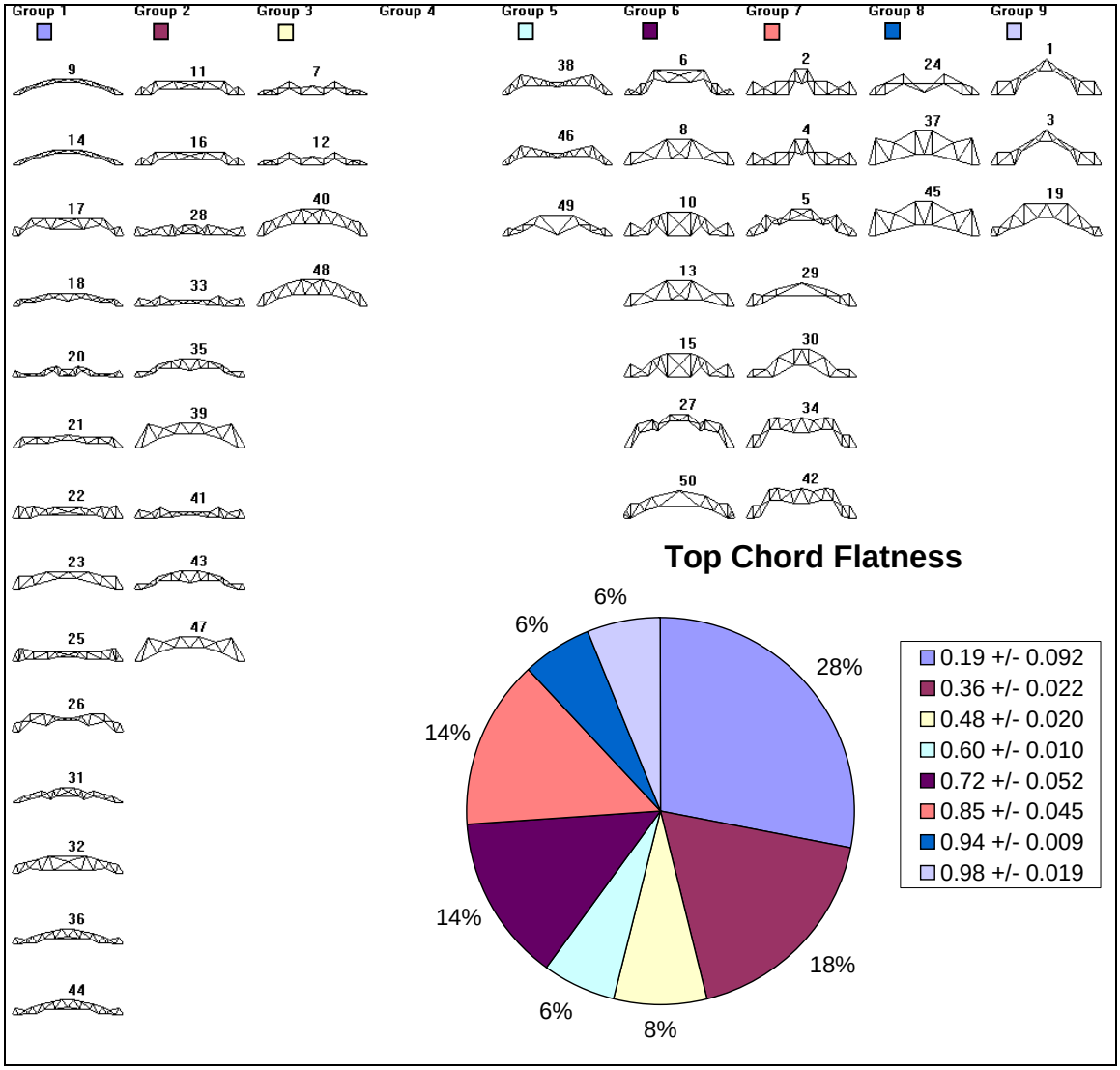


Figure 2.21. Feature map created for S50M50 with top chord flatness as input.

A clear progression from flat top chords to angular designs is visible in Figure 2.21 and in feature maps for the additional populations located in Appendix A. This feature also identifies similar truss types: trusses in group 9 arch to a single peak at mid-span, whereas trusses in group 8 are arched but contain at least one significant dip. The groups are well defined numerically with well-distributed group sizes.

Bottom Chord Nodes

The number of top chord nodes proved potentially useful in classifying truss shape. A similar tally of bottom chord nodes was proposed to determine if this feature also benefited classification efforts. Figure 2.22 presents results for population S50M50.

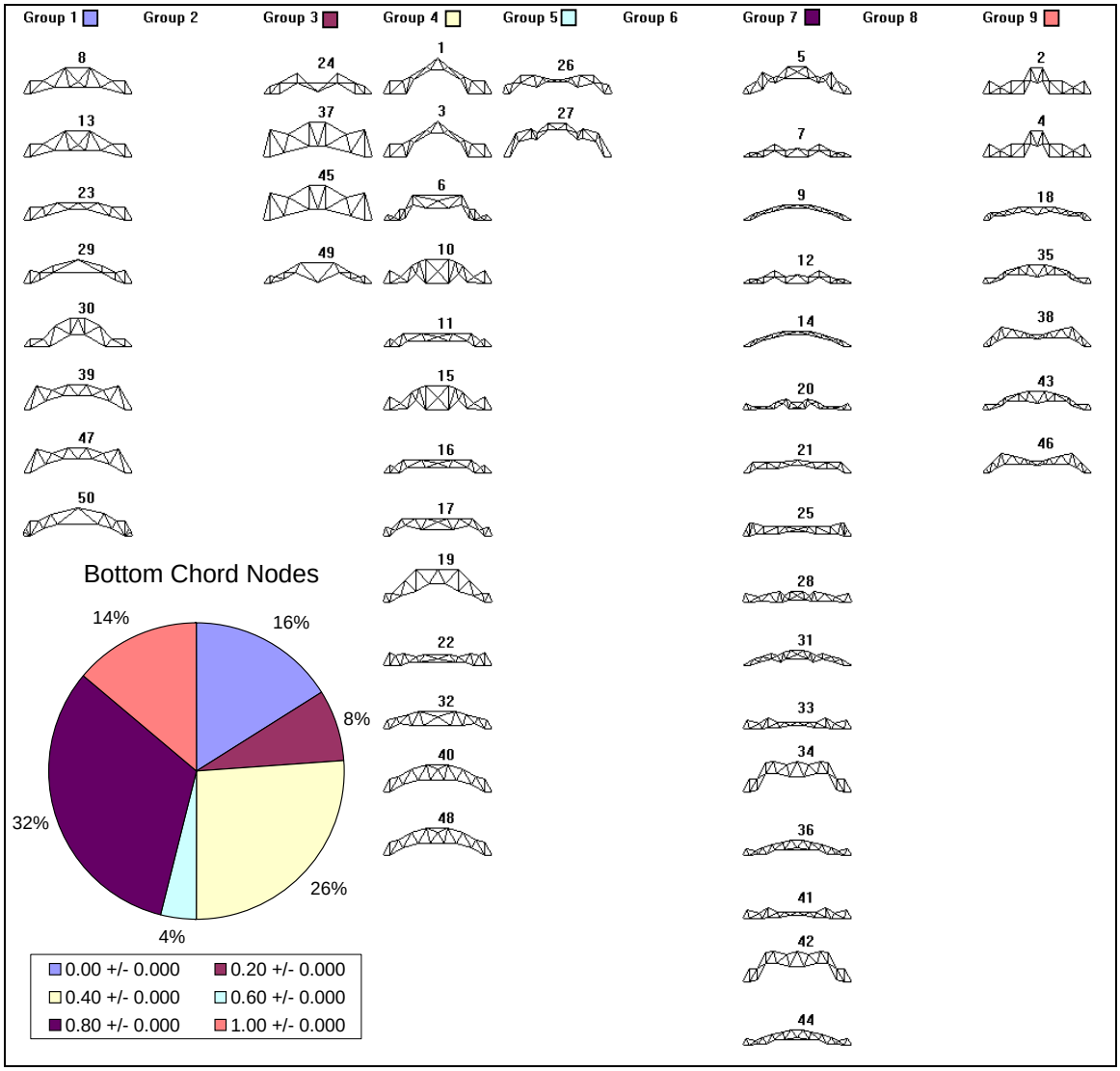


Figure 2.22. Feature map created for S50M50 with number of bottom chord nodes as input.

In the simplest populations, the number of bottom chord nodes appears to correlate with truss simplicity (see Appendix A). However, this trend appears to break down for more complicated populations, as in Figure 2.22. Moreover, characteristics describing truss simplicity are already well defined by the number of members, number of nodes, and average joint connectivity features. Therefore, the number of bottom chord nodes does not provide additional information to truss classification.

On its own, the number of bottom chord nodes, like the number of top chord nodes, does not create a clear visual distinction between trusses. From the group averages presented, however, it is easy to see that it makes clear decisions between trusses and may therefore be useful in combination with other characteristics.

Ratio of Top to Bottom Chord Nodes

Treated as individual characteristics, the numbers of top and bottom chord nodes do not readily supply information about truss aesthetics. However, a ratio of these values might serve as a useful classification feature. It seems reasonable that differing levels of chord complexity would indicate differing truss shapes. This new feature was defined by dividing the number of top chord nodes by the number of bottom chord nodes. Since both values were integers, a multiplier of 100 was used to provide information at the hundredths level. Figure 2.23 illustrates results for population S50M50.

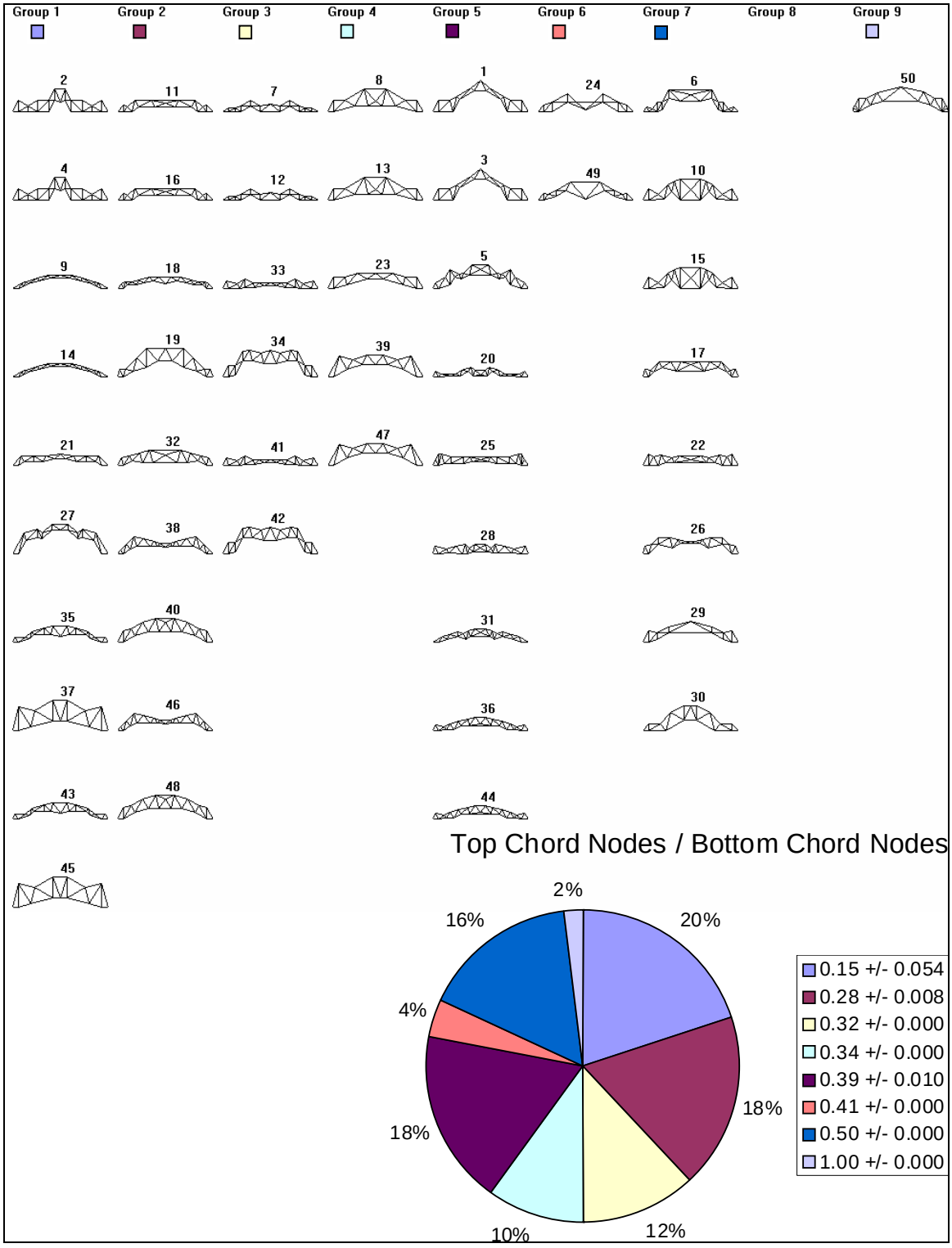


Figure 2.23. Feature map created for S50M50 with ratio of top to bottom chord nodes as input.

This feature distinguishes between trusses based on a bias towards top or bottom chord complexity. However, for complicated populations, such as S50M50, the distinguishing lines are cut rather fine. A glance at the average value for the groups in Figure 2.23 will show that small differences (difficult to notice in the feature maps) may cause unnecessary divisions in the population.

Visually, it is possible to regard these feature maps and determine that the groups have a top chord bias, a bottom chord bias, or a lack of bias towards either. However, this three-point rubric appears to be all that is necessary, and it lacks the expected aesthetic impact. Therefore, the ratio of top and bottom chord nodes does not appear to provide a strong characteristic for truss classification.

Bottom Chord Flatness

A measure of bottom chord flatness was also calculated. Height differentials among bottom chord nodes were summed and divided by the number of nodes. The goal of this characteristic was to capture the overall appearance of the bottom chord and parallel the top chord flatness criteria. Figure 2.24 shows results for population S50M50.

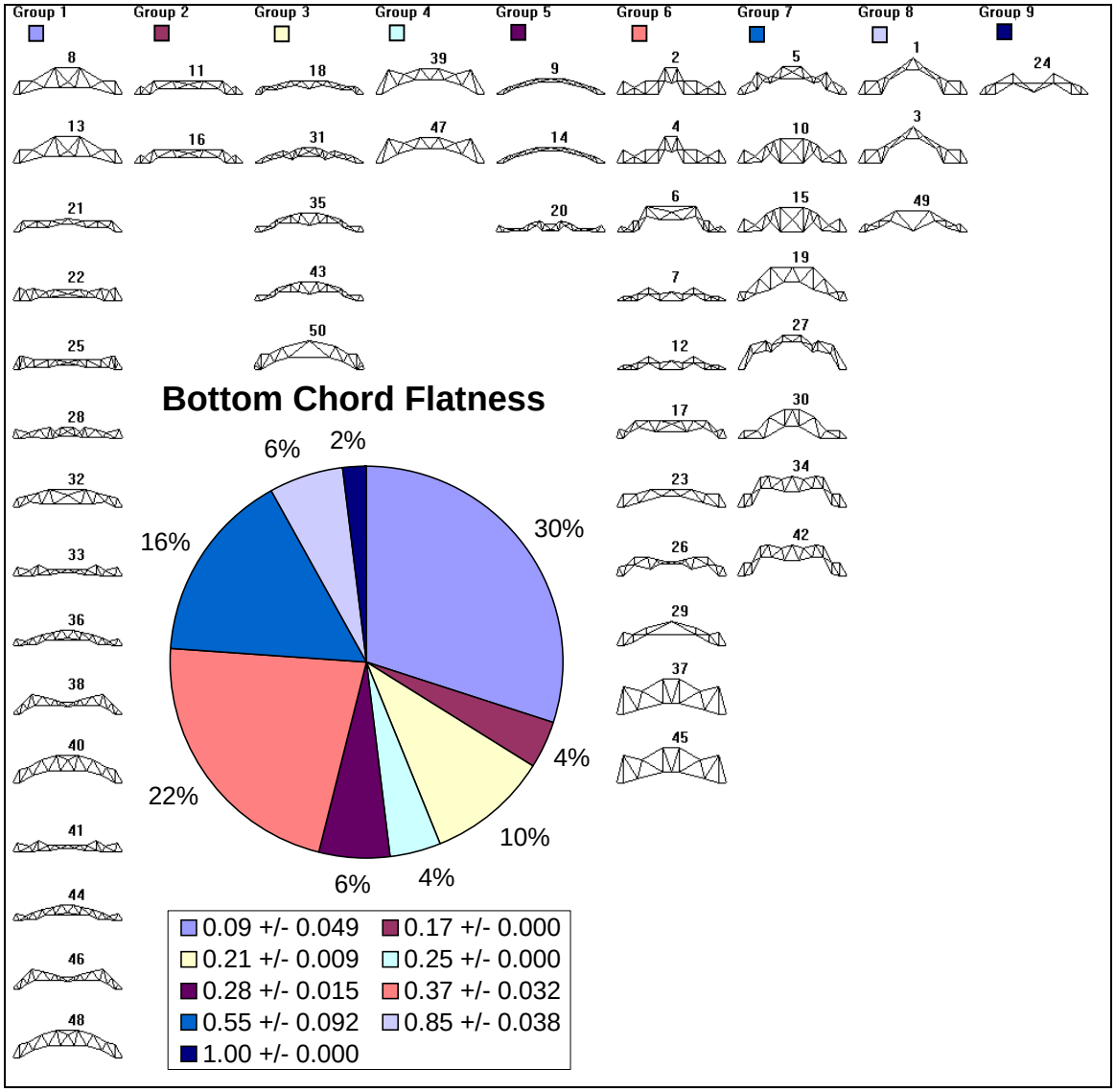


Figure 2.24. Feature map created for S50M50 with bottom chord flatness as input.

This measure of bottom chord flatness appears to be adequate for classifying different bottom chord shapes. It successfully separates the most flat-bottomed trusses from arches with more variation. As with top chord flatness, feature averages for some adjoining groups lie close together. These groups, however, show significant visual distinctions between member trusses. Therefore, bottom chord flatness may prove to be a valuable feature for truss characterization.

Top Chord Angularity

In attempting to measure the flatness of the top chord, a proposal was made to look at the angles between adjacent top chord nodes. It was thought that the sum of these angles would provide information about variations along the top chord. Vertical and horizontal distances between adjacent nodes were used to provide an inverse tangent measure of the angle, with all negative angles being translated into positive coordinates.

Because of stability problems with this measure, no more than five output neurons (classification groups) could be used for a feature map. If more neurons were used, the groups identified were not unique. Results of this classification for population S50M50 are presented in Figure 2.25.

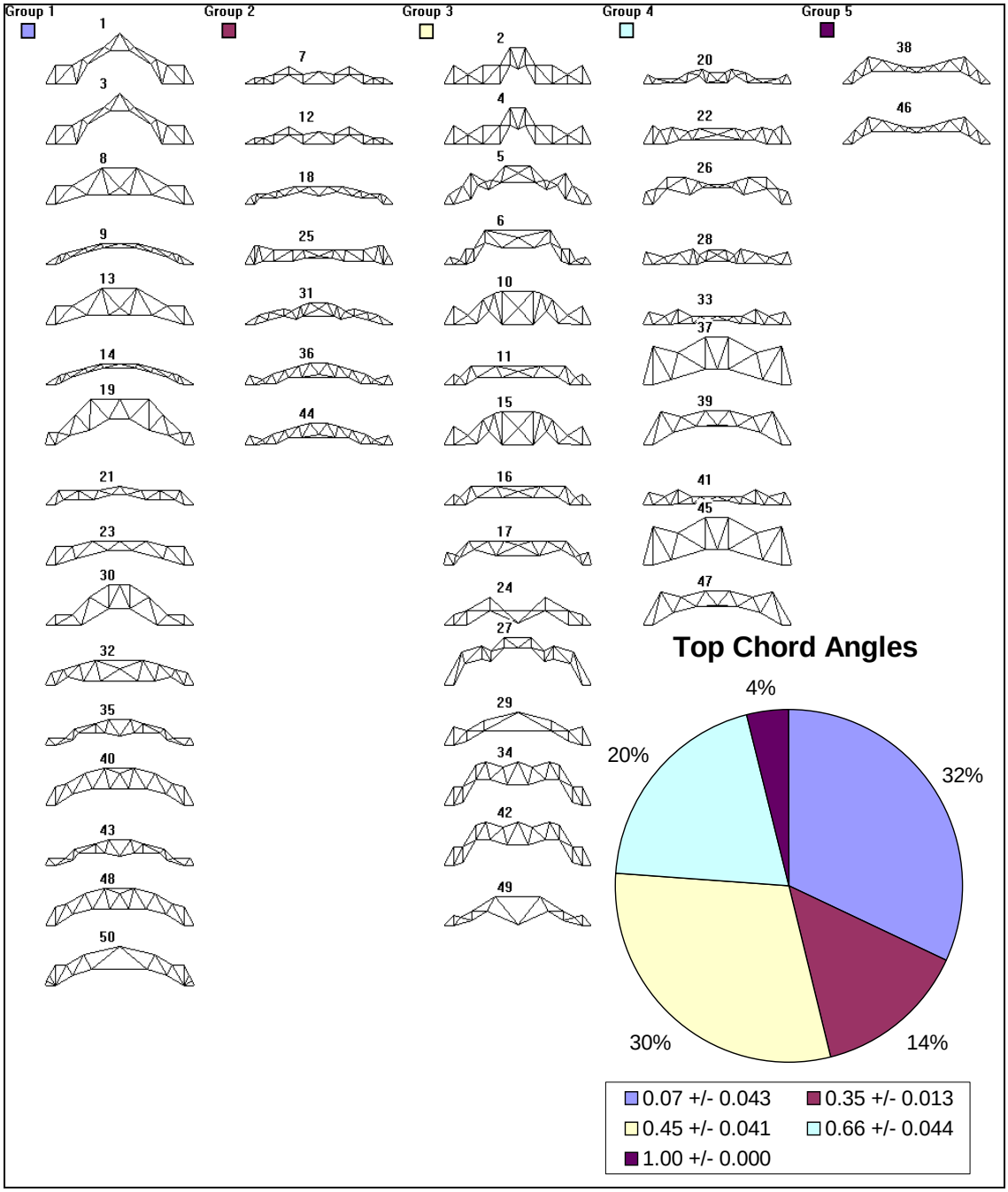


Figure 2.25. Feature map created for S50M50 with top chord angles as input.

Results for classification using top chord angles are mixed. This feature does a

good job of identifying top chords with similar shapes and angular variations for the simplest populations (S25M35, S50M25, and S50M35 in Appendix A). The characteristic does not perform as well for more complicated populations, however, and it is hard to identify clear patterns among these groups, as shown in Figure 2.25.

Perhaps this breakdown in classification ability may be attributed to the number of duplicate structures appearing in the simpler populations. Structures requiring fewer members have a greater chance of encountering duplicates in their population than structures with a higher limit on the number of members. Therefore, the KSOM may have more success classifying populations with a large number of duplicates. It seems more likely, however, that these populations are more similar with respect to this feature, and it is therefore easier to visually distinguish its effect on the population.

Another consideration against using the top chord angles for truss classification is its instability. Having to restrict larger populations to five groups will be a significant disadvantage in later analyses. It may overly constrict the feature maps, thus causing other classification methods to perform poorly.

Therefore, the sum of top chord angles, though able to capture significant patterns in simple populations, is a problematic feature overall. In its current form, it should not be retained for the final implementation.

Bottom Chord Angularity

To parallel the top chord angularity examination, the sum of all angles lying between adjacent bottom chord nodes was also considered as a potential truss feature. Figure 2.26 presents results for population S50M50.

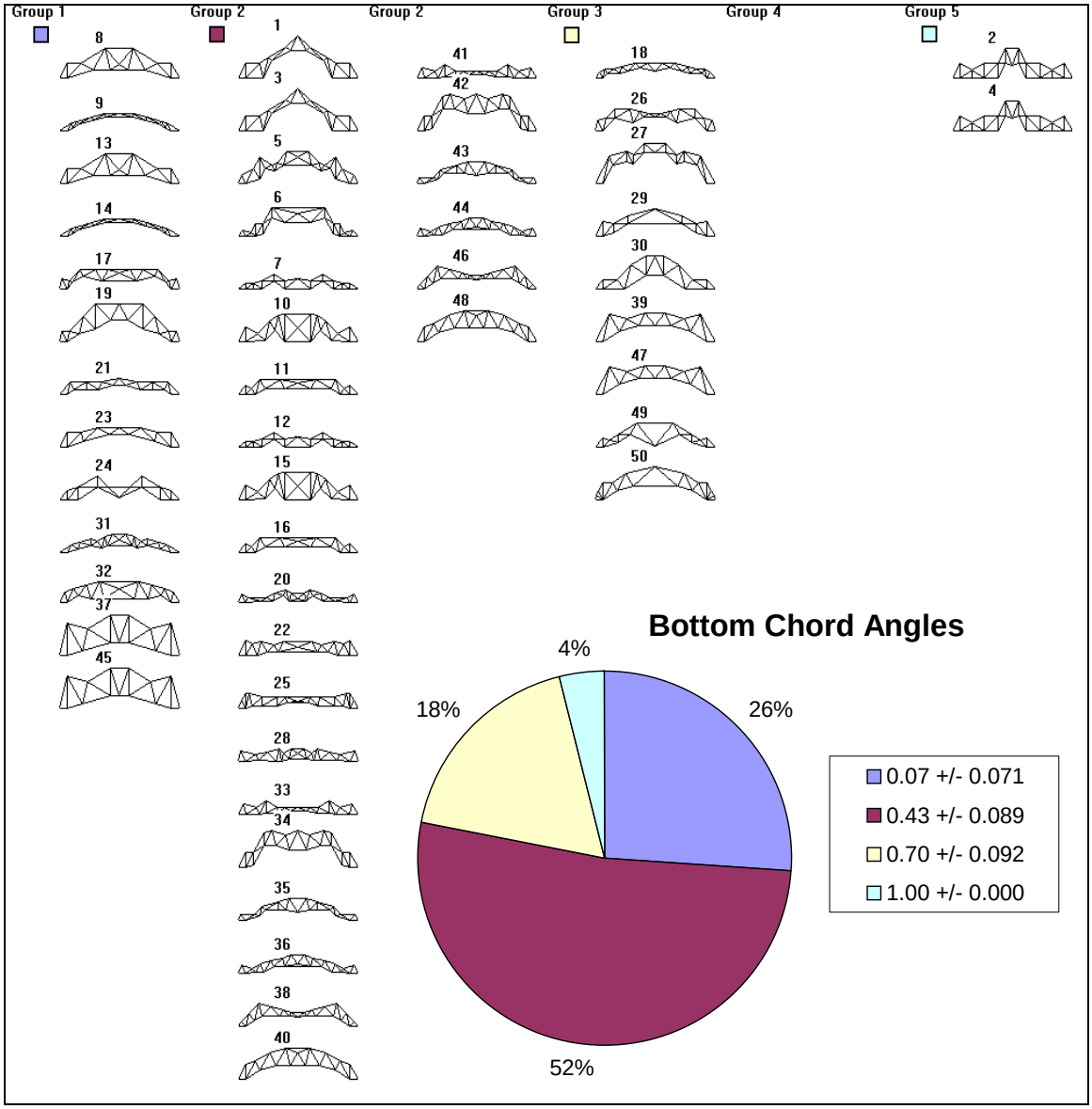


Figure 2.26. Feature map created for S50M50 with bottom chord angles as input.

Like its top chord counterpart, the sum of bottom chord angles encounters stability problems which dictate that no more than five groups may be used in creating a feature map (regardless of population size). This restriction, though not adversely affecting the ability to classify bottom chord shapes, might cause problems when

multiple features are used.

This characteristic does an adequate job of classifying bottom chord shapes for the twenty-five structure populations (see Appendix A), but problems are encountered in larger populations. The feature maps are unable to distinguish when a change in bottom chord angles is significant or nearly indistinguishable. This difficulty is illustrated in group 2 of Figure 2.26, which pairs structures 28 and 34 contrary to human intuition.

Classification performance for the bottom chord angles feature is worse than for the corresponding top chord feature, while suffering from the same stability restrictions. It is therefore unlikely that this feature will add significant information to aesthetic identification.

Top Chord Direction Changes

Two previously discussed measures of top chord shape, concavity and dip, were based on simple, binary decisions. While they performed well, these decisions required data from only one top chord node. Therefore, it seemed advisable to consider features that would more adequately reflect the shape of the entire top chord, rather than a single region of it. One alternative feature considered was the number of direction changes along the top chord.

Height differentials were compared for adjacent top chord nodes. A positive height differential indicated that the truss was growing taller as it approached mid-span, and a negative differential correlated with a decrease in truss height. A simple tally of how many times height differentials changed along the top chord was recorded. Figure 2.27 illustrates results for population S50M50.

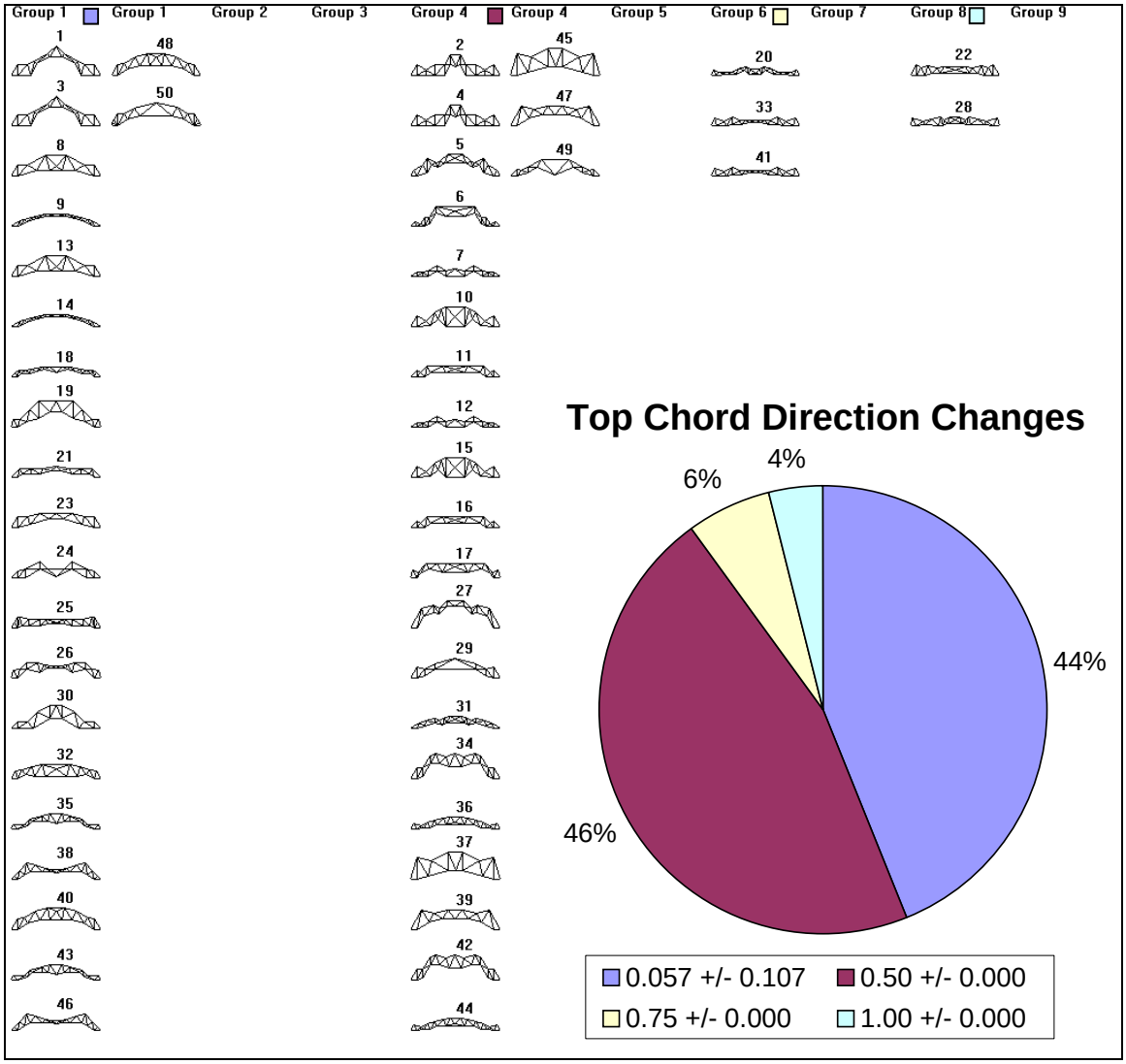


Figure 2.27. Feature map created for S50M50 with number of top chord direction changes as input.

This feature appears to have a strong effect on classification. It identifies whether a top chord dip exists and also explores the issue of concavity. In addition, the number of top chord direction changes separates trusses based on top chords intricacy. Generally, few groups are identified, but these groups are visually distinguishable.

In Figure 2.27, structures 24, 26, 38, and 46 all have one dip along their top chord but are grouped with trusses having no dips. This feature map is the only one in which the KSOM has not completely separated no-dip and one-dip trusses, and the low number of one-dip trusses might account for the strange grouping. Overall, however, the number of top chord direction changes seems to be a useful classification attribute. KSOMs using this feature also appear capable of reproducing the top chord dip and top chord concavity characteristics.

Bottom Chord Direction Changes

The number of direction changes among bottom chord nodes was also evaluated as a potential classification feature. Figure 2.28 presents KSOM results for population S50M50 using this characteristic.

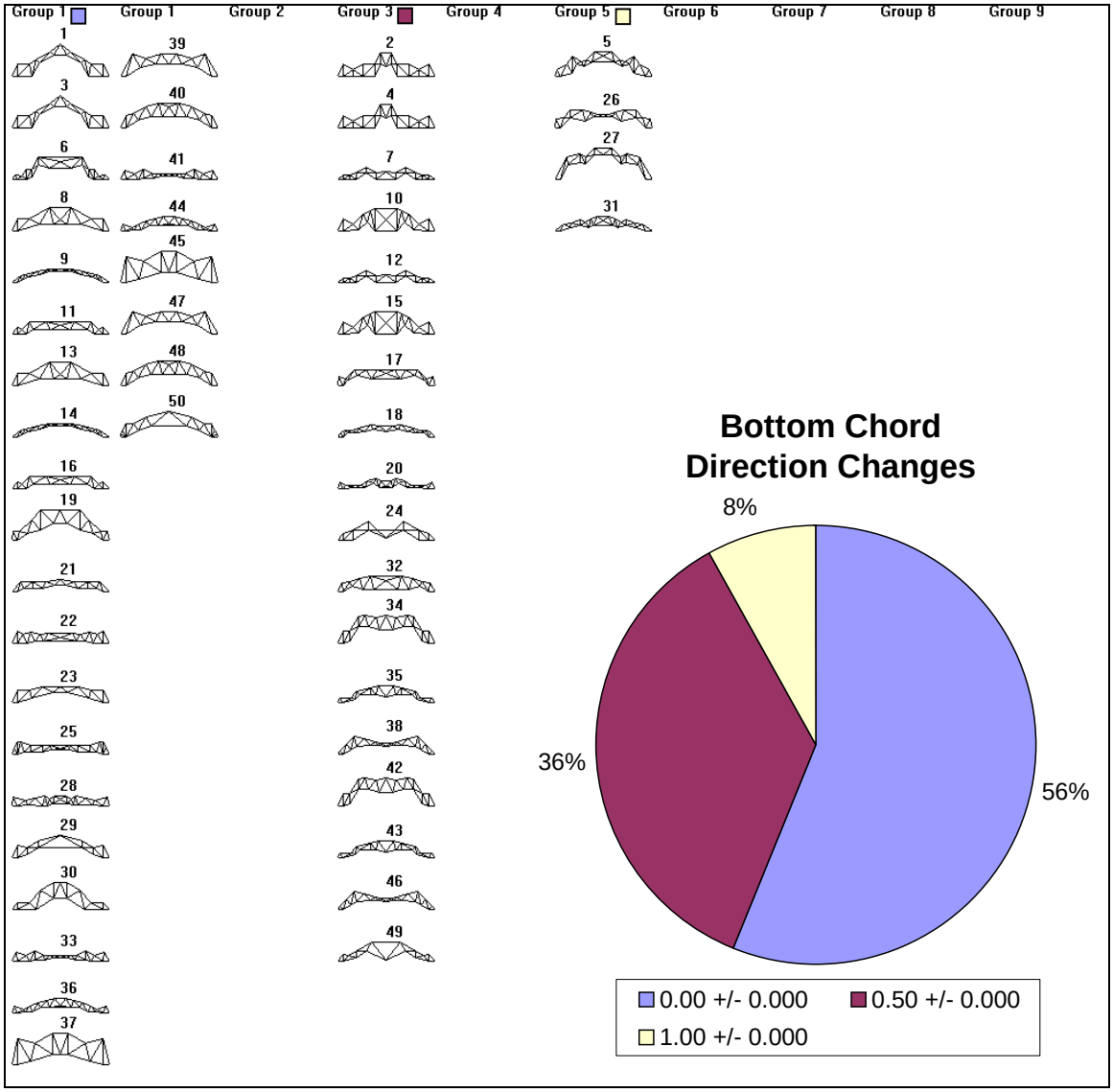


Figure 2.28. Feature map created for S50M50 with number of bottom chord direction changes as input.

As with the previous top chord characteristic, this feature clearly functions well. It indicates whether heights along the bottom chord increase monotonically or if the bottom chord experiences significant direct changes. Unlike the top chord description, however, the bottom chord direction changes do not appear valuable in truss

classification. No more than three groups are identified in each population, and most populations are dominated by one (or at most two) of these groups. Therefore, it seems as if the number of bottom chord direction changes does not offer a significant insight into truss appearance.

Truss Depth

Trusses with the same height and/or mid-span clearance generally have similar shapes. However, some trusses that may otherwise resemble each other will differ in how deep or narrow they are. Some trusses appear intricate not because they have more members or joints than their fellows but because these joints and members are placed close together. In other words, the height difference between the top and bottom chord nodes may significantly impact truss appearance.

An attempt to quantify this idea of truss depth was made by calculating a height differential between truss chords. For each joint along the top chord of a truss, the height of the bottom chord at the same x-coordinate was measured. This height was found using bottom chord coordinates. Whenever a bottom chord node lay at the same x-coordinate as a top chord node, the height differential was computed directly; otherwise, height was linearly extrapolated between the two nearest bottom chord nodes. A sum of these height differentials was then divided by the number of top chord nodes in order to normalize the measure. The results of this calculation on population S50M50 are shown in Figure 2.29.

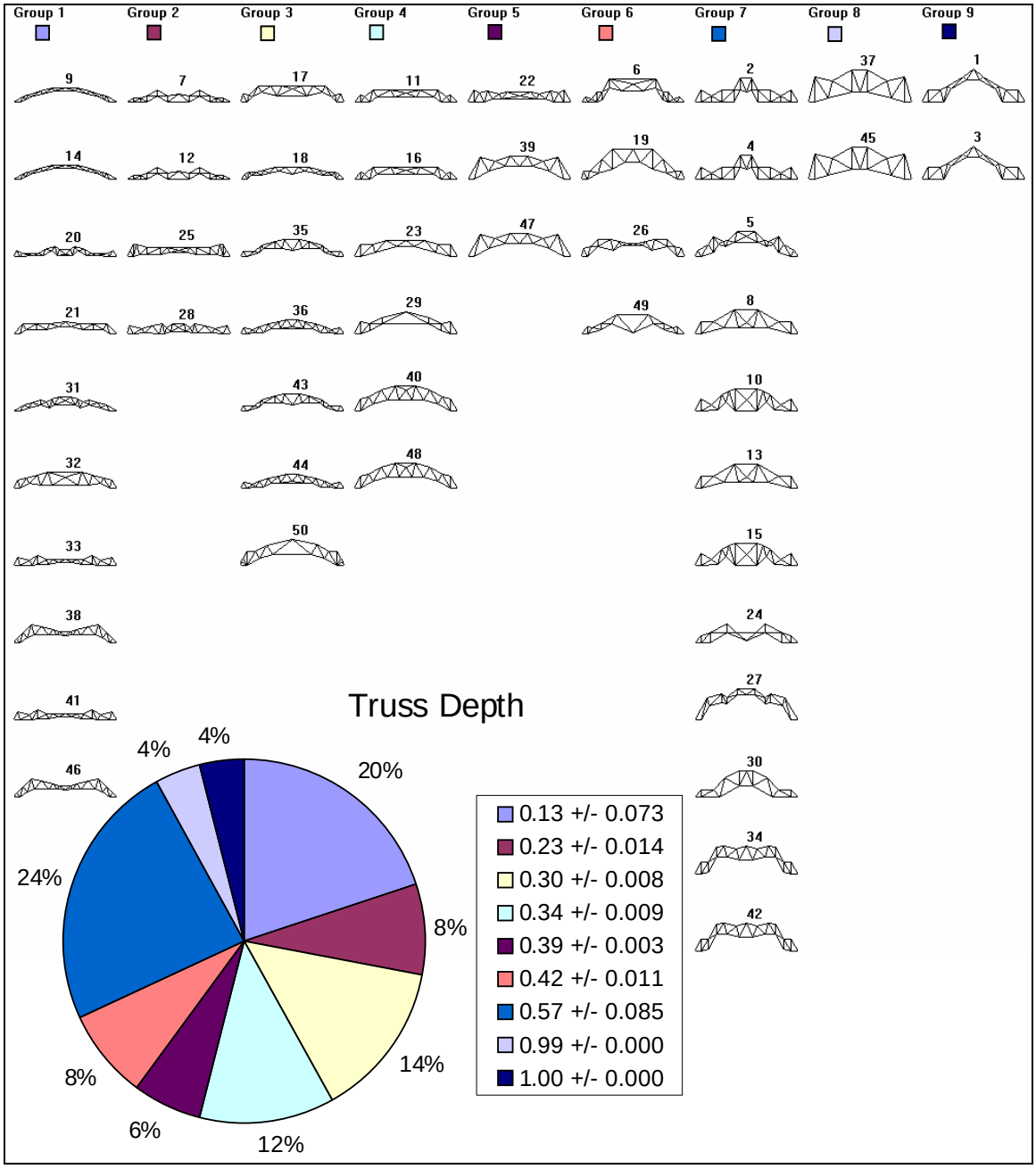


Figure 2.29. Feature map created for S50M50 with truss depth as input.

In most cases, truss depth creates a visual spectrum from the narrowest to the deepest trusses. Classification based on truss depth is not perfect, however; trusses with

a large number of diagonal members will register as deeper than intuition would dictate. This "false" sense of depth occurs because the height differential references global x- and y-coordinates, and the diagonal members draw the eye into a local, skewed orientation. Despite this local difficulty, truss depth makes an important distinction between truss shapes on a global scale.

Feature Groups

Of the twenty potential truss characteristics initially proposed, the preceding investigation identified fifteen that merited further examination. These fifteen features were combined into feature sets, and KSOM results for each set were plotted in three dimensions. The proposed sets attempted to describe the following phenomena:

- General truss shape
- Truss simplicity
- Top chord shape
- Bottom chord shape
- Overall chord shape

Since results did not vary significantly across populations, these three-dimensional plots will be shown alongside their corresponding feature maps for the S50M50 population only. Points in these figures represent the location of a truss relative to the sets' three component characteristics. Different colors are used to represent the different truss groups identified by the KSOM.

General truss shape was the first characteristic considered. Figure 2.30 illustrates the general shape feature set as defined by truss height, mid-span clearance, and the

number of top chord nodes. While the graph shows closely spaced group members, the clusters are not as well delineated as desired. Similarly, the accompanying feature map seems to most clearly reflect the truss height and mid-span clearance features.

The second attempt to capture general truss shape retained the characteristics of maximum height and mid-span clearance. The number of top chord nodes was replaced with the truss depth measurement, and the result is illustrated in Figure 2.31.

Replacing the number of top chord nodes with a measure of truss depth appears to be an effective alteration to the general shape feature set. Overall, the KSOM-generated classification shown in Figure 2.31 makes sense visually. There is a progression from highly arched topologies with narrow spaces between top and bottom chords, to high arches with a larger averaged depth, and finally to flatter, less-arched topologies that tend to be fairly narrow. The three-dimensional plot of individual trusses also shows clearer definition between groups. Therefore, it appears that the height, clearance, and depth features can be used to successfully distinguish truss shape.

It was also desirable to capture the idea of truss simplicity. This feature set was comprised of the number of members, number of nodes, and average joint connectivity characteristics. As is illustrated in Figure 2.32, these characteristics were able to successfully show variations in truss simplicity. The feature map presents a clear progression from simple to complex trusses, and the groups displayed in the three-dimensional plots are clearly separable. Note that in Figure 2.32, average nodal connectivity has been normalized to lie between 0 and 1.

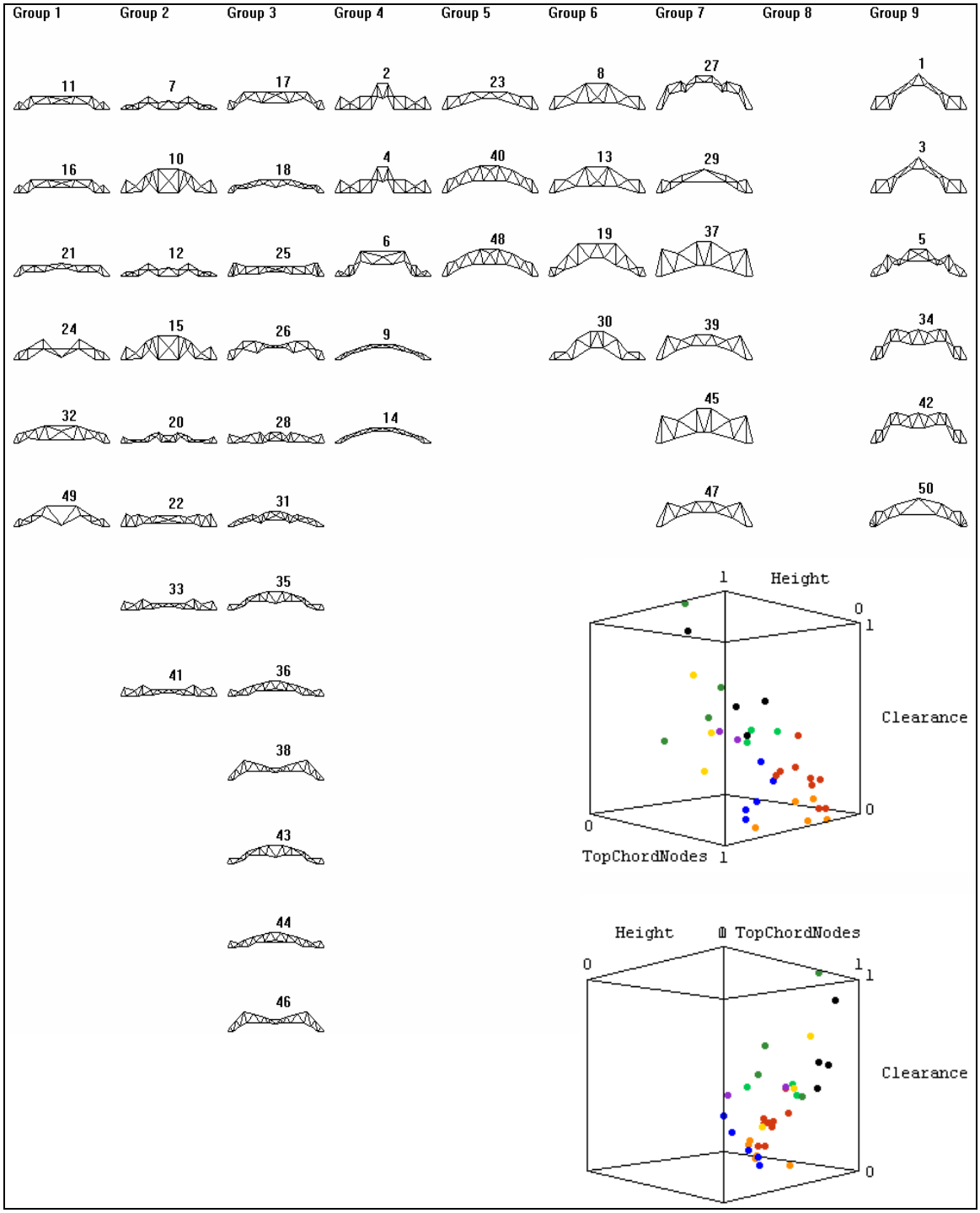


Figure 2.30. General shape feature set for S50M50 using height, clearance, and top chord nodes.

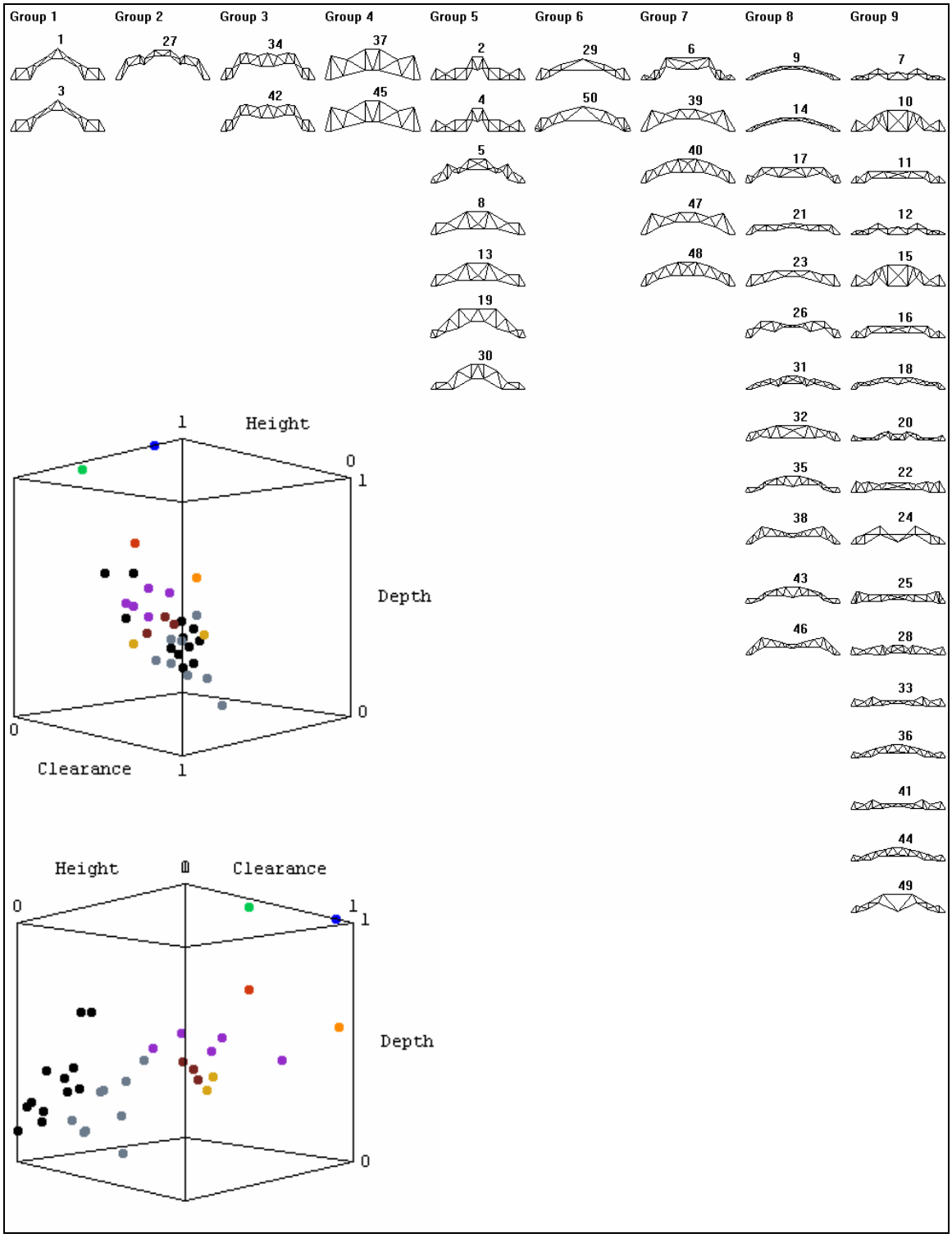


Figure 2.31. General shape feature set for S50M50 using height, clearance, and truss depth.

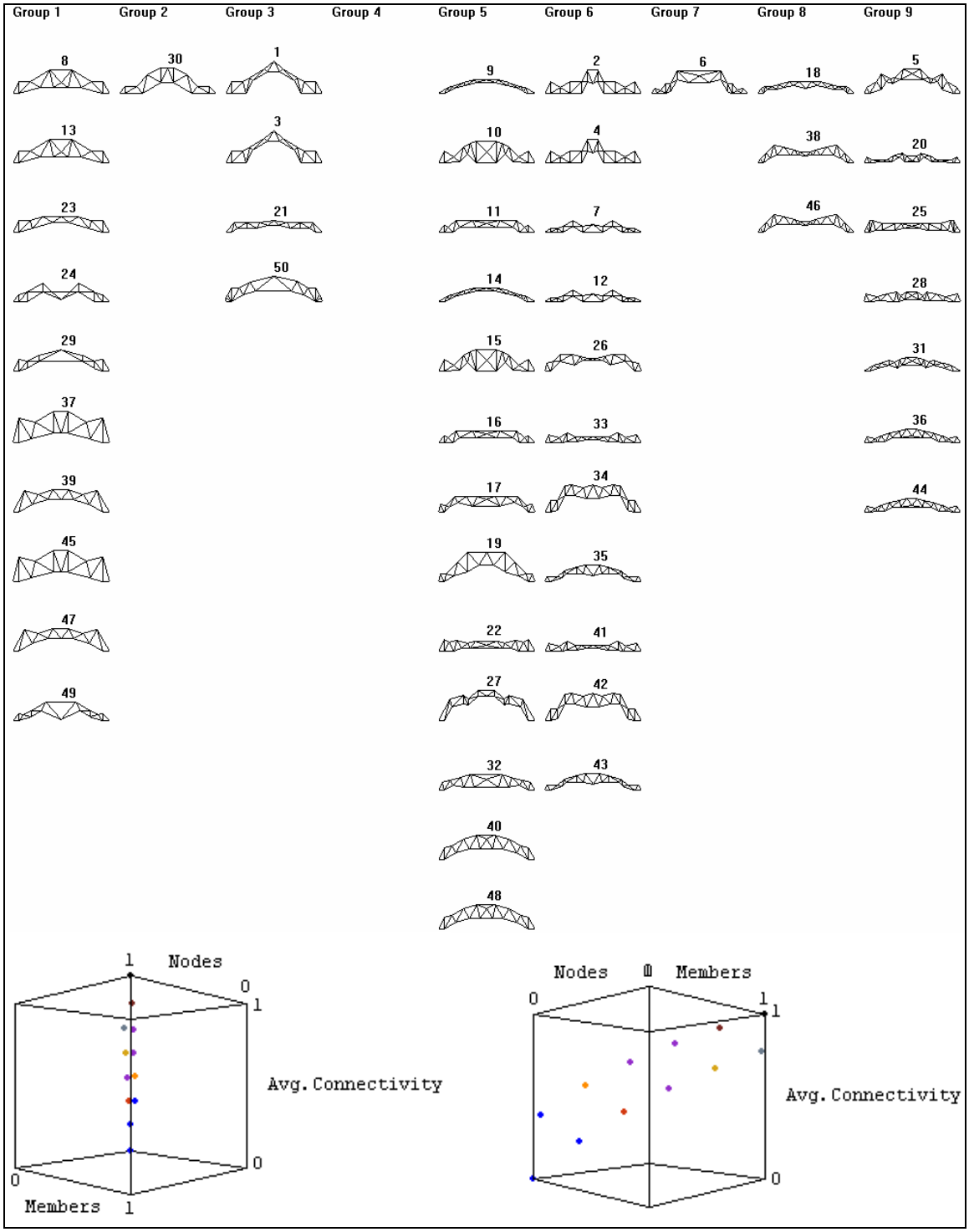


Figure 2.32. Truss simplicity feature set for S50M50 using number of nodes, number of members, and average joint connectivity.

The last three feature sets represent an attempt to capture the shape of both the top and bottom truss chords using different feature combinations. The first feature set, aimed at capturing top chord, used the binary variables defined for top chord concavity and top chord dip as well as the original measure of top chord flatness. Results for population S50M50 using these three characteristics are shown in Figure 2.33. While these three characteristics all appeared to effective on an individual scale, the three-dimensional plot shows that concavity and dip tend to dominate classification. Top chord flatness caused a minor "spread" among groups primarily determined by the dip and concavity characteristics. The KSOM groups for this feature set also lacked clarity.

A second top chord shape feature set replaced the top chord dip and concavity characteristics with the number of top chord direction changes. In order to provide a third dimension, this set also included top chord angularity. A similar feature set was created for the bottom chord. However, as mentioned earlier, top and bottom chord angularity were questionable characteristics. They created instabilities in the KSOM and added no additional information to the top and bottom chord flatness measurements.

Therefore, it was decided to unite features describing both the top and bottom chords into a single feature set. The characteristics used in this set included the number of top chord direction changes and measures of both top and bottom chord flatness. Results are illustrated in Figure 2.34. This feature set successfully combines the most descriptive characteristics of both top and bottom chord behavior. The three-dimensional graph shows tightly clustered groups that are easily distinguished. Moreover, the KSOM feature map presents an aesthetically logical classification.

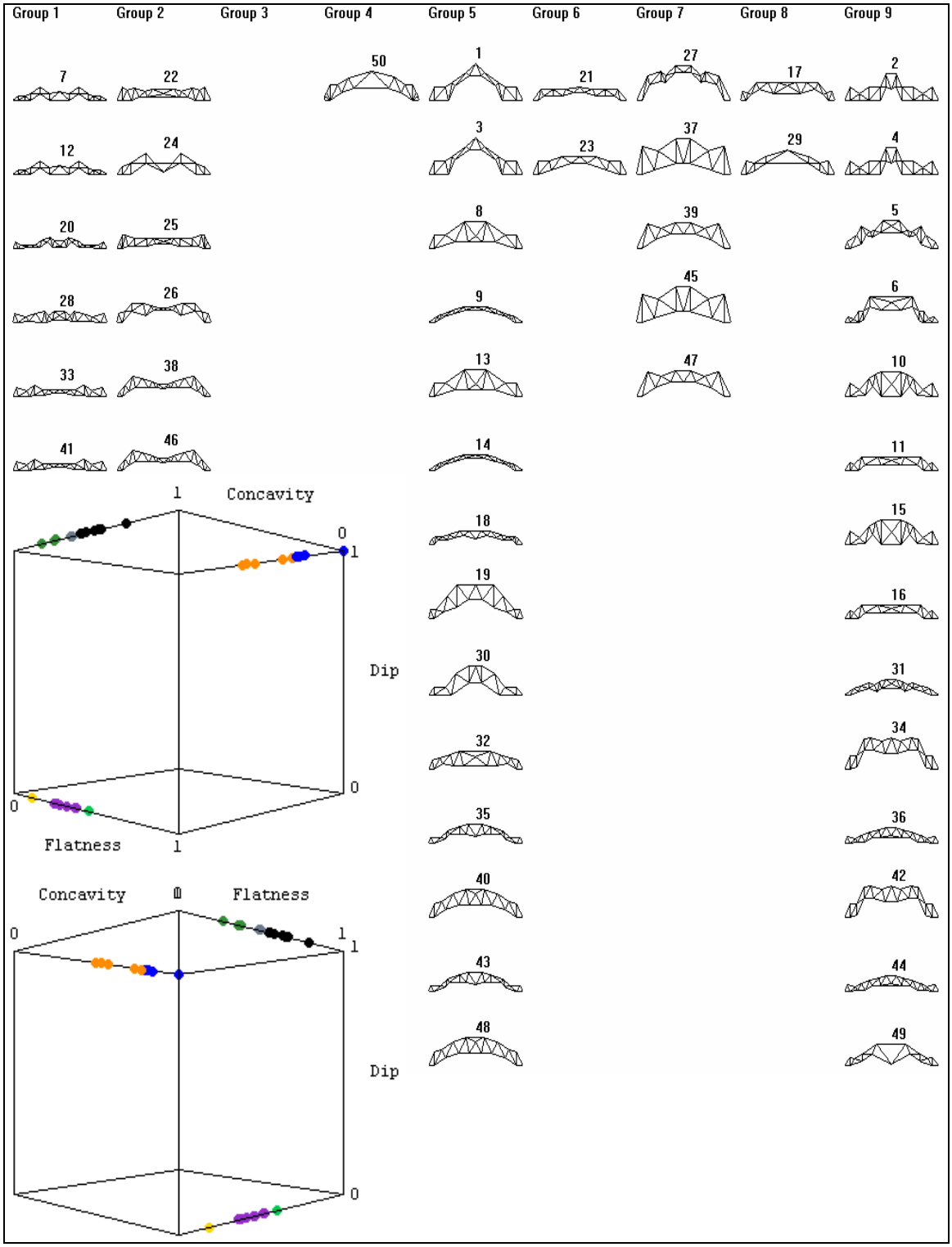


Figure 2.33. Top chord shape feature set for S50M50 using concavity, dip, and top chord flatness.

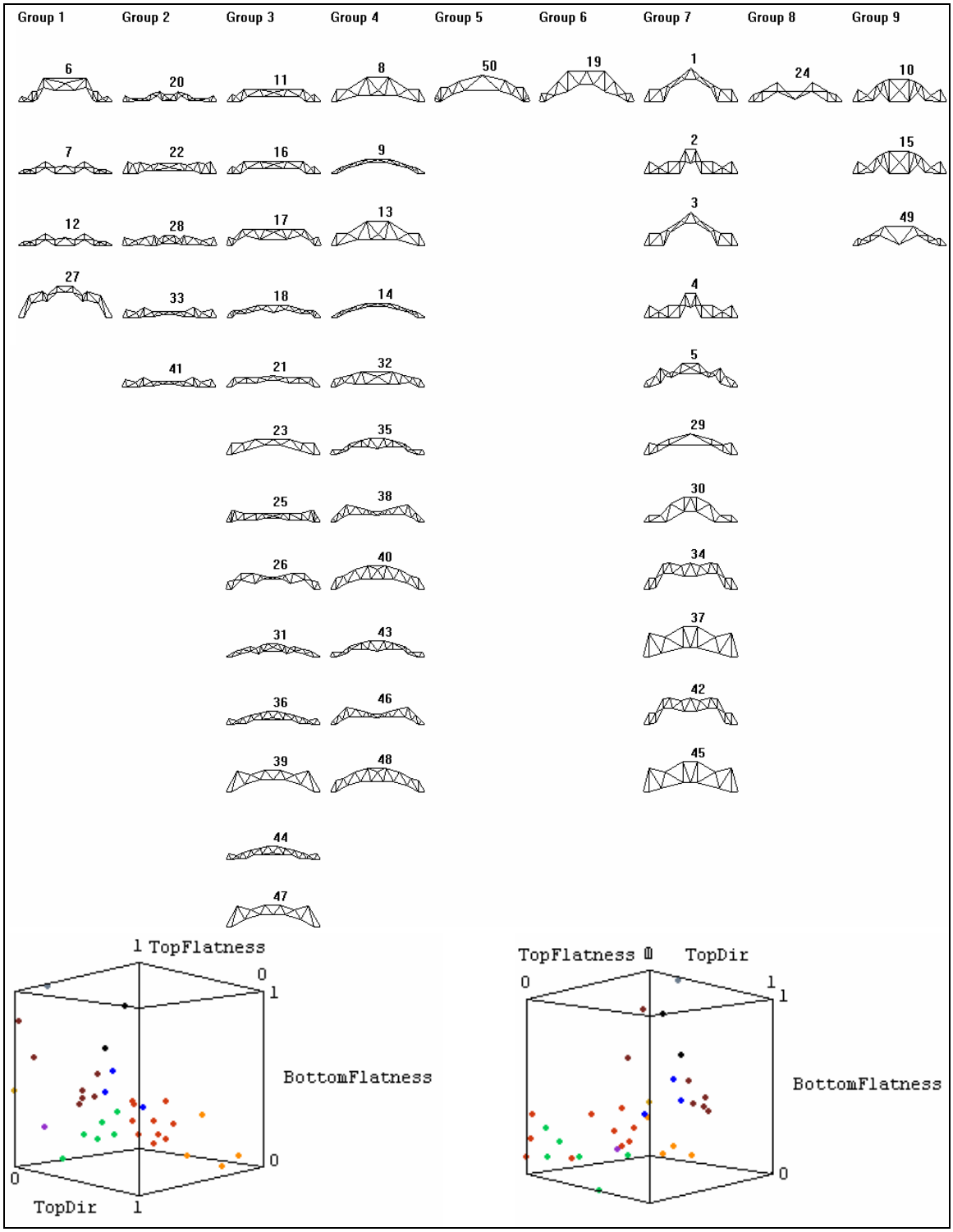


Figure 2.34. Chord shape feature set for S50M50 using number of top chord direction changes, top chord flatness, and bottom chord flatness.

Final Feature Selections

Of the twenty features originally proposed, the nine most significant descriptors of truss behavior were selected. These quantities form the basis of a characteristic feature vector that creates a discretized view of individual trusses. This feature vector will be used as input both to create meaningful truss classifications and to detect user preferences. Furthermore, when combined into feature sets, these characteristics were able to represent fundamental ideas about the general shape, simplicity, and chord shape of a truss. Table 2.2 summarizes these sets and their constituent features.

Table 2.2. Summary of Feature Sets for Truss Discretization.

General Shape	Simplicity	Chord Shape
Maximum truss height	Number of members	Top chord direction changes
Mid-span clearance	Number of joints	Top chord flatness
Truss depth	Average joint connectivity	Bottom chord flatness

The feature sets were further combined to form a three-dimensional representation of the clustering process with all nine features active. A normalized length was calculated for each feature set. These lengths were then treated as ordinates, so that each set of three characteristics then became one of three axes for plotting the combined length of these characteristics.

The feature map generated by the 1D KSOM for the complete feature vector, as well as the corresponding three-dimensional graph, is shown in Figure 2.35 for population S50M50.

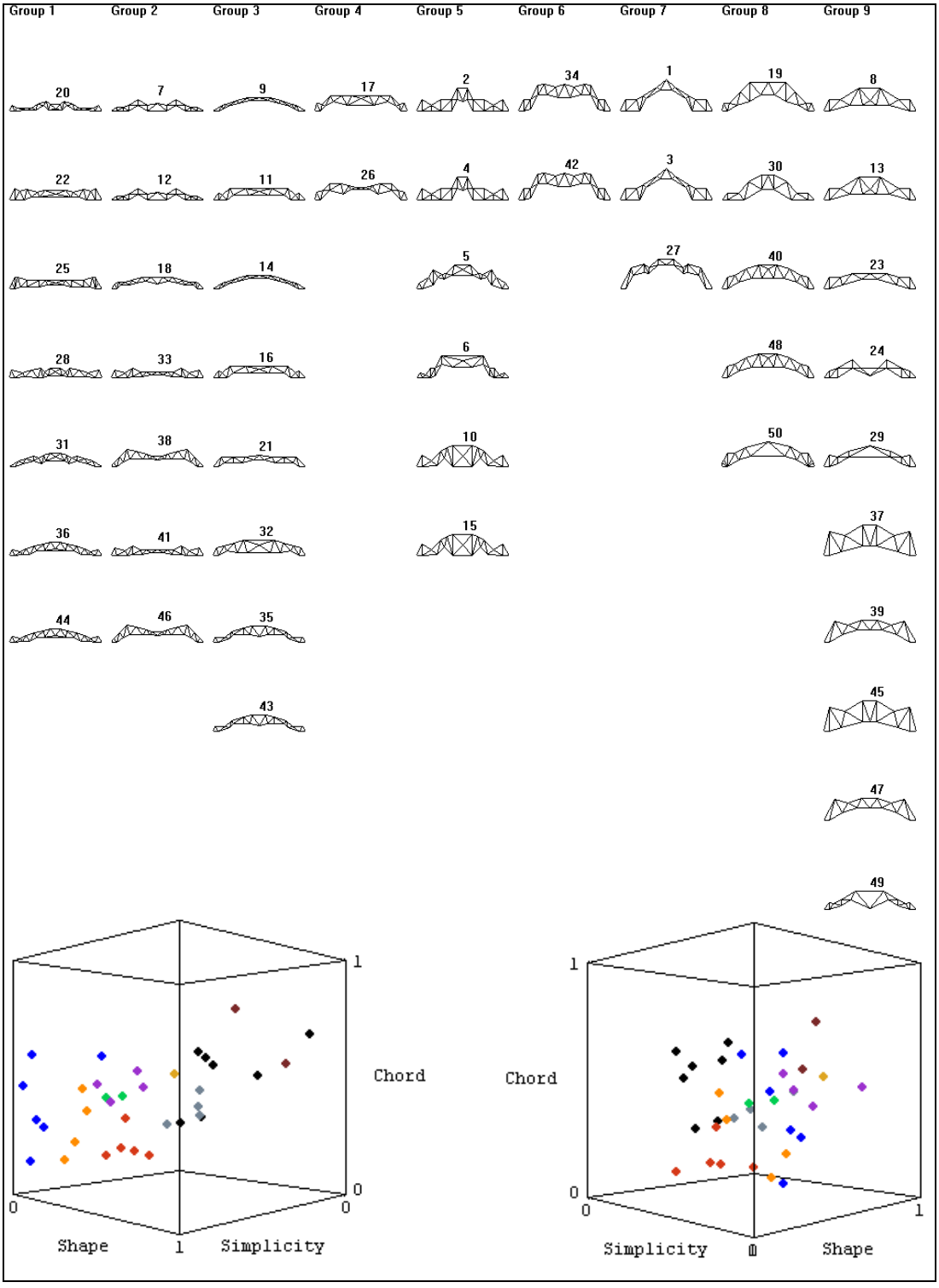


Figure 2.35. Feature map for population S50M50 using complete feature vector.

CHAPTER III

CLASSIFICATION MECHANISM

Before a user's aesthetic preferences can be incorporated into the optimization process, a mechanism for identifying these preferences must be in place. Determining a characteristic feature vector to describe an individual truss was the first step in developing such a mechanism. Now, an efficient method for capturing user evaluations is desired.

Many interactive evolutionary computing (IEC) methods that attempt to capture human evaluations will directly query the user for all feedback. This practice creates a "physical and psychological burden" on the user (Takagi and Ohya 1996). Indeed, human fatigue has been identified as the biggest barrier facing practical implementation of IEC programs (Takagi 1998). In cases where the user's evaluation directly corresponds to the fitness assessment of GA individuals, smaller population sizes may be used to decrease the number of required evaluations. These smaller populations correspond to a restricted search space (Takagi 1998). In the current multi-objective environment, already significantly constricted because of practical requirements on allowable trusses, further restriction of the search space would be severely detrimental to the creation of viable design alternatives.

Therefore, it is important to limit the number of required human interactions without truncating the search operations of the IRR GA. Two previous IEC studies provide information on how to accomplish this goal. Both studies asked users to

evaluate drawings of different facial expressions; this evaluation formed the entirety of the GA fitness function. In Ohsaki and Takagi (1998), human users assigned an actual fitness value to each expression. A neural network was then used to predict a fitness for new expressions, and facial expressions were sorted in order of predicted fitness before being presented to users. From a human point of view, however, this ordering did not matter, since people tended to focus on only one part of the expression at a time, rather than consider the picture as a whole. Many of the users in this study said that "...they first categorized the displayed individuals into several groups, e.g., similar and dissimilar groups, and then evaluated them in detail" (Ohsaki and Takagi 1998).

The importance of grouping on human evaluation is borne out in the next study, as well. Takagi and Ohya (1996) asked users to perform "continuous" and "discrete" evaluations of facial expressions. In the discrete evaluations, similar objects were grouped together, and the user would make one fitness assessment for the group as a whole. This process differed from continuous evaluations, where users assigned fitness values to the facial expressions on an individual basis. The researchers concluded that the discrete evaluation technique was easier for users and showed faster convergence than its continuous counterpart (Takagi and Ohya 1996).

Therefore, the idea of classifying similar objects before asking for a user's evaluation appears sound. As mentioned earlier, grouping truss designs by their aesthetic similarities will reduce the number of evaluations a user must make while simultaneously increasing the information available to form judgments about a user's future preferences. Both of these tasks are important to reducing overall human fatigue.

Since the population of truss designs will vary with each application of the IRR GA, it was important to determine a mechanism capable of independently detecting similarities between trusses. This mechanism would then use the perceived similarities to create an appropriate classification of the population. Some thought must be given to the selection of an automatic classification method, however, since the categorization of objects affects the way they are perceived by human users (Pothos and Chater 2002). The introduction of a category boundary may make distinguishing differences between objects in different classes more difficult and lead to confusion (Pothos and Chater 2002). Therefore, a variety of unsupervised clustering algorithms were considered for this task, including the following:

- Kohonen's self-organizing map (KSOM)
- K-Means
- Nearest Neighbor

Background Information on Unsupervised Clustering

A review of recent clustering applications led to the selection of these three methods for consideration in the truss classification task. Some of these studies will be discussed here. Background information on each of the classification methods will follow.

The KSOM algorithm is a popular data-mining tool with many classification uses. Many of the original KSOM studies focused on language applications. This algorithm was used to recognize phonetic units and organize them based upon the similarity of the sounds they made (Kohonen 1988a). Schyns (1991) used KSOMs to

model how children acquire language. KSOM-based categories were used to differentiate between cats, birds, and dogs, as well as to form sub-categories of a particular type of bird, etc. (Schyns 1991). The KSOM has also been used to classify seismic properties of potential oil well locations (Taner et al. 2001). One IEC application used self-organizing maps to group individuals in a GA population; users would then select regions where they thought higher performing individuals would be located (Hayashida and Takagi 2000).

The nearest neighbor and k-means algorithms are classical clustering techniques and were initially brought to the investigator's attention through studies comparing them to KSOM clustering. An early paper presents KSOMs as an alternative to standard statistical and clustering methods and uses k-nearest neighbors as a point of comparison (Kohonen 1982). More recently, KSOMs and a hierarchical, agglomerative clustering algorithm (similar to nearest neighbors) were both used to define clusters for lipstick and perfume types based upon chemical composition (Goodacre et al 1999). The hierarchical clustering method was considered a "conventional" approach and served to verify performance of the KSOM (Goodacre et al 1999). Becker (1996) considered several different unsupervised neural networks for classification applications and equated the KSOM, without the use of a neighborhood to adapt the weights of nearby neurons, with k-means clustering. The KSOM was considered to be "easy to implement" and "less sensitive to initial conditions" than other competitive neural networks (Becker 1996).

K-means, nearest neighbor, and KSOM mechanisms are also presented as

standard methods by Pothos and Chater (2002). K-means and nearest neighbor clustering were used to identify geographic features from GIS images (Keyes et al 2003), and both k-means and self-organizing maps were used in a study to create a hybrid classification system from classic unsupervised methods (Wemmert et al 1999). Therefore, it appeared as if these three methods appeared frequently enough in the literature to constitute significant approaches to the classification problem at hand.

Kohonen's Self-Organizing Map

The KSOM was briefly introduced in Chapter II as an unsupervised neural network capable of discovering statistical information contained within a data set and mapping this information onto a topological array of output neurons. Like all neural networks, the KSOM arose from biological observations. Willshaw and von der Malsburg (1976) were some of the earliest researchers to describe how self-organization comes about in the human brain. They used two two-dimensional sheets to represent pre- and postsynaptic nerve cells. These cells were connected to each other by axons, the synapses of which were considered to be plastic in accordance with Hebb's 1949 hypothesis. They believed that the geometric proximity of two cells on either sheet could be used to represent electrical activity between them and that the strength of a cell's firing would correlate to how much adaptation would occur in its synapses. Therefore, "...synapses of tight clusters of presynaptic cells evoking activity in tight postsynaptic clusters will be selectively reinforced" (Willshaw and von der Malsburg 1976). Computer simulations verified that nodal locations could be reproduced between pre- and postsynaptic sheets, leading to the conclusion that "...such theories of self-

organization do in fact work, and they should now be taken seriously" (Willshaw and von der Malsburg 1976).

The discovery of "supraneuronal units of functional organization", such as the cortical column, have since served to reinforce "...the central role of mapping in the nervous system" (Reeke et al 1990). Some researchers believe that the nervous system operates by a sort of Darwinian selection, and that such selective systems are "natural categorization machines" (Reeke et al 1990). This theory states that the basic operating units of the nervous system are local groups of strongly interconnected cells, and that local competition and cooperation among cells are necessary to create local order in the cortex (Reeke et al 1990). More recent work by von der Malsburg (1990) seems to agree. He outlines three principles for organizing systems: fluctuations self-amplify; limited resources lead to competition between fluctuations and selection of the most vigorous; and that fluctuations cooperate and can enhance each other despite competition (von der Malsburg 1990). These principles work together so that "...global order can arise from local interactions" (von der Malsburg 1990). Even though this cooperation and competition occurs on a local scale, self-organization allows for the emergence of global order that makes optimal use of these local rules (von der Malsburg 1990).

The KSOM uses each of the principles outlined above to identify relationships between input data. The following summary and equations, which show how this task is accomplished, are based upon the Haykin (1999), Kohonen (1988b), and Kohonen (2001) texts. As mentioned in Chapter II, the KSOM connects input neurons to an array of output neurons. The weights connecting these neurons are randomly initialized. The

competitive aspect of self-organization is represented in the selection of an output neuron to represent the input. Equation 3.1 summarizes this selection process, with i(**x**) representing the "winning" output neuron for input vector **x**, and $\mathbf{w}_j$ the weight vector connecting the input vector to the j output neurons. As shown, the Euclidean distance is used to determine the output neuron that lies closest to the input vector.

$$i(\mathbf{x}) = \min_j \| \mathbf{x} - \mathbf{w}_j \| \tag{3.1}$$

In accordance with the fact that fluctuations self-amplify, the weight of the winning neuron is adapted so that it becomes more tuned to the input that activated it. The weights of neurons within a prescribed neighborhood of the winning output will also be adapted. This lateral excitation reflects the cooperative aspect of self-organization. Weights are adapted as shown in Equation 3.2 below, which states that the new weight vector for neuron j, $\mathbf{w}_j(t+1)$, is equal to the previous weight vector, $\mathbf{w}_j(t)$, plus the product of the current gain value, $\eta(t)$; the current neighborhood function for neuron j relative to winning neuron i(**x**), $h_{j,i(\mathbf{x})}(t)$; and the difference between the input vector **x** and current weight vector.

$$\mathbf{w}_j(t+1) = \mathbf{w}_j(t) + \eta(t)\, h_{j,i(\mathbf{x})}(t)\, (\mathbf{x} - \mathbf{w}_j(t)) \tag{3.2}$$

The gain and neighborhood functions may take on different forms. The gain term, or learning rate, is a value between 0.0 and 1.0 that governs how fast neural network training occurs. The gain is often monotonically decreased during training, so that the first training steps have the greatest effect on network order, while later training serves to refine the map. The neighborhood function also lies between 0 and 1 and affects whether or not, and to what degree, neurons in the vicinity of the winning output

will be trained. A neighborhood width is generally used as part of the neighborhood calculation. In simple networks, neurons lying within this width will have a neighborhood value of 1 and will be trained, while neurons outside of this width have a neighborhood value of 0 and will not be trained. Neighborhood width refers to the distance between the winning neuron and respective neurons on the output grid. In one-dimensional networks, this distance is defined as the absolute value of the difference between neuron numbers, since all output neurons lie side-by-side. Two-dimensional networks use a neuron's place in the grid to determine the square of the Euclidean distance to the winning neuron. Selecting appropriate constants for gain calculation and neighborhood width generally relies upon a programmer's previous experience with KSOM networks.

K-Means

The k-means algorithm has long been used in clustering applications. It is an optimization algorithm and one of the earliest such techniques to use a "hill-climbing" approach (Everitt et al 2001). Although modifications to basic hill-climbing optimization have been proposed in recent years, including the introduction of the heuristic simulated annealing method, the ease with which the basic k-means algorithm may be implemented has helped to maintain its popularity (Everitt et al 2001).

First, a desired number of clusters is selected, and an initial centroid for each cluster is determined (Spath 1980). The "centroid" is a location in feature space, with each component representing a potential value for the features describing an object to be clustered. Cluster centroids may be defined randomly; by using the feature vectors

describing the first individuals in a population; or by using the feature vectors of the population members that lie furthest from each other (Everitt et al 2001).

The k-means algorithm uses these cluster centroids to partition the data set. The centroid that lies closest to a population member is determined by comparing the square of the Euclidean distance between that member and all existing centroids; the centroid with the smallest distance to the population member is taken as the winner (Spath 1980). Note that this process is equivalent to the one illustrated in Equation 2.1 if the concept of a weight vector is replaced with the n-dimensional centroid location in feature space and the distance is squared.

The population member is moved into the cluster represented by the winning centroid, and this centroid's placement in feature space is updated to reflect the position of the new cluster member (Spath 1980). The centroid's new location is specified by the average feature values for all cluster members. This calculation is shown in Equation 3.3, where $\mathbf{x}_c$ represents the vector defining a cluster centroid, N represents the total number of population members in the cluster, J represents the number of components in the feature vector, and x_{jn} represents the value of population member n for feature j.

$$\mathbf{x}_c = \left(\frac{\sum_{N}^{n} x_{1n}}{N}, \frac{\sum_{N}^{n} x_{2n}}{N}, \ldots, \frac{\sum_{N}^{n} x_{jn}}{N} \right) \tag{3.3}$$

The total error for all of the clusters thus created may be found by calculating the sum-of-squares error between the cluster centroid and individual cluster members (Anderberg 1973). Since the values defining a cluster centroid are updated whenever a

new member is placed in that cluster, it is possible for a cluster centroid to move away from a previously identified cluster member or to move closer to a member of another cluster. Therefore, population members may change clusters in response to new centroid definitions, and the assignment process should continue until the clusters have stabilized (Spath 1980). Each new placement seeks to minimize the total error in the system, so the k-means algorithm should converge to a stable number of clusters (Anderberg 1973).

It is possible for all members to be removed from a cluster, leaving that cluster empty. This situation is equivalent to what happens in the KSOM when output neurons do not lie close enough to inputs to be activated, and therefore are never trained to represent population members. In both cases, the presence of an empty group represents an ability to alter the algorithm, in terms of the number of necessary groups, to better reflect the actual clusters within the input data.

Nearest Neighbor

Like k-means, the nearest neighbor algorithm is a classical clustering technique. It is a hierarchical, agglomerative method-- hierarchical, because it creates a "tree" structure out of the data representing successive stages of comparisons, and agglomerative because the process begins with many partitions and fuses them until a desired number of groups is determined (Everitt 1980). There are many variations on the basic approach to hierarchical clustering, but the nearest neighbor algorithm, also referred to as "single linkage" clustering, is considered one of the most mathematically appealing (Everitt 1980) and easy to program (Everitt 2001).

Nearest neighbor clustering begins by placing each member of a population into

a group. For instance, if 100 trusses were being clustered, the algorithm would begin with 100 unique groups. A matrix is then calculated, with each entry representing the Euclidean distance between two members of the population (Everitt et al 2001). Entries in the matrix are used to determine which two population members are closest to each other; these members are joined into a single group. The distance matrix is then updated to reflect the distance of all population members to the new group; that is, the individual member entries are replaced with a group entry, the components of which are determined by whichever group member lies closest to the population member in question (Everitt 2001). In successive iterations, groups will continue to be joined based on the relative distances of their nearest members.

Unlike the k-means and KSOM methods, the nearest neighbor mapping does not require time to stabilize. The algorithm will end whenever the desired number of groups has been found; if left on its own, nearest neighbors would eventually fuse the entire population into a single group (Everitt 1980). The goal of this algorithm, after all, is to show the *hierarchy* of how all members of a population are related to each other. However, this approach means that the algorithm is not capable of determining how many groups best represent the data set. Discovering the optimal number of clusters for a data set has been a constant problem in clustering analysis (Duda and Hart 1973). Plots of the sum-of-squares error vs. the number of clusters, as well as chi-square and other statistical methods, have traditionally been used to measure "goodness of fit" (Duda and Hart 1973).

A common problem that affects hierarchical methods is called "chaining."

Chaining occurs when clusters are joined because "noise" points, or cluster members that do not reflect the cluster as a whole, lie between them (Everitt et al 2001). Chaining can lead to the formation of huge groups that dominate the clustering map, rather than the smaller, well-defined groups desired (Duda and Hart 1973). Chaining occurs because the nearest neighbor algorithm links groups indirectly, depending on the properties of a single member rather than the entire group (Everitt 1980). Data points joined at one cluster level will remain together for the rest of the analysis, regardless of whether the association continues to characterize the cluster as a whole (Duda and Hart 1973).

Methodology for Selecting a Classification Method

The KSOM, k-means, and nearest neighbor algorithms were used to create clustering maps of five truss populations. The populations are the same as those used in phase II of the feature investigation: S25M35, S25M50, S50M25, S50M35, and S50M50, as illustrated in Figures 2.6, 2.1, 2.7, 2.8, and 2.2. These populations were selected for the classification study because they showed the most meaningful variations in population size and complexity.

In examining the clusters proposed by each of these methods, two important issues needed to be addressed. First, it was necessary to know whether the classifications proposed by any, or all, of these methods would be considered "good" by a human user. If none of the algorithms were able to describe groups within the truss population in a manner that would accord with a human user's intuition, then the classification would not be useful in defining the user's preferences. The second goal of this investigation was to determine which of these three methods would best classify an

arbitrary population. That is, to determine whether or not an algorithm performed better for smaller populations than for larger ones, simple topologies rather than complex ones, etc.

In order to make these decisions, both visual and mathematical criteria were identified. Two subjective measures, similarity matrices and a Rand Index, were used to determine the effectiveness of proposed classifications for S25M35 and S50M35 in terms of human judgment. These populations were selected because they contained a significant number of unique topologies for which comparisons could be easily drawn by human evaluators. Additionally, the sum-of-squares error and standard deviation distributions were examined for all populations to determine the relative effectiveness of the clustering methods from a numerical standpoint.

Similarity Matrices

In the creation of clusters, two basic types of metrics may be used. The first is the distance metric, which all three of the clustering methods described above use. Alternatively, similarity coefficients may be applied. Similarity coefficients measure relationships between two individuals in a population based upon features held in common; different proposals exist for defining these coefficients (Everitt 1980).

Pothos and Chater (2002) used a similarity matrix to illustrate pair-wise comparisons of all objects in a population. The researchers presented volunteers with populations of geometric shapes (starbursts, etc.) and asked them to categorize the shapes. They then asked volunteers to rate how similar objects were on a pair-by-pair basis. Similarity matrices created from these ratings were used to calibrate a simplicity

model for cluster extraction (Pothos and Chater 2002).

In this vein, similarity matrices were created for populations S25M35 and S50M35. These matrices were used as a rubric for measuring how well classifications proposed by the KSOM, k-means, and nearest neighbor algorithms accorded with human intuition. The first step in creating these matrices was to extract the unique topologies from each population and label them as shown in Figures 3.1 and 3.2. The similarities between pairs of trusses in these populations were then recorded. Tables 3.1 and 3.2 show the resulting matrices.

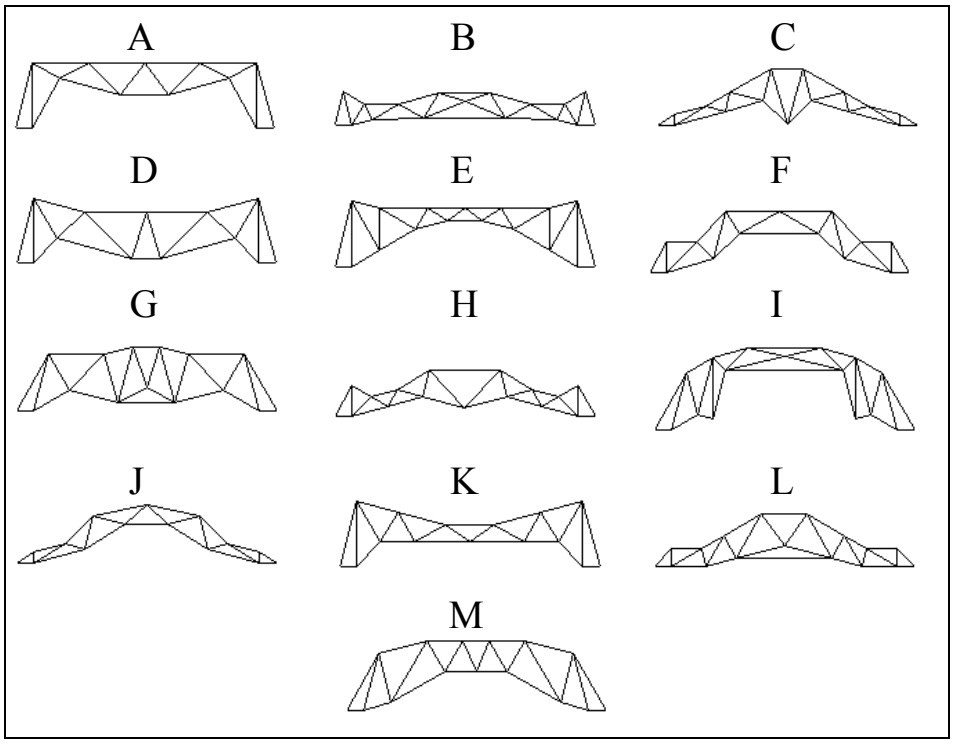


Figure 3.1. Unique topologies from population S25M35.

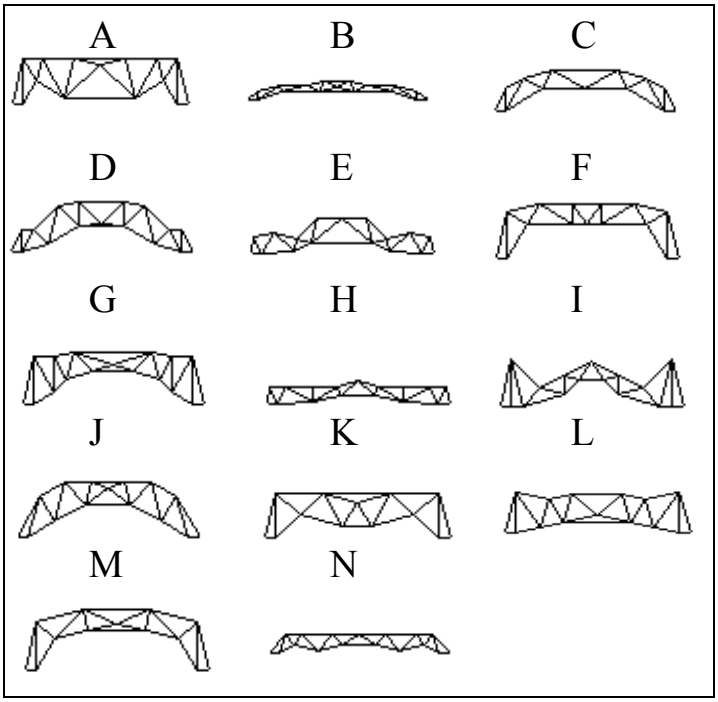


Figure 3.2. Unique topologies from population S50M35.

Table 3.1. Similarity Matrix for Population S25M35.

	A	B	C	D	E	F	G	H	I	J	K	L	M
A	--	1	1	3	3	2	3	1	1	1	2	1	2
B	1	--	1	1	2	1	1	3	2	1	2	1	1
C	1	1	--	1	1	2	1	4	1	3	1	3	1
D	3	1	1	--	3	1	3	1	2	1	3	1	2
E	3	2	1	3	--	2	2	1	2	1	3	1	2
F	2	1	2	1	2	--	2	2	2	3	1	2	2
G	3	1	1	3	2	2	--	1	2	2	2	2	3
H	1	3	4	1	1	2	1	--	1	2	1	2	1
I	1	2	1	2	2	2	2	1	--	2	2	1	2
J	1	1	3	1	1	3	2	2	2	--	1	2	1
K	2	2	1	3	3	1	2	1	2	1	--	2	2
L	1	1	3	1	1	2	2	2	1	2	2	--	2
M	2	1	1	2	2	2	3	1	2	1	2	2	--

Table 3.2. Similarity Matrix for Population S50M35.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
A	--	1	2	2	1	3	3	1	1	2	4	2	3	2
B	1	--	2	1	1	1	2	3	1	2	1	2	2	3
C	2	2	--	2	1	3	3	1	1	3	2	2	3	1
D	2	1	2	--	3	2	1	1	1	3	1	1	2	1
E	1	1	1	3	--	1	1	2	2	2	1	2	1	1
F	3	1	3	2	1	--	3	1	1	3	3	2	4	2
G	3	2	3	1	1	3	--	2	1	3	3	3	3	2
H	1	3	1	1	2	1	2	--	2	1	1	1	1	3
I	1	1	1	1	2	1	1	2	--	1	1	2	1	1
J	2	2	3	3	2	3	3	1	1	--	3	2	3	2
K	4	1	2	1	1	3	3	1	1	3	--	2	3	2
L	2	2	2	1	2	2	3	1	2	2	2	--	2	1
M	3	2	3	2	1	4	3	1	1	3	3	2	--	2
N	2	3	1	1	1	2	2	3	1	2	2	1	2	--

A similarity of "4" identifies trusses that are nearly identical, while a value of "1" represents structures that are considered to be unique. For example, in population S50M35, the only major difference between the trusses labeled "A" and "K" is the fact that truss A is deeper than K, so these trusses are considered to be nearly identical and have the highest possible similarity value. On the other hand, there is no real common ground between trusses A and B, and these structures are considered to be unique.

To convert these similarity judgments into a rubric for assessing classification performance, a point system was created. Grades were calculated for the proposed classifications. Each time two similar trusses were placed in a group together, points were added to the grade; if they were separated, points were subtracted. Conversely, if two non-similar trusses were grouped together, points were subtracted, and points were

added when non-similar trusses were placed in different groups. The number of points rewarded or subtracted for the placement of each truss depended on their similarities, and Table 3.3 shows these values.

Table 3.3. Point Values for Similarity Matrix.

Similarity	Grouped	Separated
4 = nearly identical	+ 4	- 4
3 = similar forms	+ 2	- 2
2 = some resemblance	+ 1	+ 0
1 = unique structures	- 4	+ 4

These point values were selected empirically. Survey data, discussed in the next sub-section, was used for calibration. To put this grade in perspective, consider truss pairs A-K and A-B in the S50M35 population. Since trusses A and K have a similarity of 4, classifications where these two trusses are placed in the same group will receive the highest possible reward, while separating the trusses will results in the maximum penalty. Similarly, placing trusses A and B in the same group will result in the maximum penalty, but the classification will be rewarded for separating these unique structures.

Rand Index Comparison to Survey Data

A second subjective measure of clustering ability was based upon a survey in which nine volunteers were asked to group the unique topologies of populations S25M35 and S50M35. The volunteers had varied backgrounds: non-engineers, structural engineering graduate students, structural engineering professors, practicing

non-civil engineers, and practicing structural engineers. The groups they defined were based solely on their personal perceptions of the similarities between the topologies contained in these two populations.

These surveys were then compiled into a composite clustering map. This composite map grouped or separated truss pairs based upon the placement selected by the majority of the volunteers. That is, if most people grouped trusses A and K of population S50M35 together, they would be placed in the same group in the composite map. If most people placed trusses A and B in different groups, then they would be placed in different groups in the composite map. This map therefore presented an "average" picture of how a human user would classify populations S25M35 and S50M35. Table 3.4 presents a description of these composite clusters, using the truss labels from Figures 3.1 and 3.2.

Table 3.4. Trusses Contained in Composite Map Groups.

Population	**Group 1**	**Group 2**	**Group 3**	**Group 4**	**Group 5**	**Group 6**	**Group 7**	**Group 8**
S25M35	A	B	C, H	D, E, K	F, J, L	G	I	M
S50M35	A, K	B	C, D, J	E	F, G, M	H, N	I	L

A Rand Index was used to determine how similar each of the classifications proposed by the unsupervised clustering algorithms were to the composite map compiled from survey data. A Rand Index is a standard method for measuring agreements between two categorizations (Everitt et al 2001). Agreements and disagreements between the maps are counted for each truss pair, and a ratio of the final tallies is used.

Let A represent the number of times a truss pair is placed in the same group by the two maps, B represent the number of times the pair is grouped by the first map and not the second, C represent the number of times the pair is grouped by the second and not the first, and D represent the number of times both maps place the trusses in different groups. Then, the Rand Index may be calculated using Equation 3.4 (Everitt et al 2001).

$$\text{Rand Index} = \frac{A + D}{A + B + C + D} \tag{3.4}$$

Sum-of-Squares Error

Another common method of determining the efficiency of a proposed classification is to keep track of the sum-of-squares error. This quantity measures the distance between the "center," or average feature vector, of a cluster and its component members and is useful for showing whether of not this distance has been minimized (Spath 1980). In other words, it is desirable for the clusters to be as tightly grouped as possible. A similar measure has been used to measure the quantization error for KSOMs (Cottrell et al. 2001).

For each cluster identified by one of the unsupervised classification methods, an average feature vector was calculated according to Equation 3.3. Equation 3.5 illustrates how this quantity was used in the sum-of-squares error (SSE) calculation. K represents the total number of clusters defined, J represents the number of descriptors used in the feature vector, and N represents how many trusses were placed in group k. The variable x_{njk} refers to the value of the j^{th} feature of the n^{th} member of the k^{th} cluster, and x_{cjk} is the average value for feature j of cluster k.

$$SSE = \sum_{k=1}^{K} \sum_{j=1}^{J} \sum_{n=1}^{N} (x_{njk} - x_{cjk})^2 \quad (3.5)$$

The SSE, as calculated above, was used as a measure of classification efficiency for all five of the truss populations in this study. The lower the SSE value, the more closely aligned members of truss groups were to each other. Likewise, higher SSE values indicate that truss groups contained more disparate members. While not an absolute indicator of within-group truss similarity, the SSE is a useful tool for determining relative similarities between the different clustering methods.

Distribution of Standard Deviations

One final method of exploring the numerical performance of the unsupervised clustering methods was to examine the distribution of standard deviations across the populations. No formal rubric existed for this criterion. Rather, the statistical properties of all five populations were examined. The standard deviations were of particular interest because they, like the SSE, represent how much diversity exists within identified truss groups. Group maximum and minimums, as well as averages, were also considered.

The goal of this examination was to determine how well defined truss groups were. The standard deviations for each feature of each group were tallied and placed within ranges defined by tenths (i.e., 0.00-0.10, 0.11-0.20, etc.). This process allowed for comparisons to be drawn between different clustering techniques and to see how standard deviations varied across the population as a whole. Proposed maps that had many features with small standard deviations were considered highly performing.

To balance these distributions, the number of single topology groups in a clustering map was also tallied. This additional information was necessary because each time a clustering technique isolated a truss design, the standard deviations for every feature of that group would be 0.00. Therefore, the practice of not grouping trusses artificially inflates the number of small standard deviations in the population and should not be uniformly rewarded. In some cases, it is important to isolate trusses that truly diverge from other designs in the populations. However, an excess number of single topology groups indicates a *failure* of the clustering map to correctly subdivide the truss population into groups of *similar design*.

The overall distribution of standard deviations in the population, as well as the number of times a single topology was isolated, provided additional information about how the clustering techniques behaved across different populations. This information was gathered for all five truss populations.

Presentation and Discussion of Results

As stated earlier, the KSOM, k-means, and nearest neighbor algorithms were used to create classifications of populations S25M35, S25M50, S50M25, S50M35, and S50M50 using the nine-element feature vector defined in Chapter II. Both a one-dimensional (1D) and two-dimensional (2D) arrangement of KSOM output neurons were used, so that four clustering maps were created for each population. These clustering maps show how each simulation groups the trusses in a given population. For the sake of clarity, sample maps will be presented for population S50M35; clustering maps for the remaining populations appear in Appendix B.

Figures 3.3 through 3.6 show the truss groups proposed by the unsupervised clustering methods for population S50M35. Truss groups are arranged in vertical columns, with a maximum of twenty trusses per column. Adjacent columns are used to continue groups when necessary.

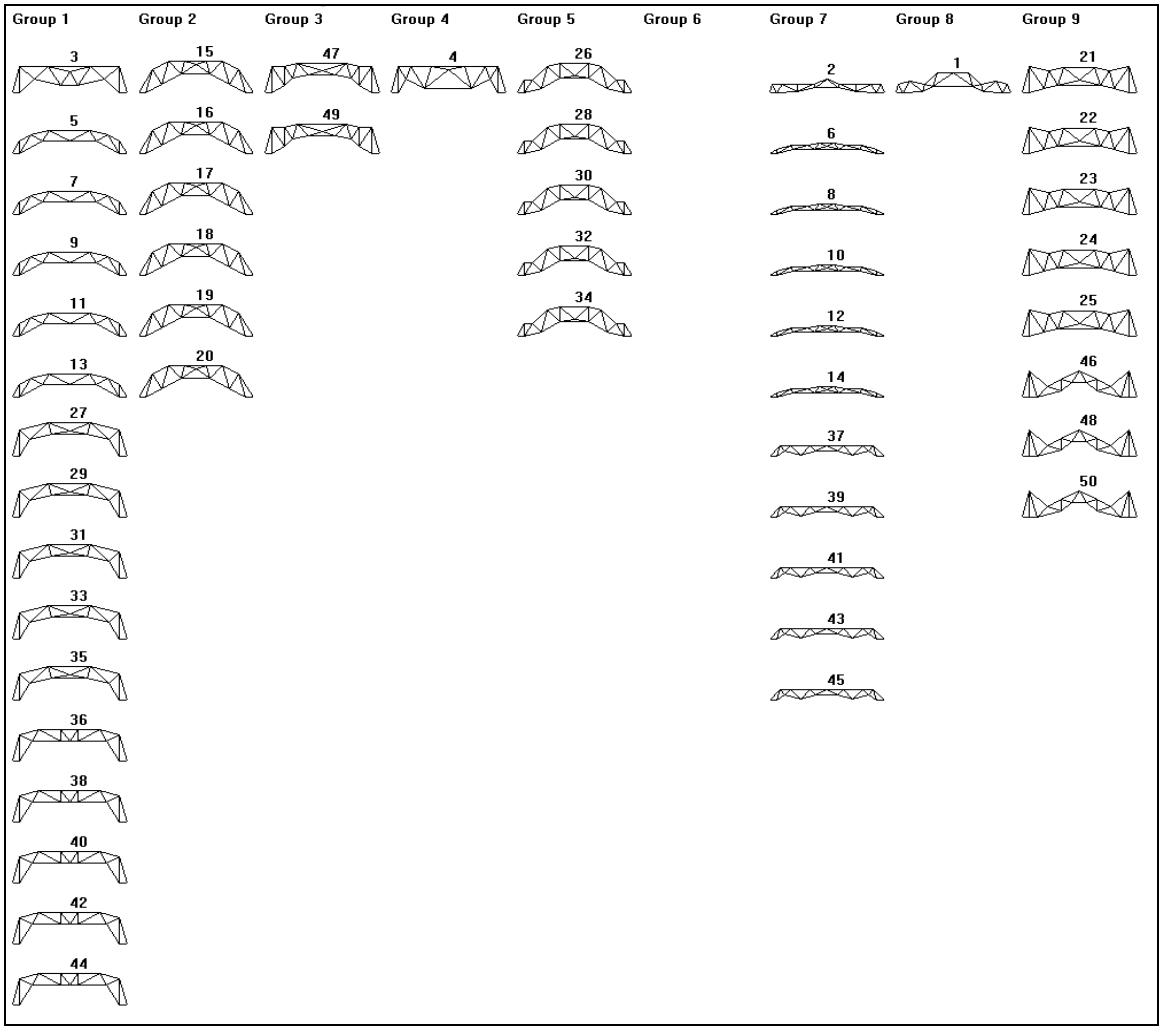


Figure 3.3. Clustering map for S50M35 using 2D KSOM.

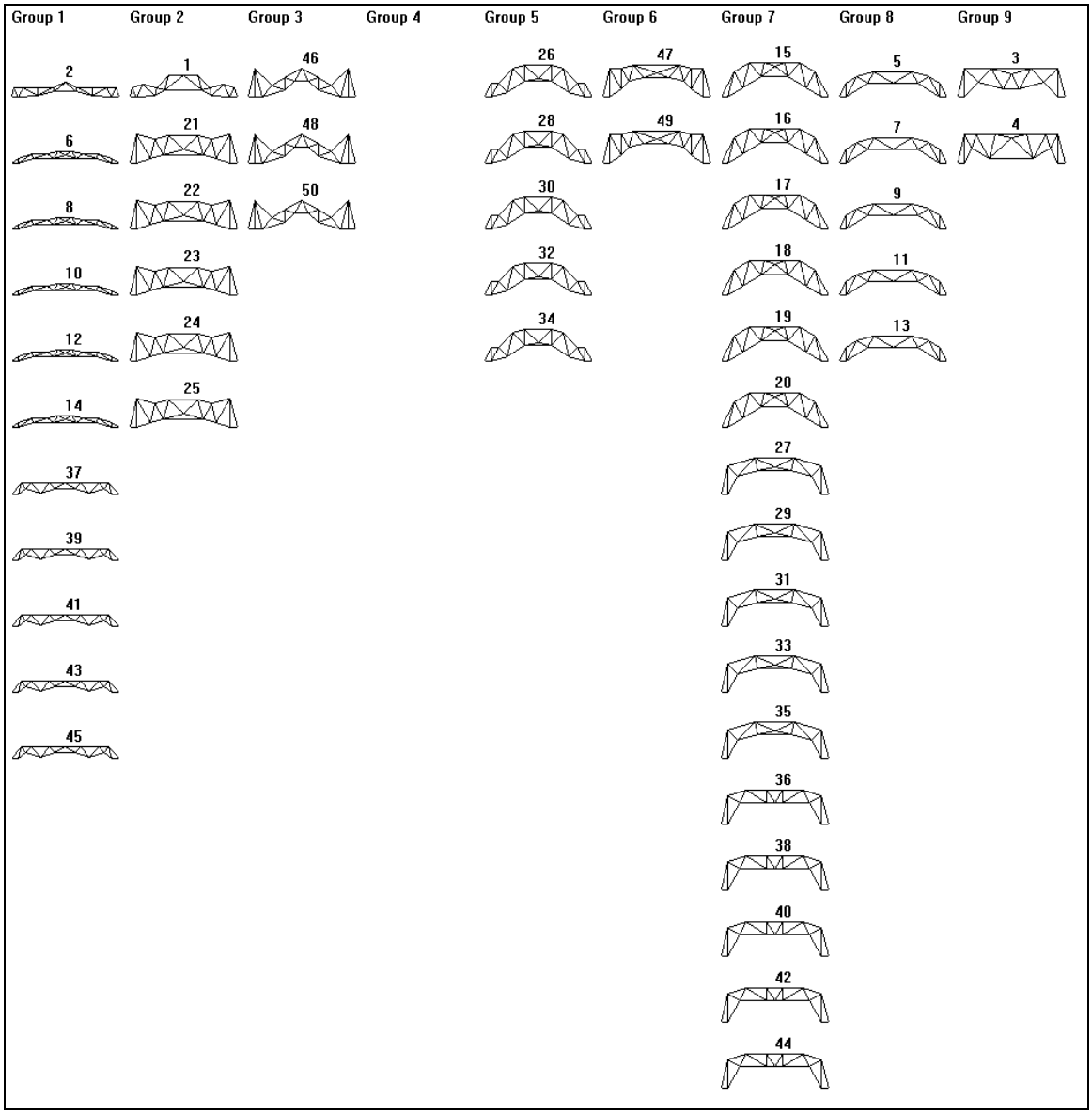


Figure 3.4. Clustering map for S50M35 using 1D KSOM.

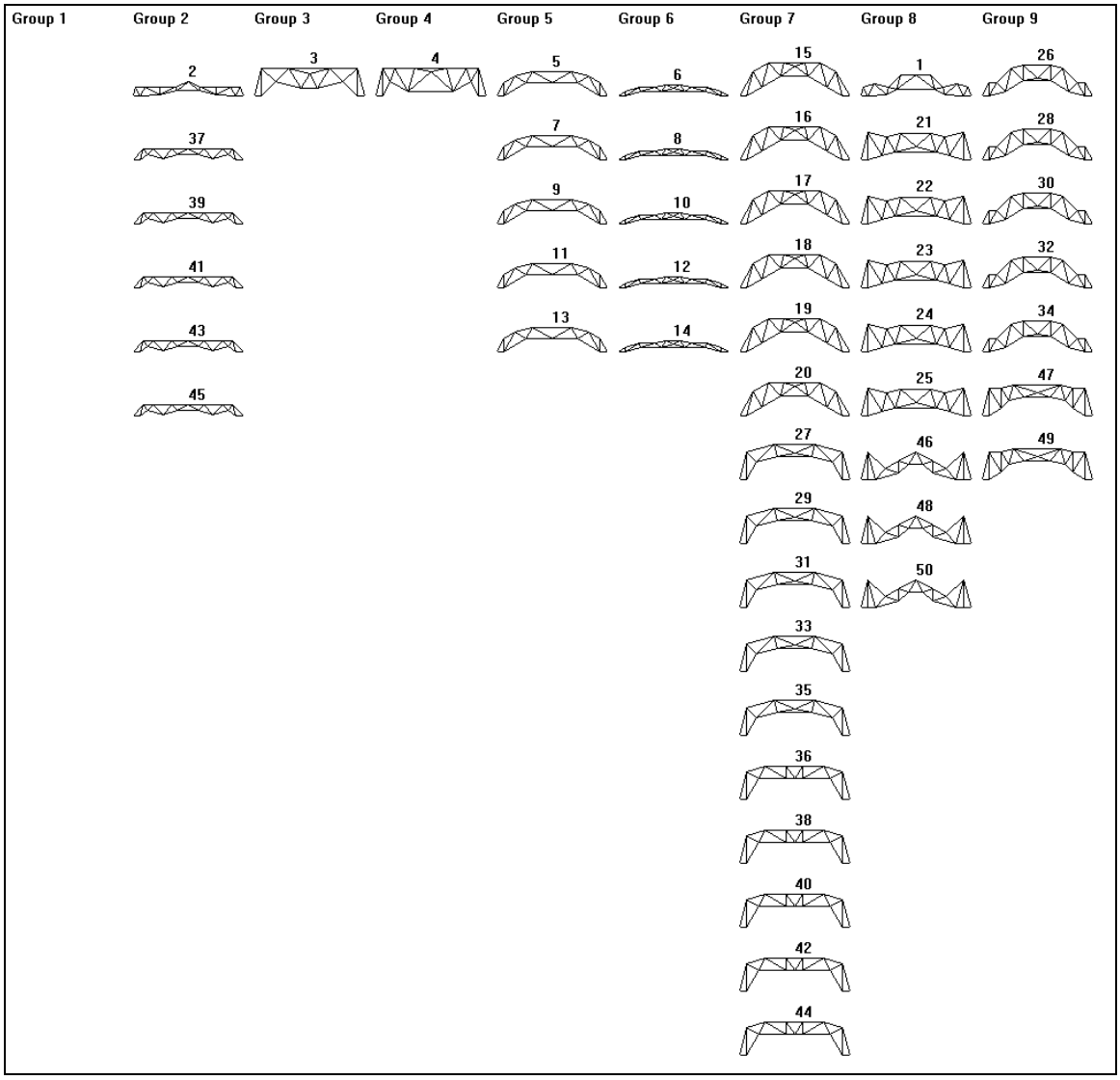


Figure 3.5. Clustering map for S50M35 using k-means.

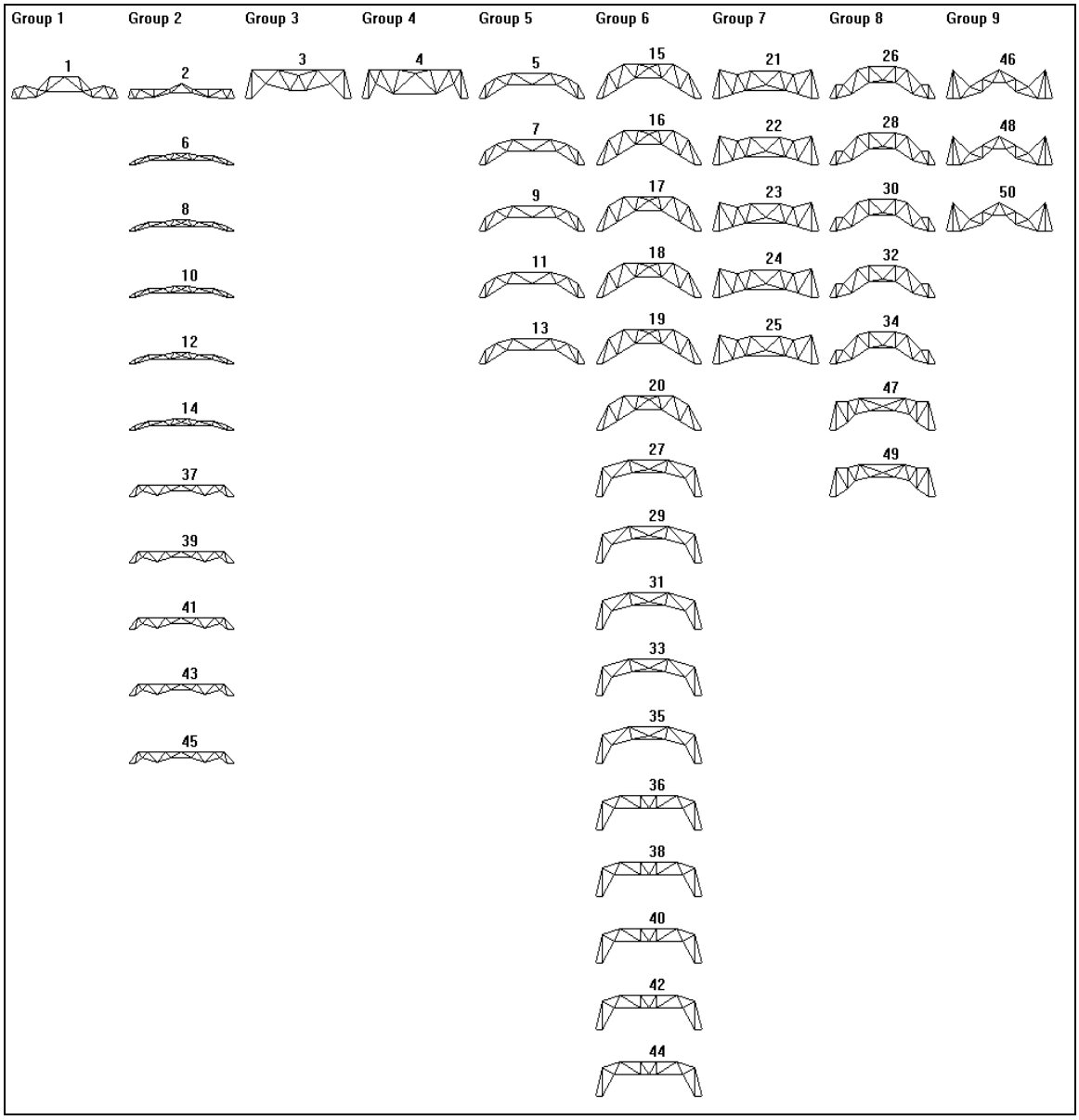


Figure 3.6. Clustering map for S50M35 using nearest neighbors.

The preceding figures illustrate how alternative clustering methods will classify the trusses of population S50M35. A few points should be mentioned about these figures. First, k-means and both KSOM methods have reduced the total number of

groups from nine to eight. This reduction seems to indicate that the topologies of this population are most naturally arranged in eight groups. Additionally, the 1D KSOM is the only map to place structures 3 and 4 (A and K from Figure 3.2) in the same group. Not only was this truss pair considered "nearly identical" by the similarity matrix, but it was the only truss pair from both population S50M35 and S25M35 that *all* volunteers placed in the same group during the survey portion of this investigation. This association, therefore, was an important touchstone for evaluating human perception of the clustering methods.

Tables 3.5 and 3.6 present results for the subjective measures of clustering behavior. These tables show Rand Index and similarity matrix scores for populations S25M35 and S50M35.

Table 3.5. Rand Index Results.

Population	2DKSOM	1DKSOM	K-Means	Nearest Neighbor
S25M35	0.808	0.846	0.833	0.808
S50M35	0.846	0.890	0.868	0.879

Table 3.6. Similarity Matrix Scores.

Population	2DKSOM	1DKSOM	K-Means	Nearest Neighbor
S25M35	95	110	99	97
S50M35	130	129	107	112

Both the Rand Index and similarity matrix attempt to capture how well the proposed clustering algorithms accord with human judgment. As indicated by the high

percentages recorded in Table 3.5, the 1D KSOM has the best performance for both S25M35 and S50M35 using the Rand Index. These results imply that the 1D KSOM grouped the trusses in these populations in the manner that was most consistent with survey results.

The 1D KSOM also performs best for population S25M35 using the similarity matrix, as measured by achieving the highest grade in Table 3.6. Both the 1D and 2D KSOMs show strong performance for population S50M35, with the 2D KSOM outperforming the 1D by one point. This behavior indicates that the 1D KSOM was able to consistently identify similar and dissimilar trusses as defined by the author, although it was not always the best performer.

This idea of consistency is very important in selecting a classification method. Note that the 1D KSOM is always the best or second best method using both subjective rubrics. The 2D KSOM, by contrast, is the best performer for the S50M35 population using the similarity matrix, and the *worst* performer for the S25M35 population *using the same rubric*. Performance for the k-means and nearest neighbor algorithms also fluctuates. These algorithms describe the two populations with differing degrees of success. Taken together, the Rand Index and similarity matrix results highlight the performance of the 1D KSOM.

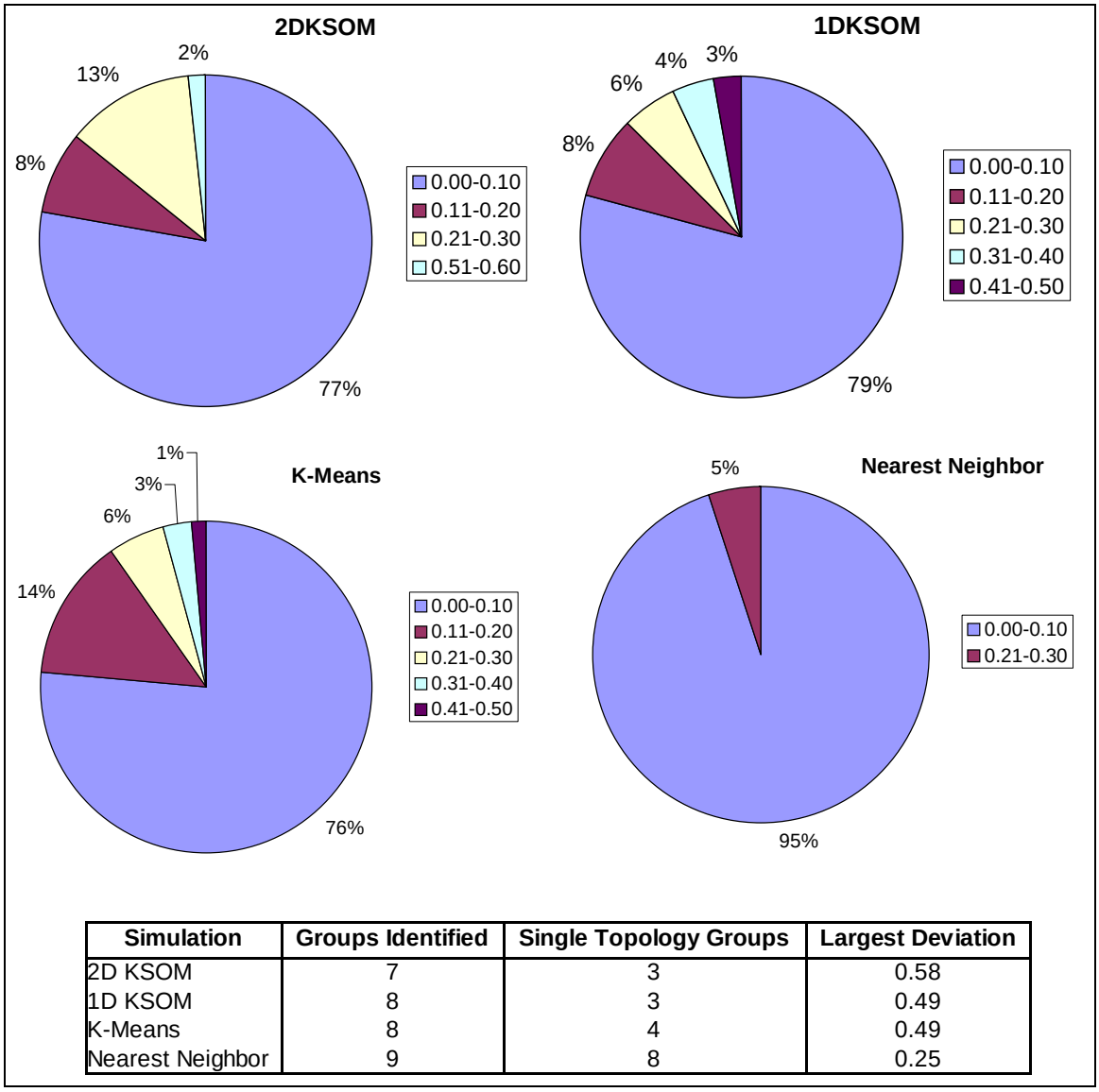
As for less subjective measures of classification efficiency, Table 3.7 presents the SSE data for all five populations. The best performing simulation method is shown in bold for each population. Recall that SSE should be minimized for effective clusters.

Table 3.7. Sum-of-Squares Errors.

Population	2DKSOM	1DKSOM	K-Means	Nearest Neighbor
S25M35	3.903	**1.494**	1.726	2.351
S25M50	**1.391**	1.615	2.233	1.778
S50M25	3.925	2.456	4.032	**2.066**
S50M35	8.480	5.118	5.783	**4.388**
S50M50	8.554	**6.496**	8.13	11.826

Both the 1D KSOM and nearest neighbor algorithms are the most highly-performing simulations for two of the five simulations. However, while 1D KSOM is never worse than the second best performer across all populations, the nearest neighbor algorithm is ranked third or fourth for the populations where it is not the optimal answer. Similarly, the 2D KSOM has surprisingly good performance for population S25M50, yet has otherwise poor SSE values. K-means seems to be more consistent than either of these methods but does not outperform the 1D KSOM.

While the SSE provides a point value for determining how tightly clustered the groups of a proposed classification are, Figures 3.7 through 3.11 present information about the standard deviations for the populations overall. Tables embedded within these figures show additional information about the number of identified groups, the number of single topology groups, and the highest standard deviation for any group. These figures were compiled from tables of statistical properties listed in Appendix B.



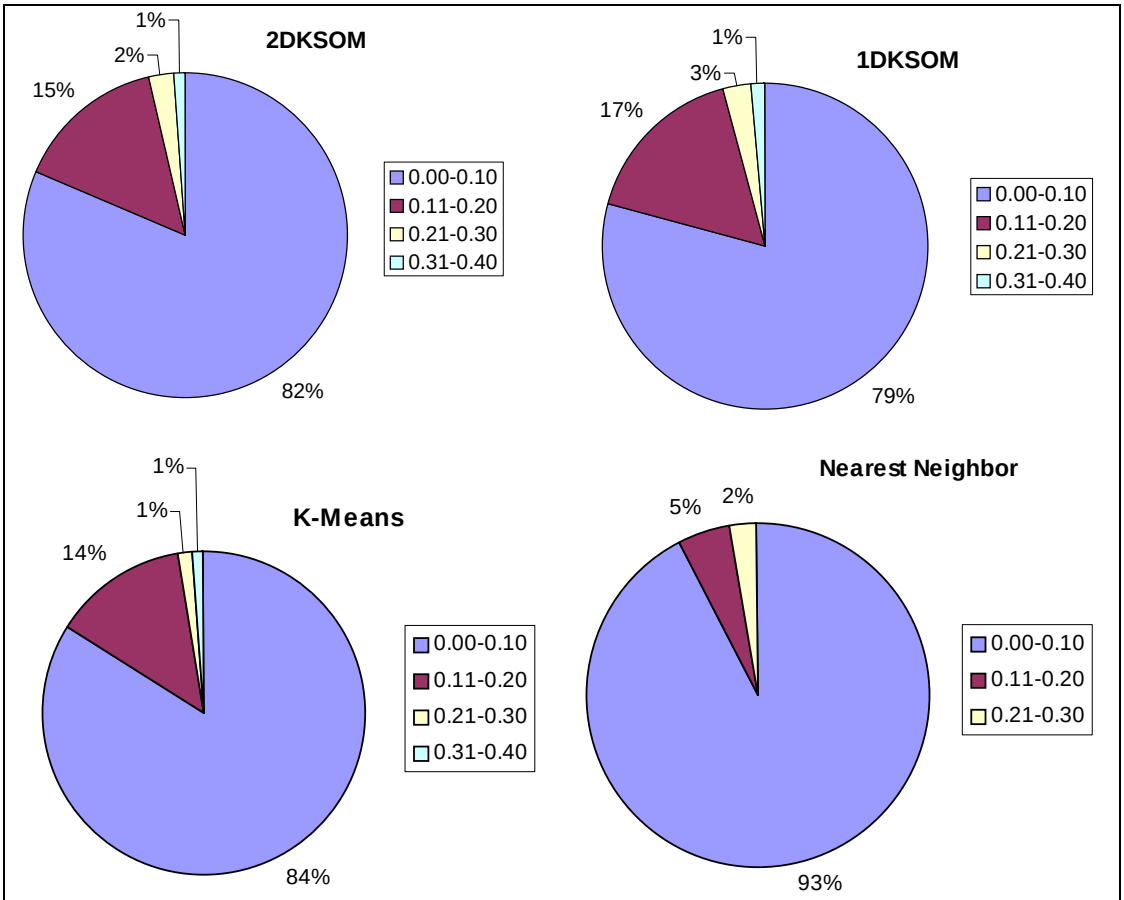
Simulation	Groups Identified	Single Topology Groups	Largest Deviation
2D KSOM	7	3	0.58
1D KSOM	8	3	0.49
K-Means	8	4	0.49
Nearest Neighbor	9	8	0.25

Figure 3.7. Distribution of standard deviations for population S25M35.

For population S25M35, shown in Figure 3.7, the nearest neighbor algorithm has the largest percentage (95%) of standard deviations in the 0.00-0.10 range and has the lowest maximum deviation (0.25). The 1D KSOM ranks second for both criteria, with 79% of the standard deviations lying at or below 0.10 and a maximum deviation of 0.49

(tied with k-means). However, the nearest neighbor simulation has eight single topology groups, which means that only one group proposed by nearest neighbors contains more than one truss design. This result indicates chaining, as can be seen in Figure B.4 of Appendix B. Conversely, the 1D and 2D KSOM simulations each created only three single topology groups, and the k-means algorithm proposed four.

The nearest neighbor algorithm also has the lowest maximum deviation and highest percentage of near-zero deviations for population S25M50, according to Figure 3.8. However, the nearest neighbor simulation proposes seven single topology groups, significantly higher than the five identified by k-means or the three for both KSOM classifications. K-means and the KSOM algorithms all have a maximum deviation of 0.32, only slightly higher than the nearest neighbor algorithm. The k-means algorithm ranks second in the percentage of small deviations, with both 1D KSOM and 2D KSOM following closely. It should be noted that the 1D KSOM reduces the required number of groups to eight, yet still shows small deviations among the identified groups.

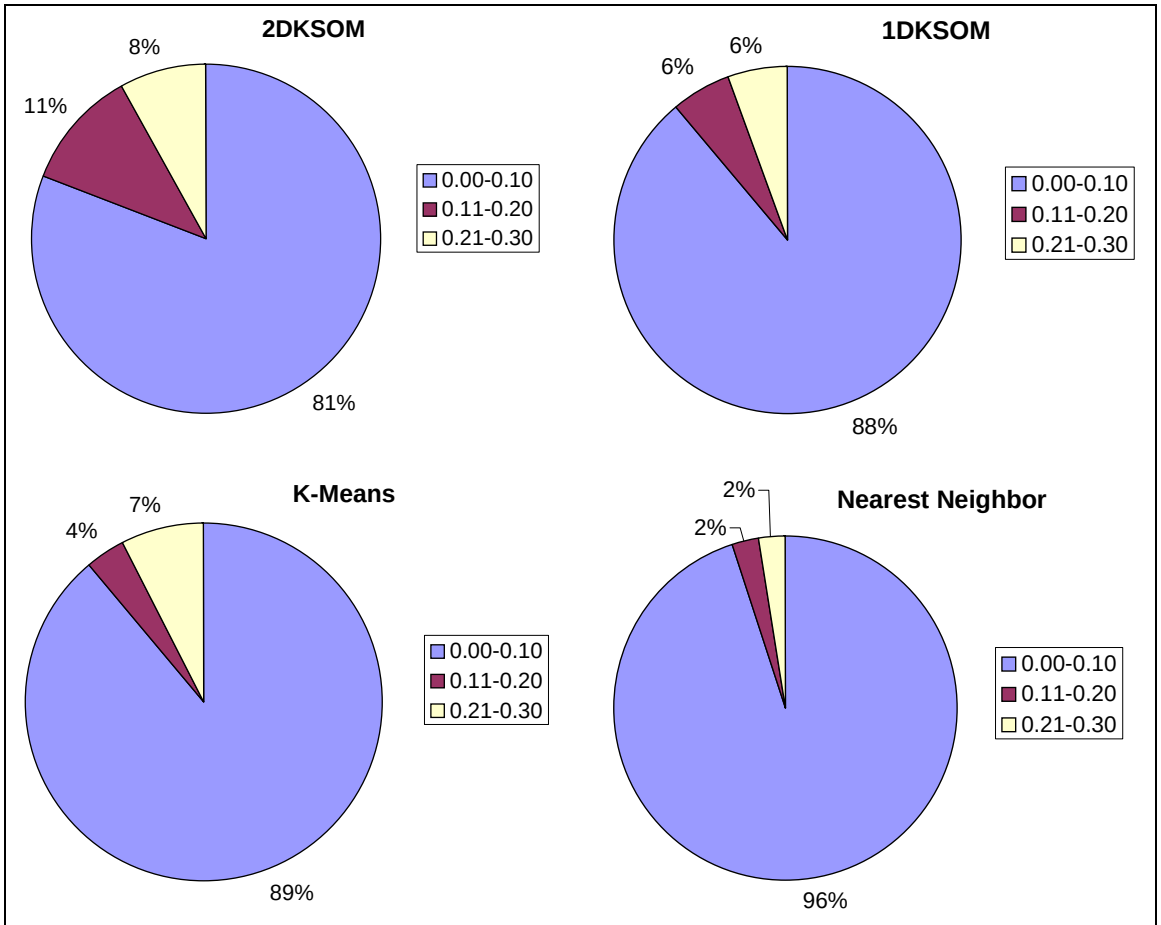


Simulation	Groups Identified	Single Topology Groups	Largest Deviation
2D KSOM	9	3	0.32
1D KSOM	8	3	0.32
K-Means	9	5	0.32
Nearest Neighbor	9	7	0.29

Figure 3.8. Distribution of standard deviations for population S25M50.

In population S50M25, shown in Figure 3.9, the nearest neighbor algorithm continues to have the largest percentage of small deviations and the smallest maximum deviation at the cost of forming more single topology groups than the other methods. K-means performance, overall, for this population is mixed: the algorithm has the second highest percentage of low standard deviations (differing from the 1D KSOM by 1%) and

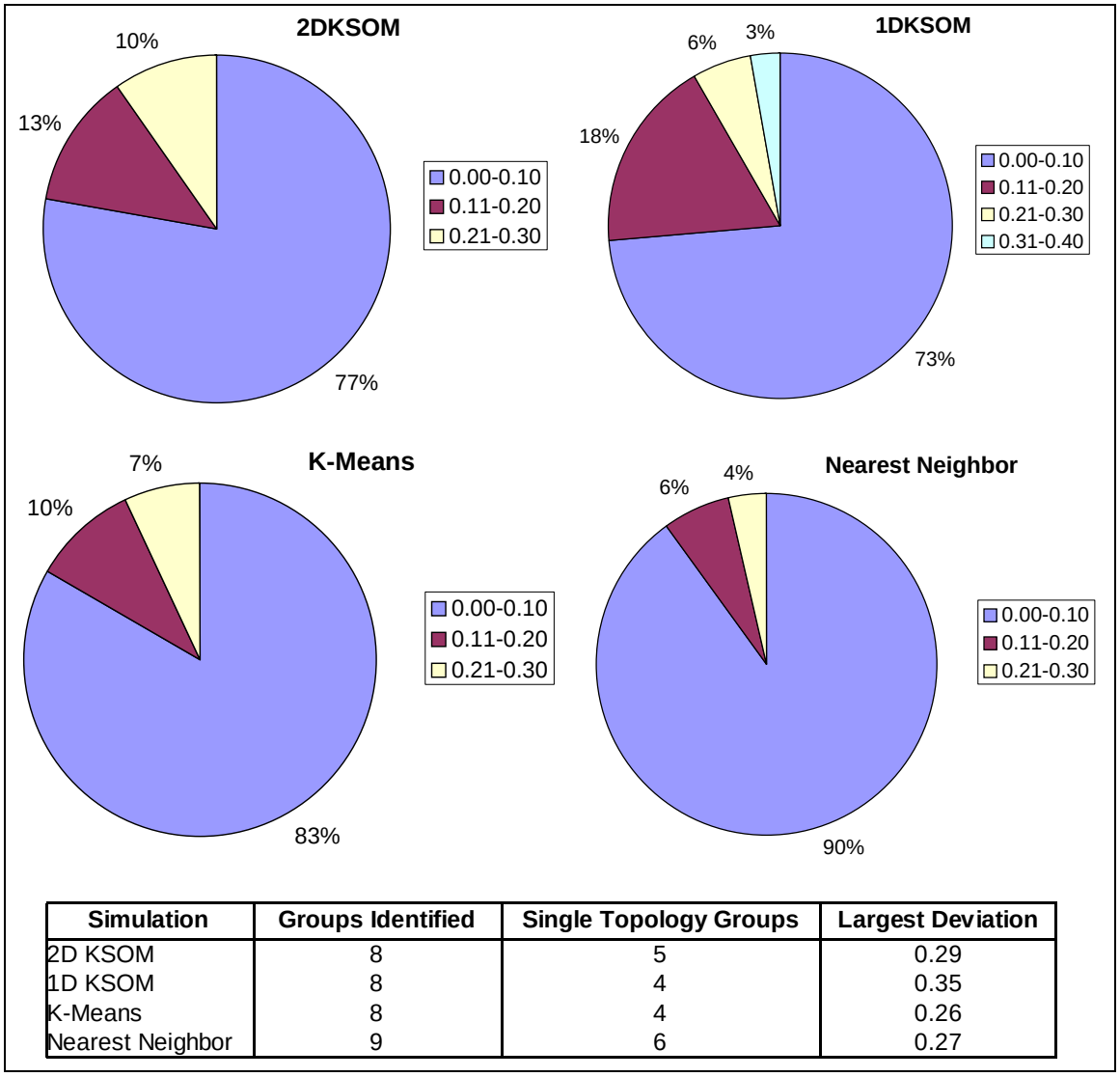
is tied for the largest number of single topology groups as well as the largest deviation. The 1D KSOM, by comparison, has a strong, if not optimal performance, in all three characteristics. This contrast helps illustrate the variability that may exist in performance between populations.



Simulation	Groups Identified	Single Topology Groups	Largest Deviation
2D KSOM	7	3	0.29
1D KSOM	8	4	0.26
K-Means	9	6	0.29
Nearest Neighbor	9	6	0.26

Figure 3.9. Distribution of standard deviations for population S50M25.

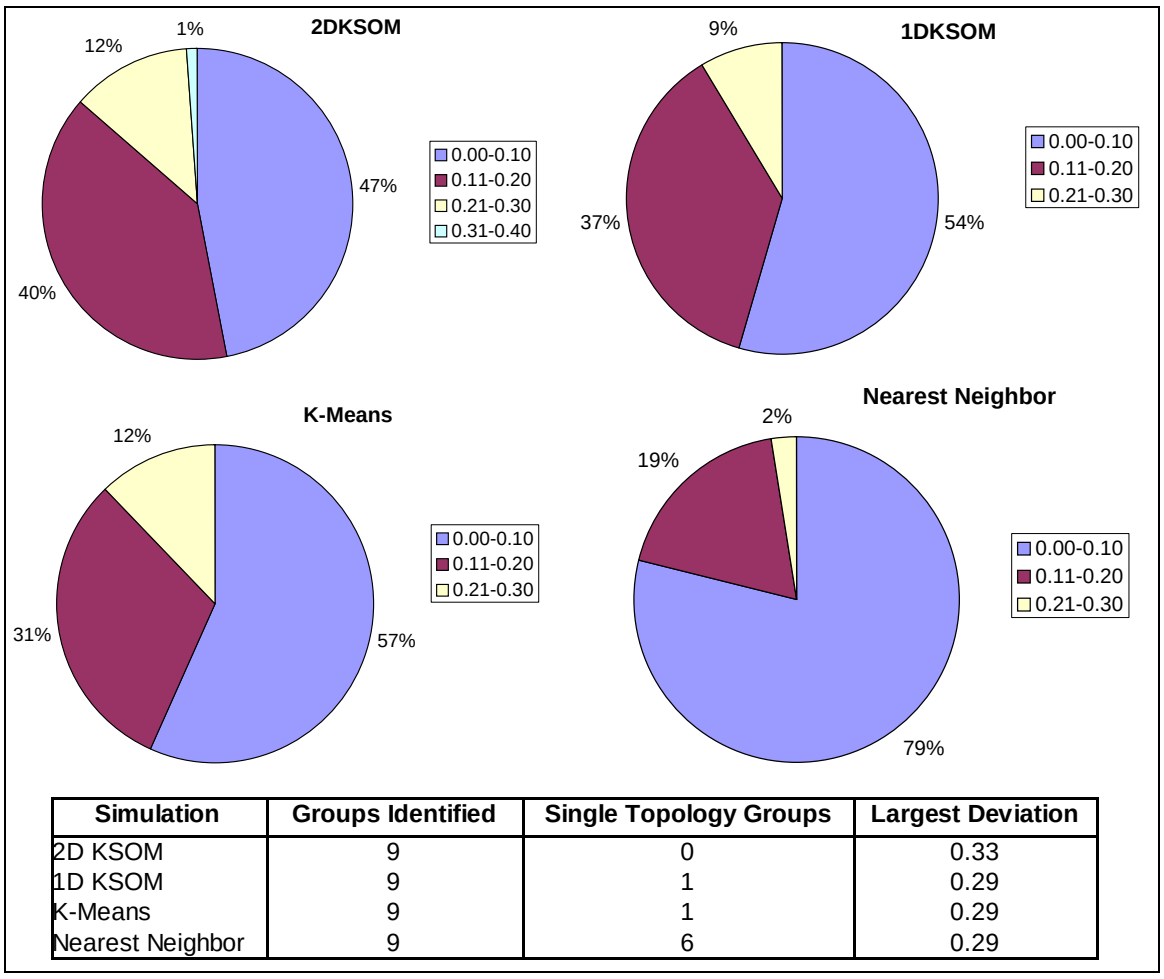
Figure 3.10 presents summary information for population S50M35. The nearest neighbor algorithm continues to have the largest percentage of small standard deviations and highest number of single topology groups. With the smallest maximum deviation and four single topology groups, k-means performs well for this population. The 1D KSOM also suggests four single topology groups but ranks poorly for the other criteria.



Simulation	Groups Identified	Single Topology Groups	Largest Deviation
2D KSOM	8	5	0.29
1D KSOM	8	4	0.35
K-Means	8	4	0.26
Nearest Neighbor	9	6	0.27

Figure 3.10. Distribution of standard deviations for population S50M35.

The final population, S50M50, has the standard deviations shown in Figure 3.11. Three of the methods have the same maximum standard deviation of 0.29, while the 2D KSOM has a maximum value of 0.33 and a significantly lower percentage of near-zero standard deviations. Both 1D KSOM and k-means algorithms identify only one single topology group. K-means has a larger percentage of 0.00-0.10 deviations as well as deviations lying in the 0.21-0.30 range when compared to the 1D KSOM.



Simulation	Groups Identified	Single Topology Groups	Largest Deviation
2D KSOM	9	0	0.33
1D KSOM	9	1	0.29
K-Means	9	1	0.29
Nearest Neighbor	9	6	0.29

Figure 3.11. Distribution of standard deviations for population S50M50.

Selected Classification Method

Four classification algorithms, encompassing three unsupervised clustering methods, have been considered for the task of subdividing truss populations into groups of similar design. The results presented on the previous pages clearly indicate that no single method will provide an optimal answer for every imaginable population. Variations in the size and complexity of the populations have lead to variations in the advantageousness of one or the other methods.

However, the preponderance of evidence suggests the one-dimensional Kohonen self-organizing map as an appropriate solution. This method reflects human judgments about truss placement as measured through the similarity matrices and the Rand Index. No other method's performance was as consistent with user input for both the S50M35 and S25M35 populations or made the key association between trusses "A" and "K" of population S50M35.

The 1D KSOM also shows significant advantages in terms of numerical measurements. At first glance, the nearest neighbor algorithm offers low SSE values for some populations and tends to keep the standard deviations for all populations low. This numerical efficiency, however, comes at the cost of creating many single topology groups. The nearest neighbor algorithm frequently falls victim to chaining and *fails* to group a significant portion of the population.

The creation of single topology groups is important because it indicates the presence of "super groups" (instances where the majority of truss designs are placed in one or two overcrowded categories). Not only would super groups be of little use in

reducing the selection burden on a human user, they would also fail to provide information about a user's design preferences. Either a user would select groups with only one or two topologies, in which case insufficient information would be gathered to discern preferences, or they would select overpopulated groups that would hamper preference identification by providing too much information. If preferences are poorly defined, a user-guided exploration of the IRR GA search space becomes infeasible.

As a counterpoint, the 1D KSOM may not always offer the *best* clustering map in terms of numerical efficiency, but the classifications proposed by this method are *consistently* good. The 1D KSOM does not show wide swings in performance when the complexity or size of a population alters, as do the other methods considered here. The SSE values for this method are low across the board and a minimum of single topology groups are created. Moreover, the 1D KSOM is capable of adaptively reducing the number of truss groups in the classification when appropriate, while the nearest neighbor algorithm can make no judgments on this matter.

Therefore, the 1D KSOM was selected as the classification mechanism for the present application. The 1D KSOM will be used to group truss designs based upon the characteristic feature vector defined in Chapter II. The user's selection of desirable topologies from among these groups will form an integral part of preference prediction, as will be discussed in Chapter IV. The user's evaluation of KSOM-generated feature maps will provide the information necessary to discern aesthetic design preferences so that these preferences may be applied to the fitness assessment of individuals in the IRR GA populations.

CHAPTER IV

PREFERENCE DETECTION

The classification mechanism discussed in the previous chapter provides information about a user's aesthetic and practical preferences. By categorizing the trusses according to perceived similarities in their characteristic feature vectors, the 1D KSOM arranges potential designs and allows users to select the most pleasing groups. This selection process allows for a user's likes and dislikes to be recorded for a given population.

However, the IRR GA creates many potential truss populations in the search for optimal designs. Therefore, it is crucial to convert information about what designs a users has selected using the 1D KSOM into a prediction of what designs a user will like from a *previously unexplored* population. It would be inefficient to directly query the user at each step of the IRR GA for feedback about the current population of trusses, since this would involve asking the user to evaluate thousands of designs by the time the genetic algorithm completed. Moreover, many of these designs would be presented repeatedly-- either in identical forms or with slight variations in geometry or member sizing. Certainly, direct feedback would allow the user to shape the conceptual design of the truss, but it would be a time consuming and fatiguing process.

This chapter explores methods for translating user selections into predictions of user preference. The characteristic feature vector developed in Chapter II remains the most effective source of information from which to make these predictions. The vectors

for trusses selected by the user will be compared to the vectors of trusses the user has never examined. Three heuristic learning tools were considered for this application and include the following:

- Kohonen's Self-Organizing Map (KSOM)
- Rough Set Reduct (RSR)
- Back-Propagation Neural Network (BPNN)

Background Information on Prediction Methods

This section will discuss the prediction methods considered in this study and how they were applied to the problem at hand.

Kohonen's Self-Organizing Map

Because of its success in perceiving similarities between truss topologies in the classification arena, it was thought that the KSOM could prove equally effective in identifying similarities among user preferences. Once a mapping of the initial population was complete, the trained weights of the KSOM were saved in a text file. During the prediction phase, these weights were used to reinitialize the network. It was then possible to categorize the new truss topologies according to the group classifications discovered in the previous population.

This process allows a new truss design to be classified according to how similar it is to a design already evaluated by the user. The user's group preferences were then used to determine what new trusses a user would like. For instance, if the user expressed interest in group 1 trusses during the initial evaluation, then all *new* designs classified as group 1 trusses because of similarities to these original designs are identified as user

preferred.

As before, a one-dimensional array of nine output neurons was used for the KSOM. During the classification phase, the neighborhood size started at eight neurons and decreased linearly every 1000 steps. Weight adaptation used 1000 ordering steps and 5000 convergence steps. The gain, or learning rate, started at 0.9 and decreased exponentially to 0.035.

Rough Set Reduct

The rough set theory developed by Pawlak in the 1980s is one method for dealing with vagueness and uncertainty in data (Pawlak et al. 1995). This method defines upper and lower approximations for data sets and uses these approximations to form conclusions. For instance, an information/decision table could list symptoms for a group of patients that would allow physicians to make decisions on whether or not a given patient had the flu (Pawlak et al. 1995). The lower approximation would contain the smallest number of variables necessary to make the decision, and the upper approximation would contain all of the information available. This process has been used for data reduction as well as determining dependencies between data and generating decision algorithms; applications extend to medicine, engineering design, and image processing (Pawlak et al. 1995).

In rough set theory, as explained in Pawlak (1981), objects are described by attribute sets. This description is similar to the characteristic feature vector defining individual truss designs in the current application, except that classic rough set theory uses discrete, rather than continuous, attributes. The attribute sets allow for comparisons

to be drawn between objects, or partitions of those objects (Pawlak 1981). However, not all attributes will be independent of each other. For example, if describing a group of people according to age, gender, education level, etc., it seems likely that an individual aged only two years will have an education level of zero (Pawlak 1981). Therefore, even if the education attribute were removed from the individual's set of descriptors, it would be possible to accurately predict education level. This process is an example of a "reduct." Rough set reducts allow for "superfluous" information to be removed from a data set (Pawlak 1981).

Reducts isolate the attributes that most clearly affect a classification decision. In the previous medical example, an attribute reduct could isolate the symptoms that would mostly likely indicate whether or not a patient was ill. "Our claim is that knowledge is deep-seated in the classificatory abilities of human beings and other species. ... Classification on more abstract levels, seems to be a key issue in reasoning, learning and decision making..." (Pawlak 1991).

Yanagisawa and Fukada extended rough set reduct (RSR) for use in interactive design applications. They considered user preferences for simple shapes, such as cylinders, as well as more complex designs, such as car profiles (Yanagisawa and Fukada 2003; Yanagisawa and Fukada 2004). The attributes describing these objects contained both discrete and continuous variables.

The researchers used the reduct concept from rough set theory to identify a user's "favored features" for a given product (Yanagisawa and Fukada 2003). They created a comparison matrix to discriminate between user-designated "good" and "bad" designs.

The most significant features distinguishing good designs from bad were held constant, and the remaining attributes were altered until users were satisfied with the end product (Yanagisawa and Fukada 2003; Yanagisawa and Fukada 2004). This adapted RSR method was shown to be more effective, in terms of faster convergence as well as total user satisfaction, than interactive GAs where user evaluation formed the total fitness function for a product (Yanagisawa and Fukada 2003). This improved performance was believed to occur because, while GAs randomly alter feature values through crossover, mutation, etc., RSR discovers areas of "attention", overall design concepts that strongly appeal to the users, and ensures that these characteristics are passed on to future generations (Yanagisawa and Fukada 2005).

Yanagisawa and Fukada's work was taken as a model for creating a predictive RSR algorithm for the truss application. The groups selected using the 1D KSOM were denoted "good" designs, and the unselected groups were considered "bad." The statistical averages and standard deviations were determined for both good and bad groups, and a discernability matrix was calculated according to Equation 4.1. A discernability matrix shows the significance of differences between the characteristics of good vs. bad designs. For each good group i, D_{ik} represents the group's discernability for attribute k. A feature's discernability was calculated by considering the absolute value of the difference between the feature average for the good group i, x_{cik}, and the feature average for each bad group j, x_{cjk}. The minimum discernability for the attribute was taken to be D_{ik}.

$$D_{ik} = \min_{j} \left| x_{cik} - x_{cjk} \right| \tag{4.1}$$

This process creates a matrix where the row number refers to the number of groups selected by the user and the column represents the number of attributes in the characteristic feature vector. The entries of this matrix identify the smallest difference between an attribute in the "good" designs and the same attribute in all of the "bad" designs. If any of these discernabilities prove to be significant, then they represent clues about *why* a user selected that group. These clues allow for deductions to be made about what features a user may prefer in new designs.

Attributes were considered significant indicators of user preference when the discernability for a selected group was larger than the standard deviation for the corresponding feature of that group. This criterion was selected because it had physical meaning. If trusses within a group closely resembled each other with regards to one characteristic, the standard deviation for that feature would be small. A user's decision to select the group would likely take into account such a dominant feature. If the groups not selected by the user did not display similar values for this attribute, the conclusion that the group was selected in part because the user favored the dominant feature seems logical. On the other hand, if a feature in a selected group showed significant variation within that group, as indicated by a high standard deviation, it becomes less likely that the user considered that feature in selecting the group.

Once this process had been applied to determine a user's "favored" features, it became possible to consider new designs in light of this information. Each new truss topology was compared to the previously selected good designs, and a score tallied to represent similarities. The score was calculated by taking the absolute value of the

difference between the group's average feature for the favored characteristic and the corresponding value for the new design. This difference was then compared to the standard deviation for the favored feature, and points were added to the score in response to how many standard deviations existed between the new design and the group average.

The smallest score calculated for a new truss topology across all features was saved to represent how close that topology was to the favored features of the closest preferred group. Once all of the new designs had been graded, the best performing topologies were output as predictions. After experimenting with the number of predictions made for each new population, it was decided that the 25% of topologies with the lowest scores offered a reasonable estimate of a user's preferences. In the event that more than 25% of the population shared an identical low score, all topologies with this score would be predicted.

Back-Propagation Neural Network

The back-propagation neural network (BPNN or back-prop) is a supervised learning algorithm capable of mapping highly nonlinear relationships between a set of causal variables and their corresponding outcome(s). The BPNN is therefore useful in pattern recognition applications such as the user preference problem posed here. The back-prop algorithm uses a gradient-descent search to learn patterns from a series of input-output training data (Hassoun 1995). This training occurs efficiently and in a straightforward manner that has made the back-prop algorithm extremely popular in a variety of applications and helped to spark current interest in neural networks as a whole (Hassoun 1995).

The BPNN was selected as a potential method for predicting user preferences among unexplored truss topologies because it can be trained to recognize input-output pairs. The user's evaluations of the initial population would form the basis of this training set. The characteristic features of these trusses would be the input variables, and a yes/no indication of whether or not they were selected by the user would comprise the output variable. After training, the BPNN would be able to discern a user's selections from the original input and extend this learning to judge the acceptability of newly proposed designs. Both Hassoun (1995) and Haykin (1999) provide summaries of the back-prop algorithm, and the following discussion and equations are derived from these sources.

The training set is represented by a vector of I attributes, **x**, and a vector of L expected responses, **d**, arranged in the ordered pair (**x**,**d**). Given the attributes (x_1, x_2,...,x_i,...,x_I) of an individual, the trained network should respond with outputs (y_1, y_2,...,y_l,...,y_L) that correspond to the expected responses (d_1, d_2,...,d_l,...,d_L). The error between vectors **d** and **y** is used to adjust the network weights to better reflect the desired output.

The BPNN contains three or more layers of neurons, as shown in Figure 4.1. The first, or input, layer contains one neuron for every member of the attribute vector, **x**, describing an individual, plus an extra neuron used as a "bias" with an input of 1. The input layer is connected to the first layer of hidden nodes by a set of weights, w_{ji}. The values of the hidden layer node activations, represented by the vector **z**, are calculated using these weights, the input values, and a sigmoidal activation function, as shown in

Equation 4.2. Here, z_j is the value of the j^{th} hidden layer node, λ_h is a multiplier for the hidden layer that lies between 0 and 1, and net_j is defined in Equation 4.3.

$$z_j = \tanh(\lambda_h \, net_j) \tag{4.2}$$

$$net_j = \sum_{i=1}^{I+1} w_{ji} \, x_i \tag{4.3}$$

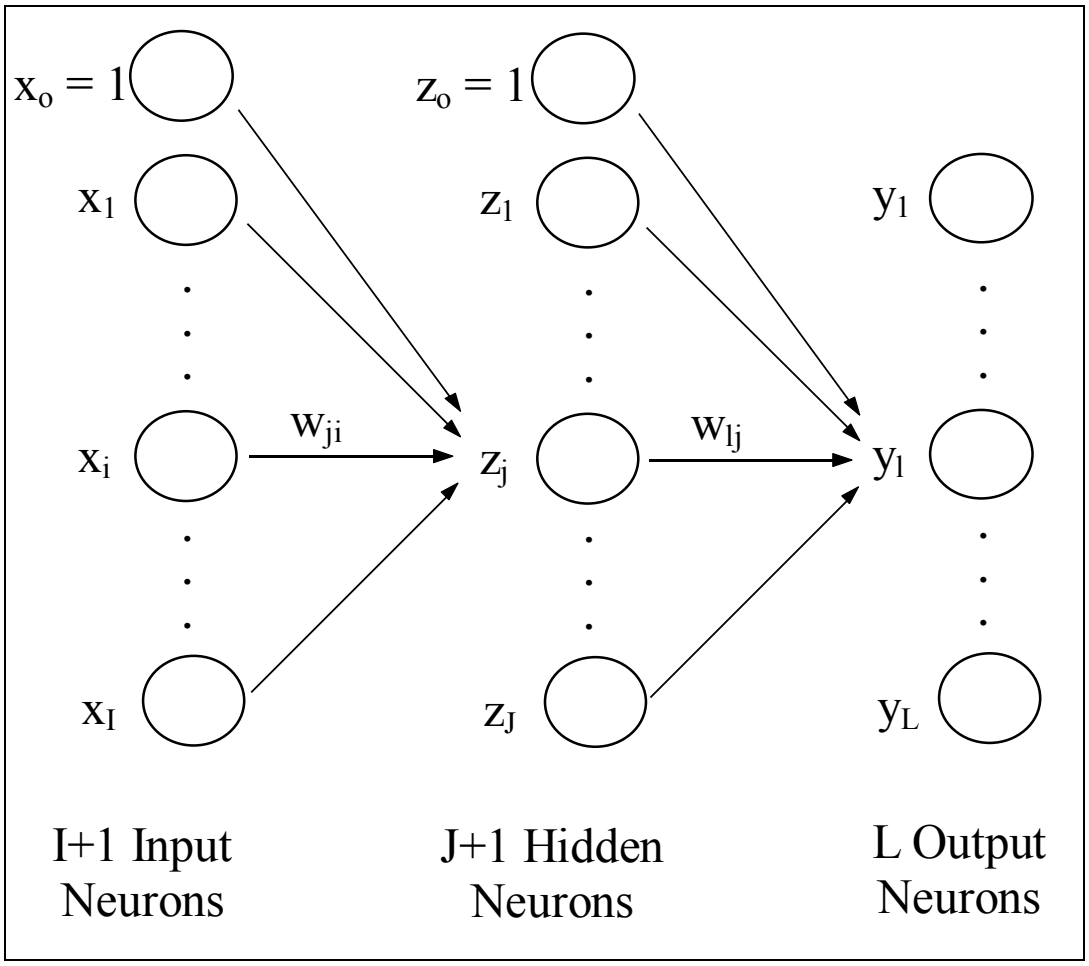


Figure 4.1. Schematic view of a back-prop network with a single layer of hidden nodes.

The hidden layer nodes are in turn connected either to the output layer or to a

second layer of hidden nodes. An additional bias term is added for each hidden layer, once again having a value of 1. If only a single layer of hidden nodes is included, then the **z** vector may be used to calculate output values according to Equations 4.4 and 4.5. Note that these expressions are similar to Equations 4.2 and 4.3. The w_{lj} terms refer to the weights connecting the hidden layer nodes to the output layer nodes, and a different multiplier, λ_o, may be used to distinguish the output layer. This multiplier serves to flatten the sharp S-shape of the hyperbolic tangent, thus softening the on/off effect of the activation function.

$$y_l = \tanh(\lambda_o \, net_l) \tag{4.4}$$

$$net_l = \sum_{j=1}^{J+1} w_{lj} \, z_j \tag{4.5}$$

The total error between the vectors **d** and **y** is expressed in Equation 4.6, with respect to all of the network weights **w**. If this quantity is differentiated with respect to **w**, it becomes possible to propagate the error backwards through the system as shown in Equations 4.7 and 4.8. Changing both sets of weights in the manner shown will help bring the system more closely in line with the desired outcomes. The learning rate parameter, ρ, controls how quickly these changes occur. The properties of sigmoidal functions mean that the derivatives expressed in these equations exist for all points and may be written simply in terms of the **y** and **z** vectors (see Equation 4.9).

$$E(\mathbf{w}) = \frac{1}{2} \sum_{l=1}^{L} \left(d_l - y_l\right)^2 \tag{4.6}$$

$$\Delta w_{lj} = \rho_o \left(d_l - y_l\right) y_l' \, z_j \tag{4.7}$$

$$\Delta w_{ji} = \rho_h \left[\sum_{l=1}^{L} (d_l - y_l)\ y_l^{'}\ w_{lj} \right] z_j^{'}\ x_i \tag{4.8}$$

$$\frac{d}{dx}\left(\tanh(x)\right) = 1 - x^2 \tag{4.9}$$

The Hassoun (1995) and Haykin (1999) texts present suggestions for improving this basic back-prop algorithm as well as tips for selecting parametric values. This information was considered in the creation of the BPNN used in this research. One suggestion was to add a momentum term to Equation 4.8 in order to speed convergence. The final network configuration had one output neuron and ten input neurons; the input neurons correspond to the nine elements of the characteristic feature vector plus a bias. From empirical studies, one hidden layer of eighteen neurons, including a bias, provided sufficiently accurate convergence. Other parameters were selected for the network after experimentation aimed at optimizing BPNN performance and are listed in Table 4.1.

Table 4.1. Back-Propagation Neural Network Parameters.

Parameter Name	Variable	Value
Hidden layer multiplier	λ_h	0.9
Output layer multiplier	λ_o	0.1
Hidden layer learning rate	ρ_h	0.2
Output layer learning rate	ρ_o	0.1
Momentum rate	α	0.1

As stated previously, training pairs were created from the user's KSOM selections. First, the feature values of the input trusses were re-normalized to lie between -0.5 and 0.5. These features, along with the desired outcome, were written to

an input file. If a truss belonged to a group selected by the user, it was given an outcome of "1;" otherwise, the desired output was "-1." The feature patterns and outcomes were then presented to the BPNN as randomly ordered batches until a desired level of convergence was reached.

After training was complete, the new truss designs were presented to the network. If the BPNN assessed a positive-valued output, then the user was predicted to like the truss. Likewise, negative-valued outputs corresponded to trusses the user would not prefer.

Methodology for Selecting a Preference Detection Method

As with the selection of a classification mechanism in Chapter III, identifying a method for predicting a user's preferences required performance evaluations on several levels. Judgments need to account not just for which method was the most numerically efficient but also which method's results most coincided with human intuition. Before a prediction method could be incorporated into the dynamically changing populations of the IRR GA, its ability to identify user preferences in a static setting had to be established.

To accomplish this goal, two types of preference trials were conducted. These trials made use of static truss populations generated using the algorithm described in Agarwal and Raich (2005). The first set of populations, referred to as the "original set", contains the S25M35, S25M50, S50M25, S50M35, and S50M50 populations examined in previous chapters. These populations will be referred to as S25M35(a), etc., from this point. The second set of populations, called the "preference set", was created using the

same inputs as the original set. Thus, the preference set contained five populations, S25M35(b), S25M50(b), S50M25(b), S50M35(b), S50M50(b), each of which mirrored the complexity and population size of one of the populations in the original set. These new populations are shown in Figures 4.2 through 4.6.

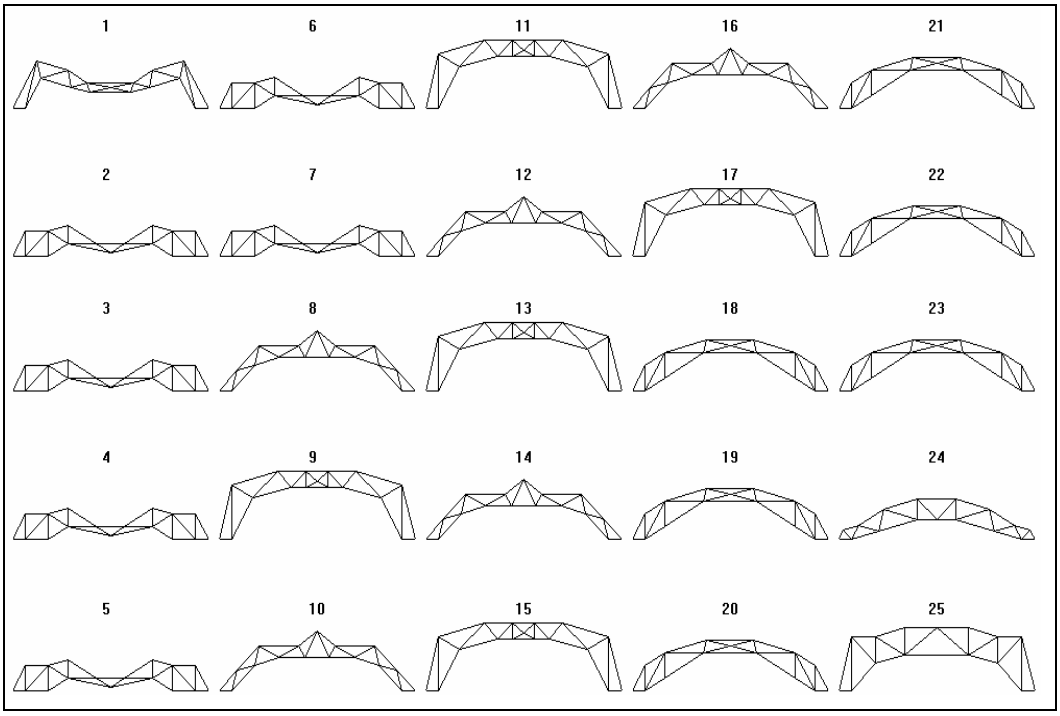


Figure 4.2. Population S25M35(b) has 25 trusses with 35 members per truss.

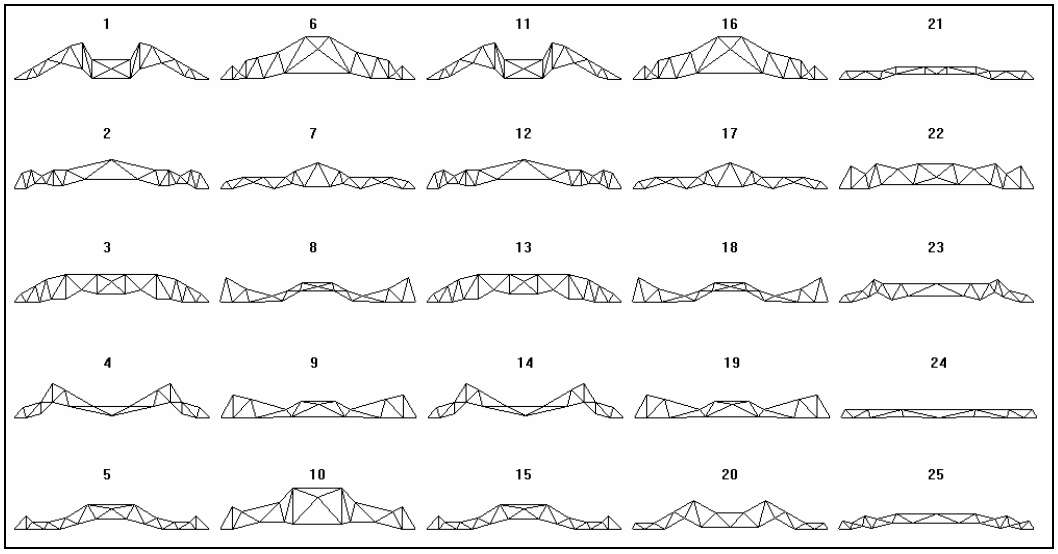


Figure 4.3. Population S25M50(b) has 25 trusses with 50 members per truss.

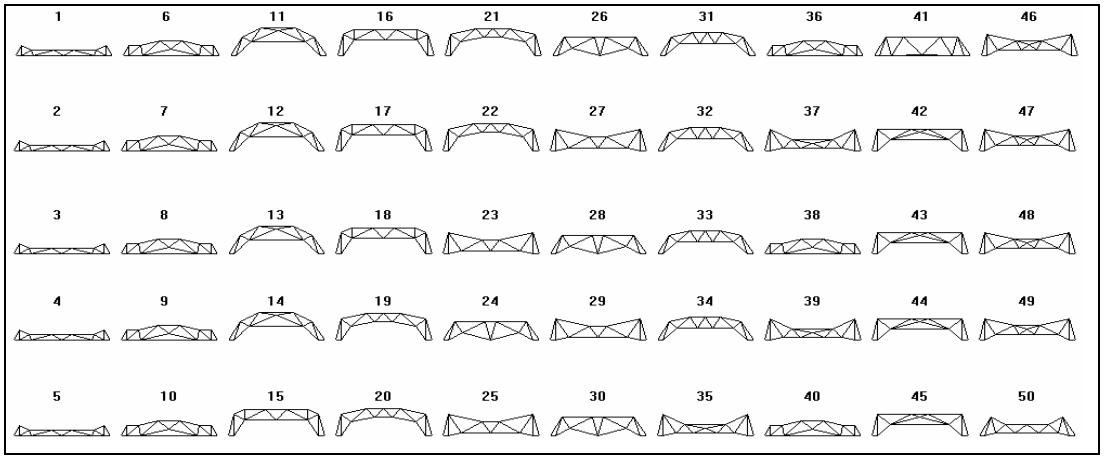


Figure 4.4. Population S50M25(b) has 50 trusses with 25 members per truss.

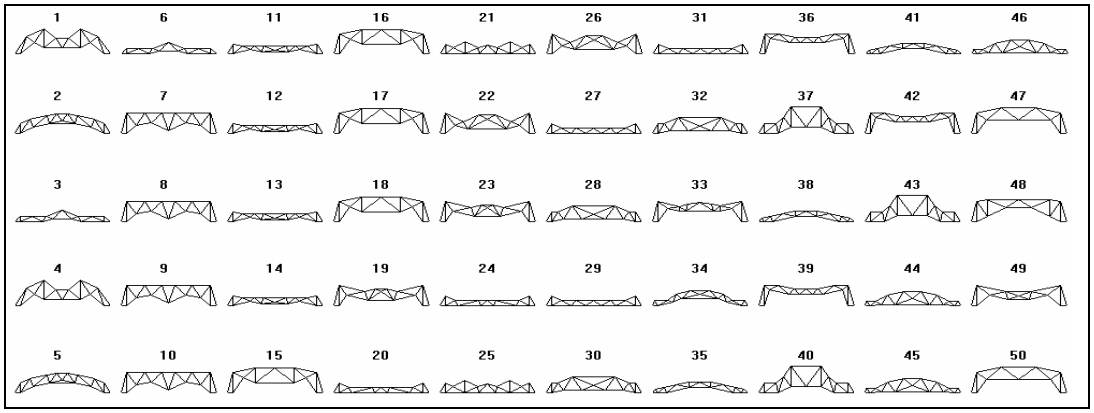


Figure 4.5. Population S50M35(b) has 50 trusses with 35 members per truss.

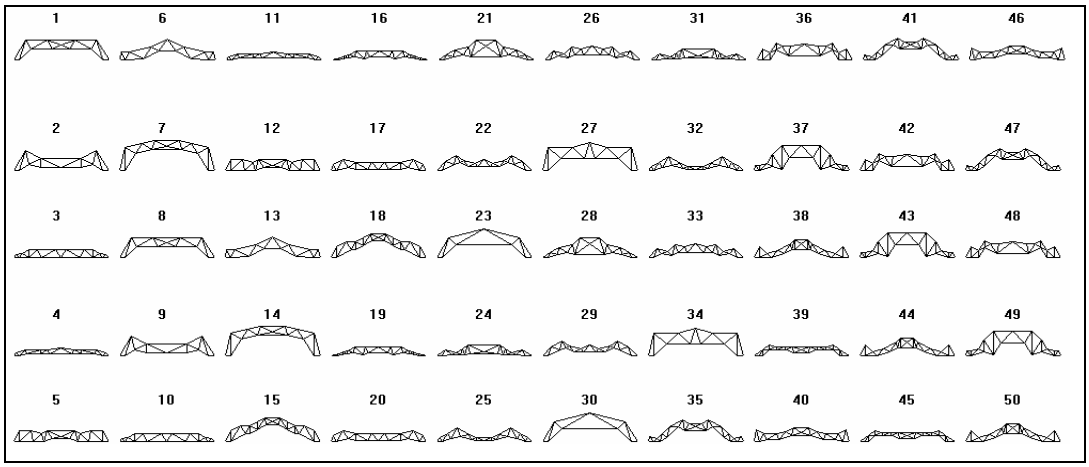


Figure 4.6. Population S50M50(b) has 50 trusses with 50 members per truss.

The first trials conducted with these populations were the input reproduction trials. Here, the user made selections from among the original set populations; the preferred topologies are shown in Figures 4.7 through 4.11. These preferences were presented as input for all of the preference detection techniques. Once this data had been adequately analyzed, the algorithms were asked to extrapolate the user's preferences to a

second population. The key element of the input reproduction trials was that the *original set* of populations was presented to the algorithms a second time, rather than presenting a new population. Therefore, each of the prediction algorithms were asked to identify preferences they had already seen. The purpose of these trials was to *reproduce* the user's inputs. Success at this task was viewed as critical to a method's ability to identify and distinguish between trusses using the characteristic feature vector.

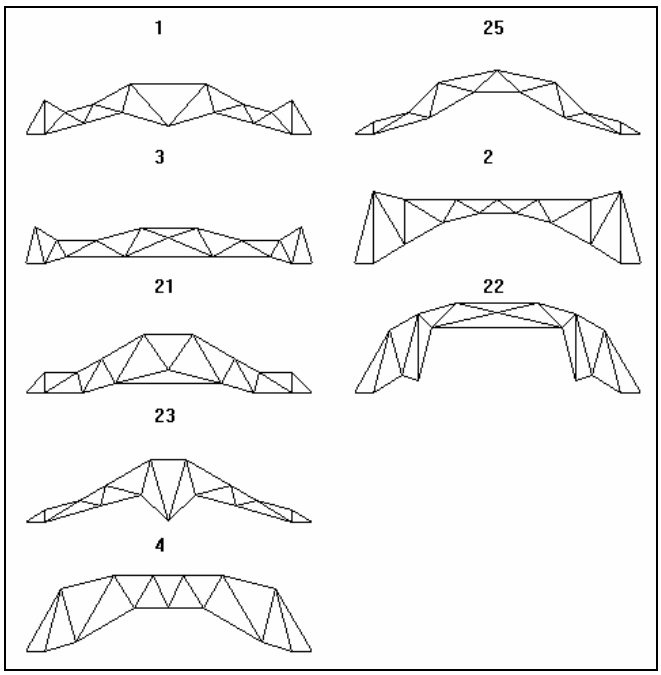


Figure 4.7. User preferred topologies from population S25M35(a).

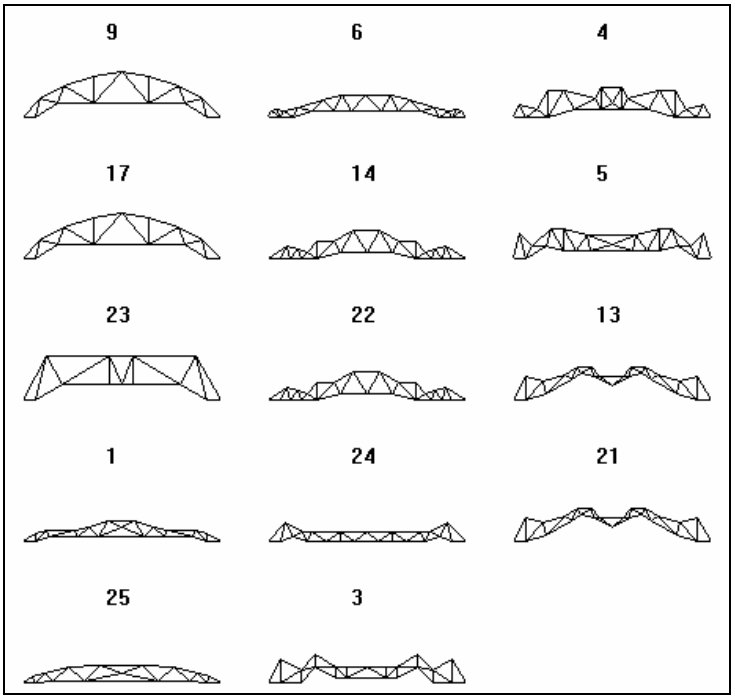


Figure 4.8. User preferred topologies from population S25M50(a).

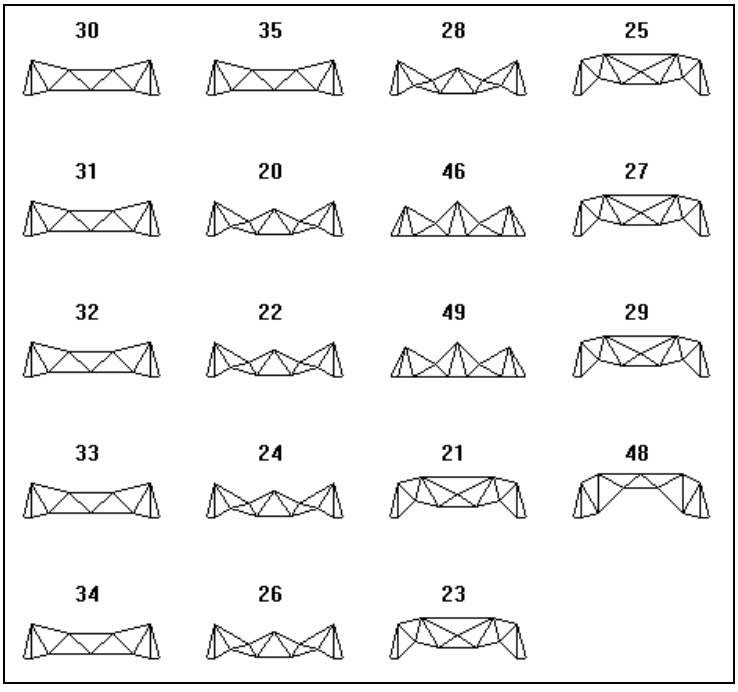


Figure 4.9. User preferred topologies from population S50M25(a).

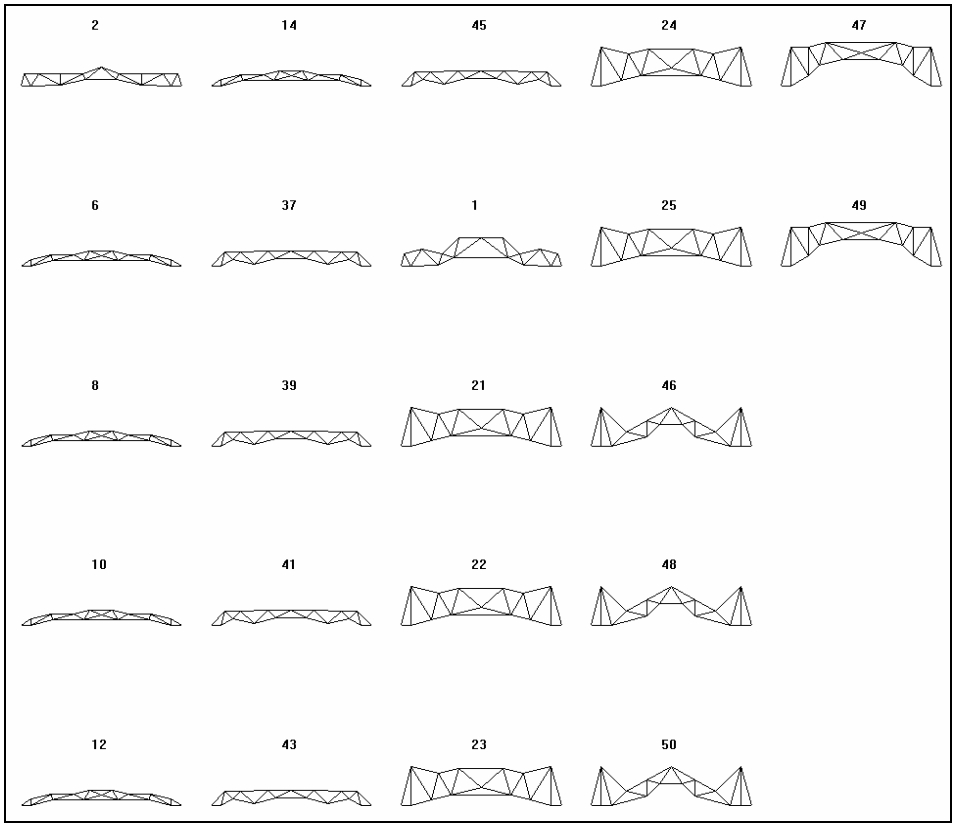


Figure 4.10. User preferred topologies from population S50M35(a).

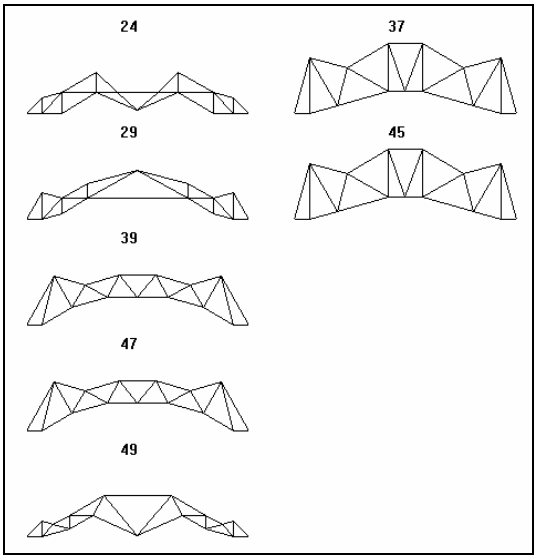


Figure 4.11. User preferred topologies from population S50M50(a).

As in the input reproduction trials, the preference detection trials began by presenting the user's selections from the original population set to the prediction algorithms. Once again, the different methods predicted a user's preferences in a second population based upon preferences previously explored. In this instance, however, predictions were drawn from the *preference set* populations. Therefore, the goal of this analysis was to extrapolate a user's selections in the original set to topologies they were likely to prefer in the preference set. Before presenting the preference set topologies to the prediction algorithms, these topologies were identified as being preferred, acceptable, or unacceptable to the user. Figures 4.12 to 4.16 show the topologies that a user either preferred or found acceptable in each population of the preference set.

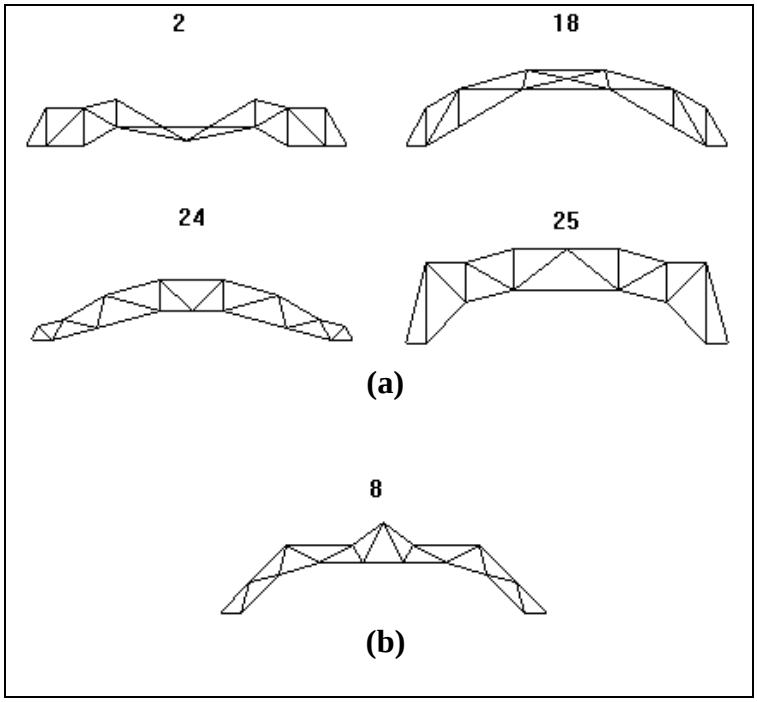


Figure 4.12. (a) Preferred topologies and (b) acceptable topologies for population S25M35(b).

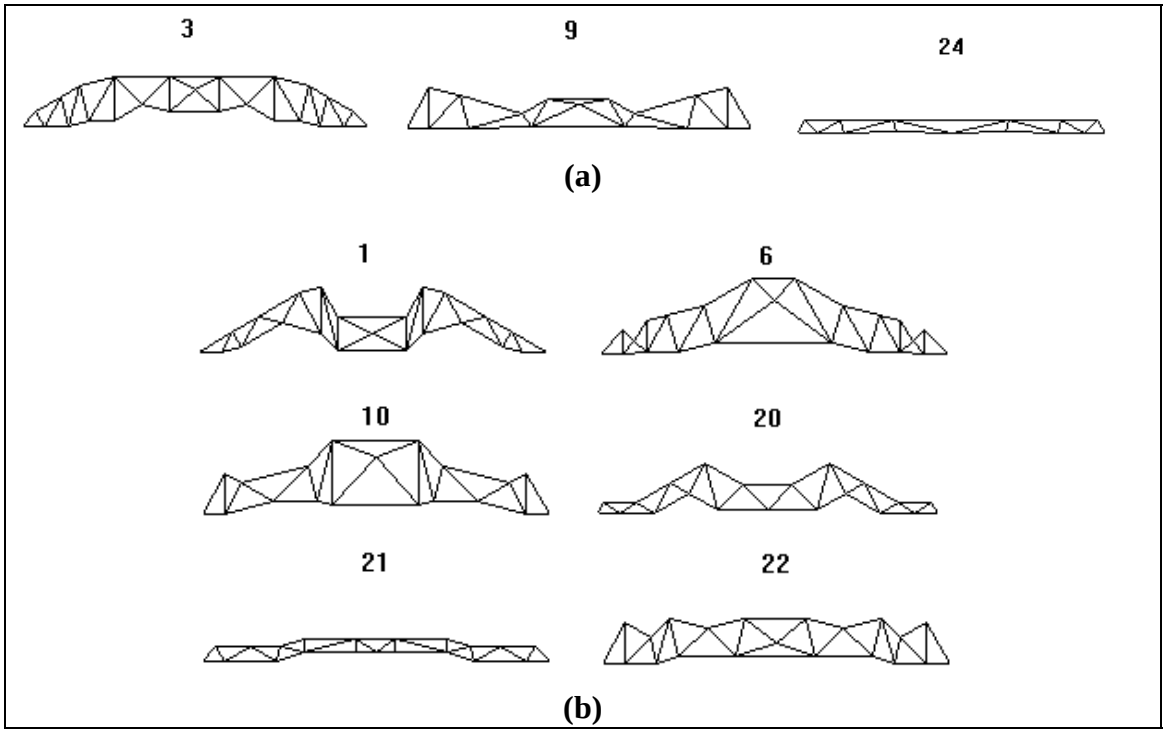


Figure 4.13. (a) Preferred topologies and (b) acceptable topologies for population S25M50(b).

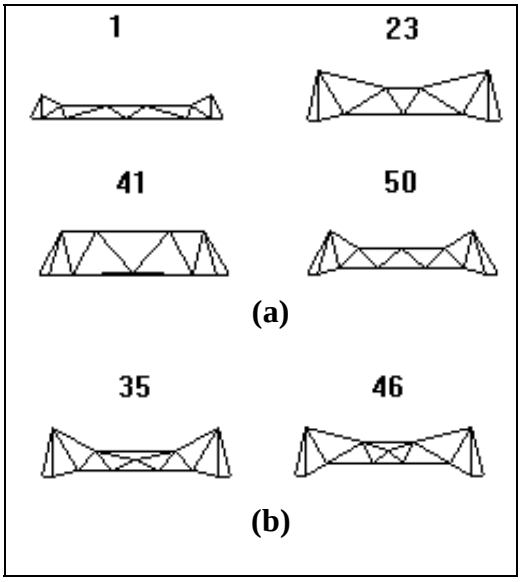


Figure 4.14. (a) Preferred topologies and (b) acceptable topologies for population S50M25(b).

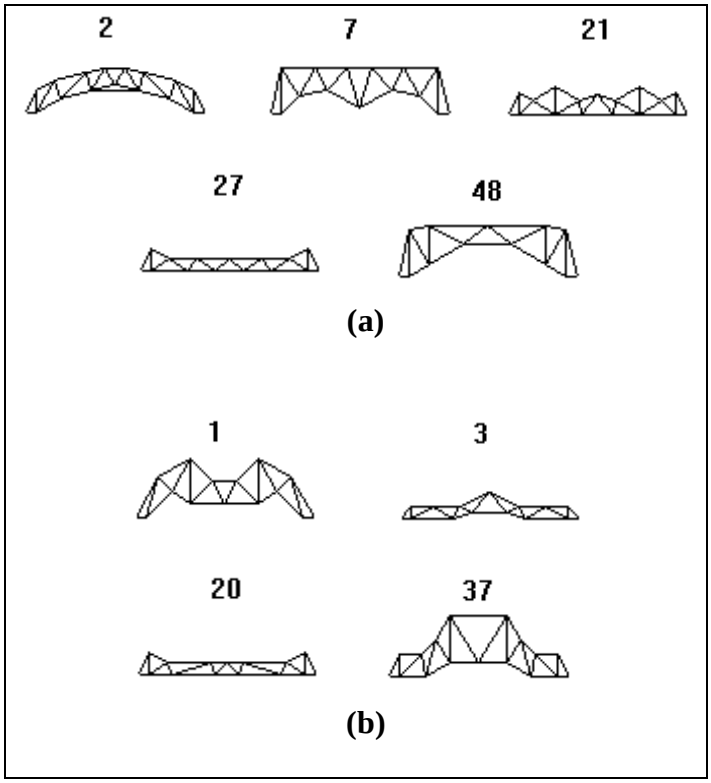


Figure 4.15. (a) Preferred topologies and (b) acceptable topologies for population S50M35(b).

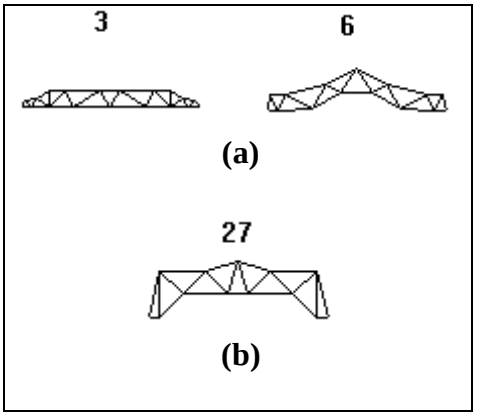


Figure 4.16. (a) Preferred topologies and (b) acceptable topologies for population S50M50(b).

The input reproduction and preference detection trials were used to verify different aspects of the prediction algorithms' performance: whether or not the inputs were correctly analyzed, and whether or not predictions based on these analyses were accurate. In order to answer these questions from a numerical standpoint, both trials used a sum-of-squares error (SSE) to quantify the distance between the originally selected topologies and those predicted by the proposed algorithms. As in Equation 3.4, SSE sums the squared Euclidean distance between the vectors of average feature values and the feature values of each predicted truss. Since the RSR and BPNN algorithms did not directly assign new designs to the KSOM-identified groups, membership to the closest lying group was inferred. The Euclidean distance of each new topology to the existing KSOM groups was calculated, and the topology was associated with the group having the lowest distance.

For ease of comparison between methods, the SSE was normalized by the number of trusses predicted in each algorithm. Therefore, prediction algorithms were neither rewarded nor penalized for the number of predictions they made. Instead, the SSE was used to measure the degree, on average, that these predictions accorded with the previously identified preferences.

In the input reproduction trials, performance judgments also made use of two percentages. These percentages were calculated based on the number of topologies selected by the user, the number of topologies predicted by the algorithm, and the number of predicted topologies that were among the user's selections. The first percentage represented how many of the topologies predicted by the algorithms had been

selected by the user, as shown in Equation 4.10. The second percentage illustrated how many of the user's selections were predicted; that is, it helped indicate whether *all* of a user's preferences were being represented or only a portion of these preferences. This percentage is shown in Equation 4.11.

$$\text{Correct Predictions} = 100\left(\frac{\text{Number of Selections Predicted}}{\text{Number Predicted}}\right) \tag{4.10}$$

$$\text{Selections Predicted} = 100\left(\frac{\text{Number of Selections Predicted}}{\text{Number Selected}}\right) \tag{4.11}$$

The preference detection trials made use of two similar percentages. These calculations, however, were based on tallies of the number of preferable, acceptable, and unacceptable topologies predicted by the algorithm. The number of total user preferences for each population was also used. The first percentage, shown in Equation 4.12, represented how many of the predicted topologies were acceptable to the user; that is, how many of them were either the user's first or second choice. The second percentage tracked how many of the user's preferences were correctly predicted by the algorithms; see Equation 4.13.

$$\text{Acceptable Topologies} = 100\left(\frac{\text{Number Preferred} + \text{Number Acceptable}}{\text{Total Number Predicted}}\right) \tag{4.12}$$

$$\text{Preferences Predicted} = 100\left(\frac{\text{Number Preferred}}{\text{Total Number of Preferences in the Population}}\right) \tag{4.13}$$

Mathematica-generated plots were also used to visualize how well the preference detection trials matched the selections of the original set. As before, these plots are three-dimensional representations of truss locations in feature-space. The nine default

feature characteristics of each truss were combined to create these graphs using the normalized lengths described in Chapter II. Points shown in red represent the trusses selected from the original population set, and blue points illustrate the proximity of the user's selections to preference set truss predictions.

Presentation and Discussion of Prediction Results for KSOM, RSR, and BPNN

Results for the input reproduction and preference detection trials are presented in the next two sections. Once again, pictorial results presented in the text will be limited to avoid confusion; the preference predictions for both trials will be displayed for populations S50M35(a) and S50M35(b). Results for all other populations may be found in Appendix C.

Input Reproduction Trials

The predictions made by the KSOM, RSR, and BPNN algorithms during the input reproduction trials are presented in Figures 4.17 through 4.19 for Population S50M35(a). Recall that the purpose of these trials was to indicate how well the prediction algorithms were able to replicate the user selections shown in Figures 4.7 through 4.11.

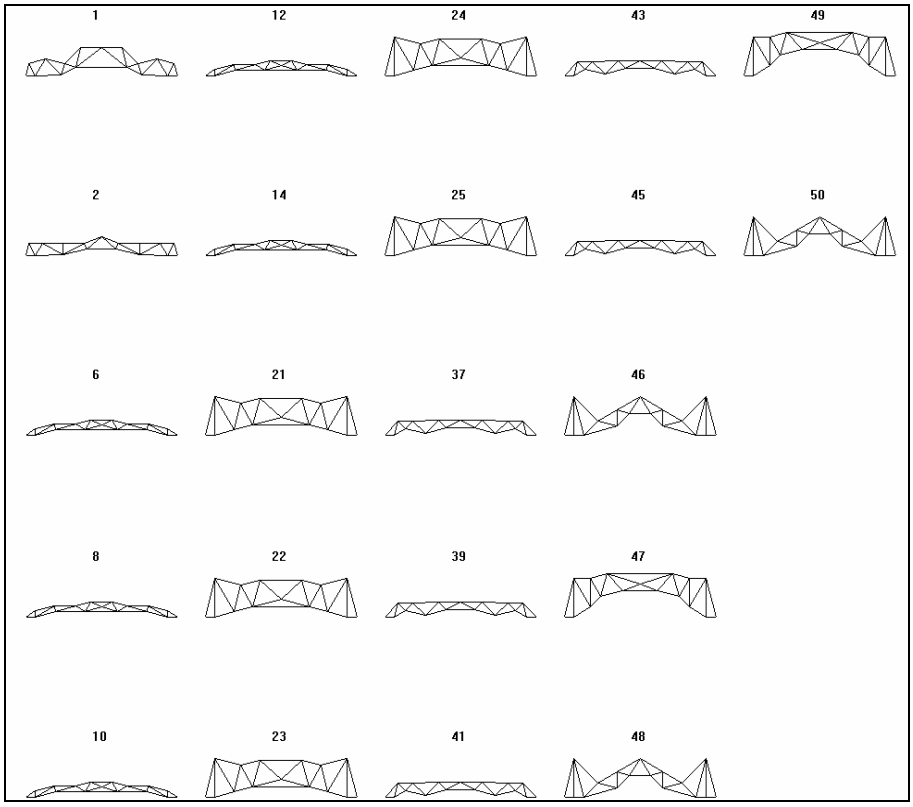


Figure 4.17. KSOM predictions for population S50M35(a) during input reproduction trials.

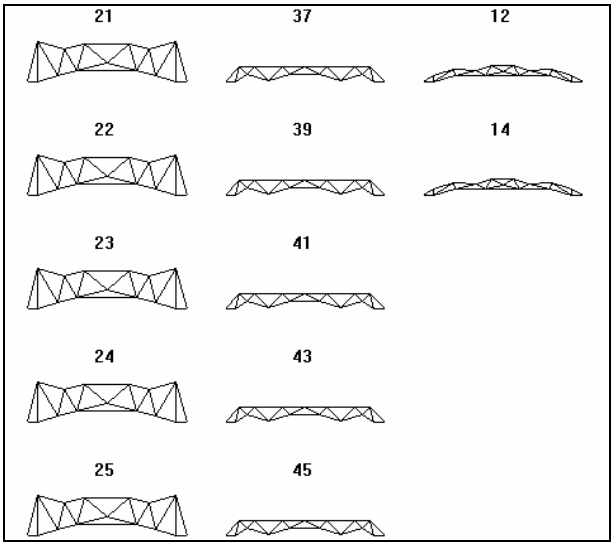


Figure 4.18. RSR predictions for population S50M35(a) during input reproduction trials.

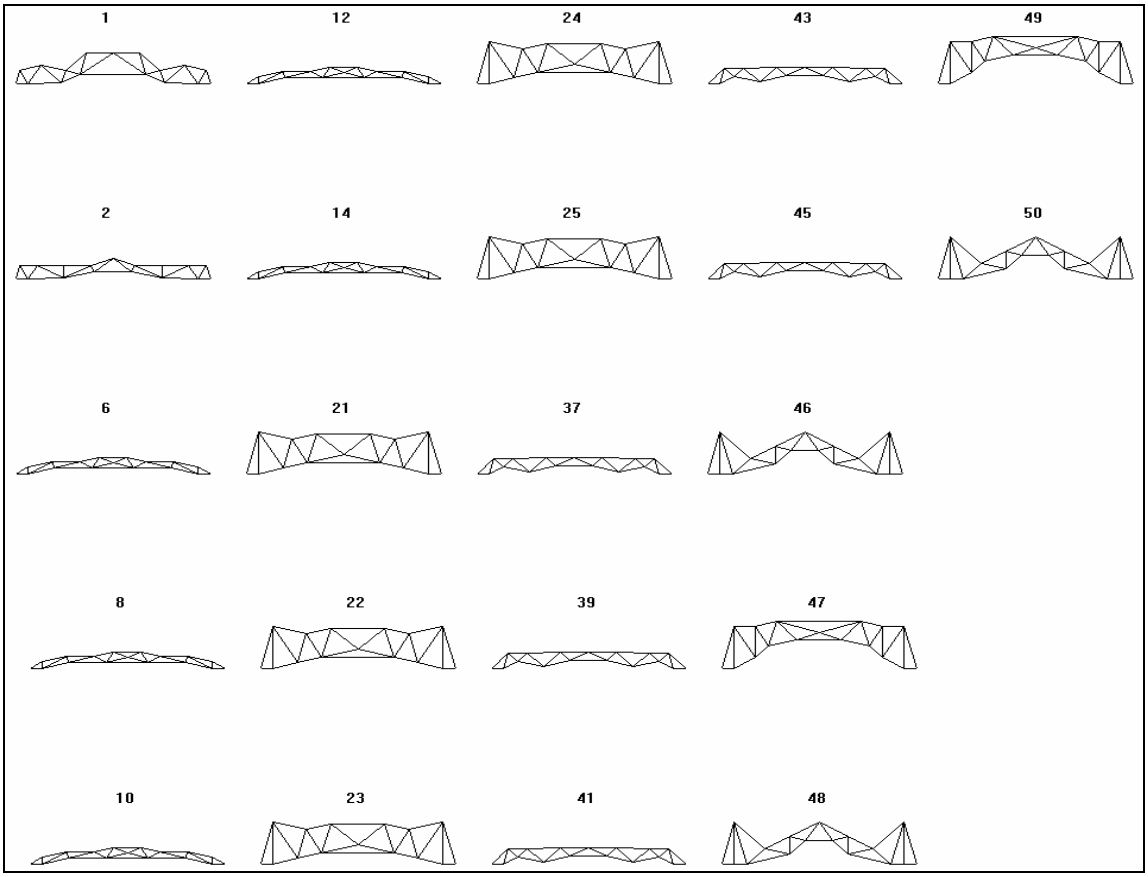


Figure 4.19. BPNN predictions for population S50M35(a) during input reproduction trials.

Figures 4.17 and 4.19 exactly reflect the user preferences shown in Figure 4.10. These figures correlate to predictions made by the KSOM and BPNN algorithms. In fact, both algorithms successfully reproduced the user's selections for every population in the input reproduction trials. The RSR algorithm, however, had difficulty reproducing inputs for the original set of truss populations. Although all topologies shown in Figure 4.18 were user preferences, several of the selected topologies were omitted. This behavior is typical of RSR results for populations S25M35(a) and S25M50(a), as well.

In populations S50M25(a) and S50M50(a), however, the method predicts all existing trusses (50 predictions each). This behavior indicates that RSR was unable to distinguish between preferred and non-preferred trusses in these populations. At least one user-selected group in both populations contained no favored features; that is, discernabilities for these groups were too small to differentiate them. Therefore, when a member of the second population, from which predictions were chosen, was compared to the favored features of each "good" group, at least one of the scores calculated would be zero. The RSR method implemented in this investigation uses the minimum score to indicate the user-selected group most similar to the current population member, which means that each member of the new population tied with an equally low score of zero.

Tables 4.2 through 4.4 summarize the percentages of correct predictions and selections predicted, as calculated using Equations 4.10 and 4.11. This tabulated data was derived from the above figures as well as those shown in Appendix C. Also included in these tables are the SSE results for the input reproduction trials. Recall that the topology generator used in this research creates duplicate truss topologies within the same population. Tables 4.2 through 4.4 show information about the number of unique topologies selected or predicted, rather than the total number of trusses selected or predicted. SSE values are normalized with respect to the total number of trusses, however, since this calculation did not account for recurring topologies. For example, in Table 4.3, thirteen topologies are listed as predictions for population S50M25(a) because this is the number of unique topologies among the predictions; the SSE value, however, is normalized by 50 because this is the total number of predictions made.

Table 4.2. KSOM Results for Input Reproduction Trials.

Population	SSE	Number of Selected Topologies	Number of Predicted Topologies	Number of Selections Predicted	Correct Predictions (%)	Selections Predicted (%)
S25M35(a)	0.16	8	8	8	100	100
S25M50(a)	0.08	11	11	11	100	100
S50M25(a)	0.06	5	5	5	100	100
S50M35(a)	0.09	7	7	7	100	100
S50M50(a)	0.15	5	5	5	100	100

Table 4.3. RSR Results for Input Reproduction Trials.

Population	SSE	Number of Selected Topologies	Number of Predicted Topologies	Number of Selections Predicted	Correct Predictions (%)	Selections Predicted (%)
S25M35(a)	0.19	8	6	6	100	75
S25M50(a)	0.06	11	6	6	100	55
S50M25(a)	0.50	5	13	5	38	100
S50M35(a)	0.07	7	3	3	100	43
S50M50(a)	1.33	5	35	5	14	100

Table 4.4. BPNN Results for Input Reproduction Trials.

Population	SSE	Number of Selected Topologies	Number of Predicted Topologies	Number of Selections Predicted	Correct Predictions (%)	Selections Predicted (%)
S25M35(a)	0.16	8	8	8	100	100
S25M50(a)	0.08	11	11	11	100	100
S50M25(a)	0.06	5	5	5	100	100
S50M35(a)	0.09	7	7	7	100	100
S50M50(a)	0.15	5	5	5	100	100

Performance for these trials was identical for both the KSOM and the BPNN. The RSR results showed the most significant variation, as Table 4.3 helps illustrate. For population S25M35(a), the SSE value for RSR is slightly higher than, but comparable to,

the values for the KSOM and BPNN. Populations S25M50(a) and S50M35(a), however, show SSE values for RSR that are slightly better than the other algorithms. The differences between these SSE values are small enough to be neglected, but the rough set reduct behavior for populations S25M50(a) and S50M50(a) cannot be ignored. Large SSE values and low correct prediction percentages highlight the fact that RSR does not effectively learn user preferences for these more complicated populations. RSR also failed to completely replicate all user selections, as shown by imperfect selections predicted percentages for populations S25M35(a), S25M50(a), and S50M35(a).

Preference Detection Trials

The KSOM, RSR, and BPNN predictions for the preference detection trials are displayed in Figures 4.20 through 4.22 for population S50M35(b). As in the input reproduction trials, these algorithms were trained to recognize user preferences from among the original set topologies (see Figures 4.7 to 4.11). The preference set topologies were then presented to each algorithm and predictions of user preference recorded.

In addition to showing the preference predictions made by each algorithm, Figures 4.20 through 4.22 also show Mathematica-generated plots of truss locations in feature space. Data points in red correlate to the user's selected preferences, while predicted preferences are in blue. Appendix C contains additional results for the populations not shown in the text.

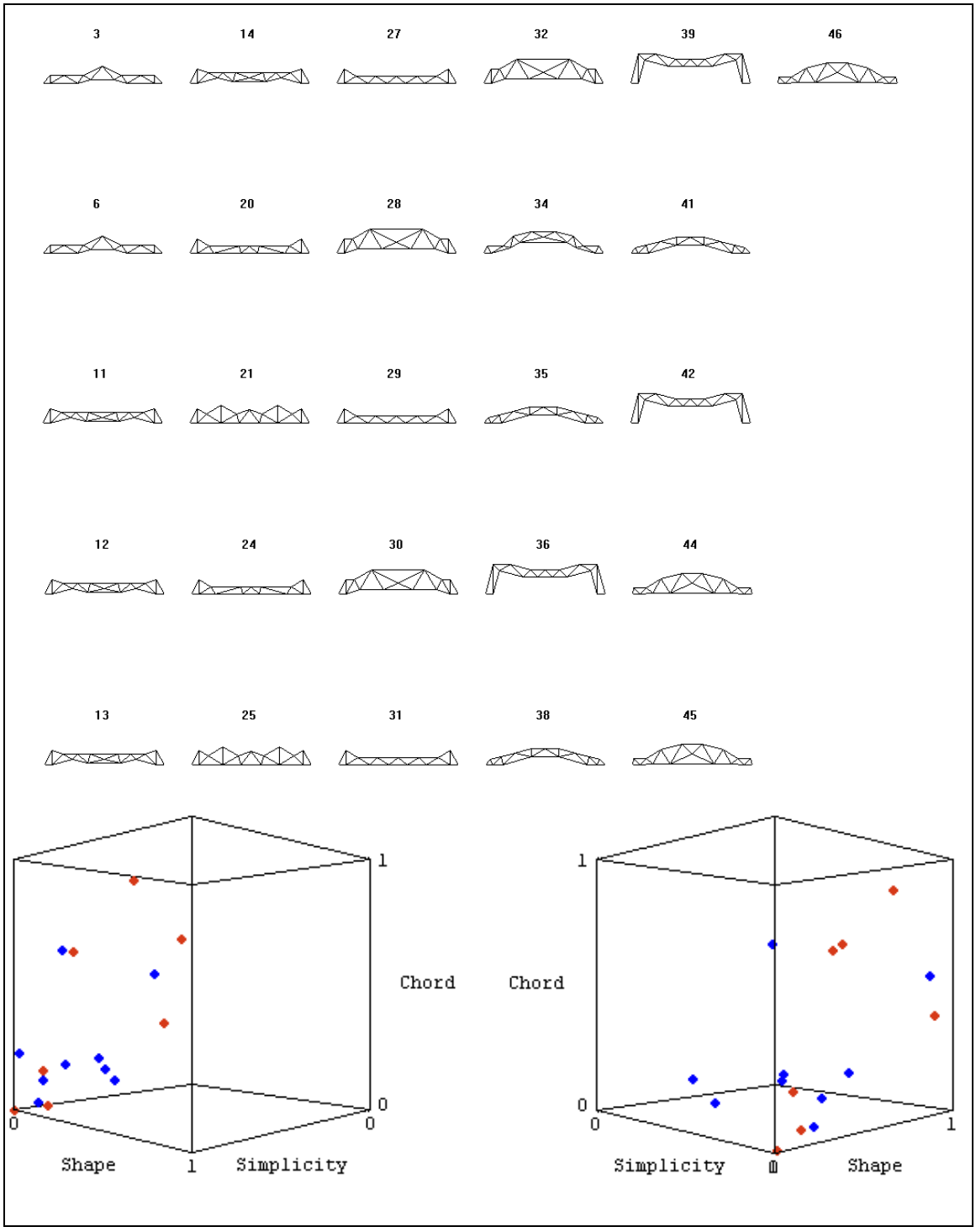


Figure 4.20. KSOM predictions for population S50M35(b) during preference detection trials.

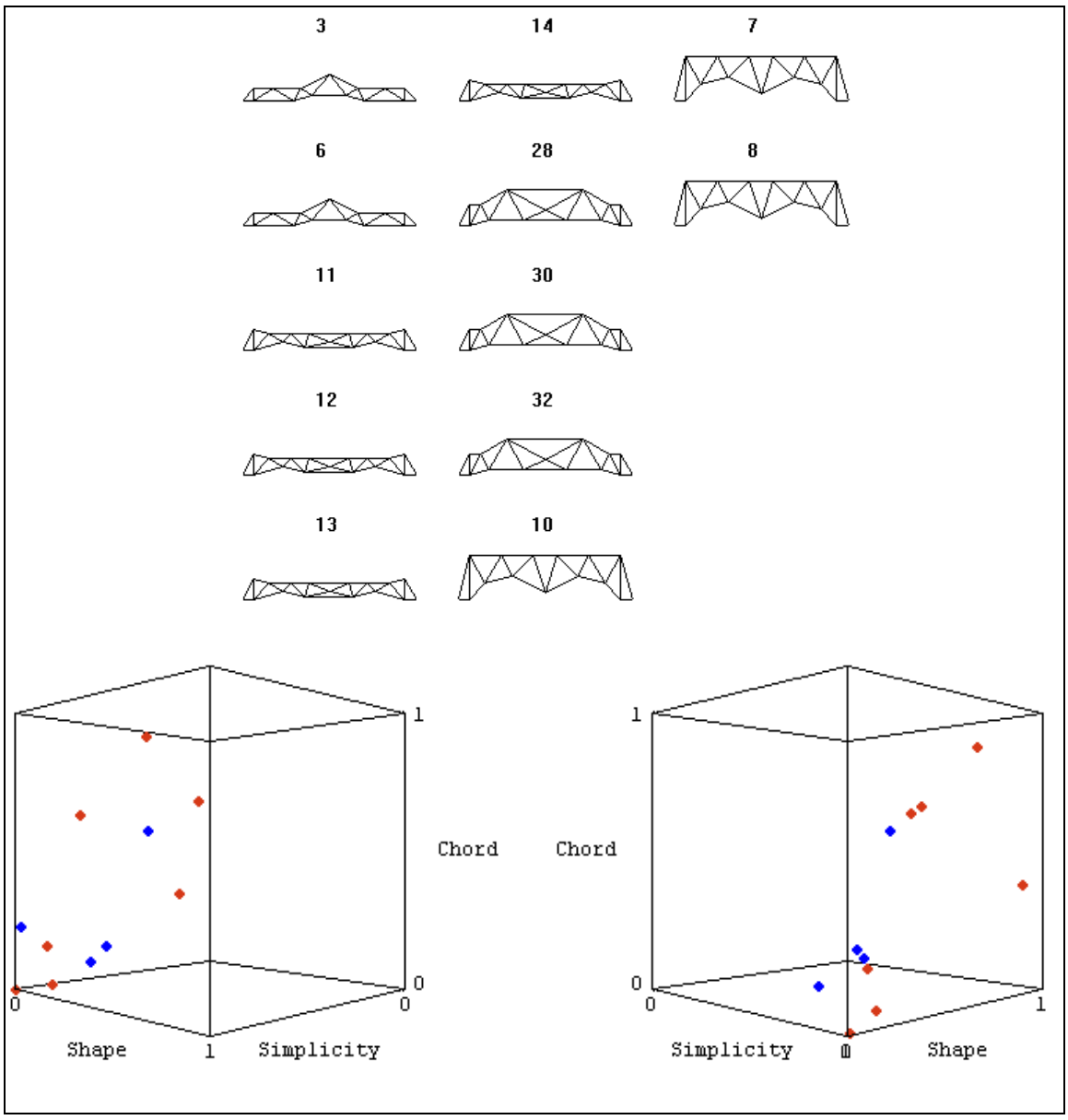


Figure 4.21. RSR predictions for population S50M35(b) during preference detection trials.

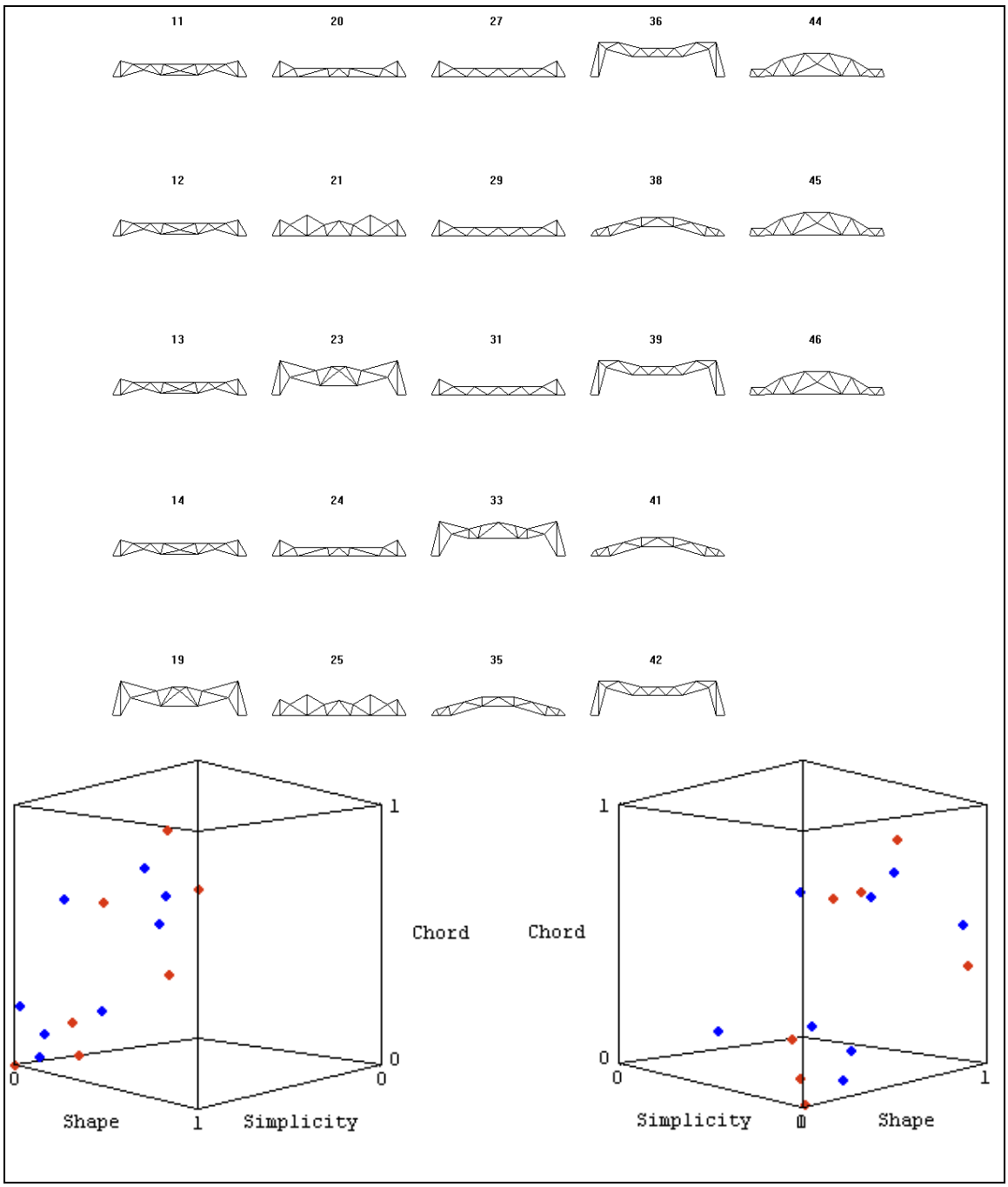


Figure 4.22. BPNN predictions for population S50M35(b) during preference detection trials.

Two populations were quickly pinpointed as sources of concern during these

trials. Population S50M50(b) was the first of these. The KSOM predicted two topologies for this population, and the BPNN could predict only one. While Mathematica plots reveal that these predictions did, in fact, lie close to the user's original selections, neither prediction was an actual preference. Therefore, both algorithms completely failed to identify preferences for this population. Conversely, RSR predicted all 50 topologies in the population, therefore hitting on the user's actual preferences but keeping the overall performance extremely low.

Poor performance occurs for the S50M50(b) population for a very simple reason: the trusses in this population and in S50M50(a) were so complex that there were very few likeable designs among them. Fewer user preferences were identified for S50M50(a) than for any other population, and S50M50(b) also had the smallest number of preferred and acceptable trusses. However, the topologies/topology proposed by the KSOM and BPNN are not wholly out of line. As illustrated by the three-dimensional graphs in Figures C.27 and C.29 of Appendix C, these predictions strongly resemble the user's selections, even more so than the user preferences shown in Figure 4.16.

Although performance for population S50M25(b) was significantly better than for S50M50(b), this population highlighted a different concern. Originally, groups 2, 3, and 6 (see Figure B.10) were selected from population S50M25(a) as inputs for the preference trials. Group 6 contains structures 21 and 48, which differ significantly from the selections in groups 2 and 3 and from each other. The user liked structure 48 but not structure 22. However, no mechanism existed for separating the two topologies. These two structures acted as outliers and skewed the preference analyses. This difficulty was

most pronounced in the RSR method, which again predicted all 50 topologies when it failed to discriminate user preferences.

It seemed likely that the user's preferences would be more accurately identified if group 6 was eliminated from the selections. Accordingly, a second set of trials was run for population S50M25(a) with the problematic topologies eliminated. The alternate inputs for S50M25(a) are shown in Figure 4.23.

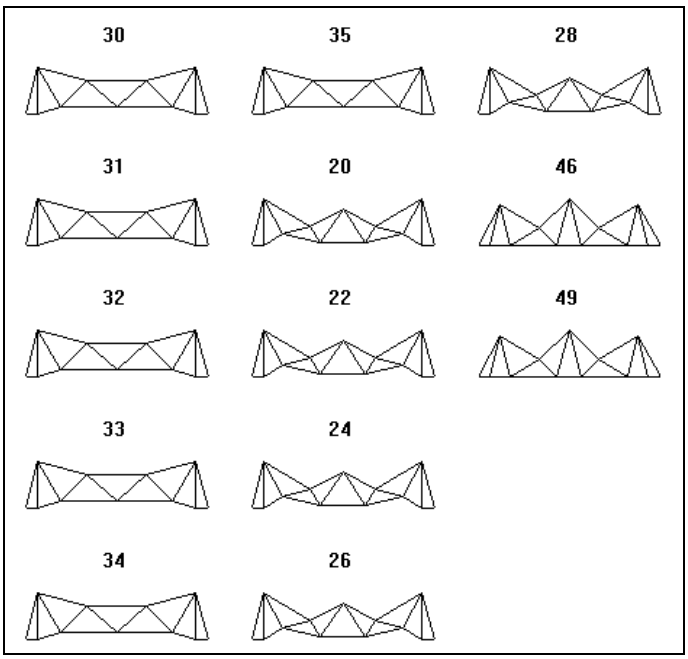


Figure 4.23. Alternate user selections for population S50M25(a).

The alternate user selections improved prediction rates for all three algorithms, as shown in Tables 4.5 through 4.7. The tables summarize the acceptable and unacceptable predictions and SSE values for these trials. The tables also illustrate how many user preferences, shown in Figures 4.12 through 4.16, were predicted by each algorithm.

Table 4.5. KSOM Results for Preference Detection Trials.

Population	SSE	Number of Preferred Topologies	Number of Acceptable Topologies	Number of Unacceptable Topologies	Acceptable Topologies (%)	Preferences Predicted (%)
S25M35(b)	0.76	2	1	2	60	50
S25M50(b)	0.55	0	3	2	60	0
S50M25(b)	0.91	3	0	2	60	75
S50M25(b)*	0.73	3	0	0	100	75
S50M35(b)	0.34	2	2	6	40	40
S50M50(b)	0.38	0	0	2	0	0

* S50M25(b) analyzed with alternate S50M25(a) selections.

Table 4.6. RSR Results for Preference Detection Trials.

Population	SSE	Number of Preferred Topologies	Number of Acceptable Topologies	Number of Unacceptable Topologies	Acceptable Topologies (%)	Preferences Predicted (%)
S25M35(b)	0.48	0	0	2	0	0
S25M50(b)	1.10	3	1	1	80	100
S50M25(b)	1.05	4	2	7	46	100
S50M25(b)*	0.94	2	1	0	100	50
S50M35(b)	0.58	1	1	2	50	20
S50M50(b)	1.33	2	1	20	13	100

* S50M25(b) analyzed with alternate S50M25(a) selections.

Table 4.7. BPNN Results for Preference Detection Trials.

Population	SSE	Number of Preferred Topologies	Number of Acceptable Topologies	Number of Unacceptable Topologies	Acceptable Topologies (%)	Preferences Predicted (%)
S25M35(b)	0.76	2	1	2	60	50
S25M50(b)	0.54	1	2	2	60	33
S50M25(b)	0.97	4	2	4	60	100
S50M25(b)*	0.80	4	1	1	83	100
S50M35(b)	0.50	2	1	6	33	40
S50M50(b)	0.35	0	0	1	0	0

* S50M25(b) analyzed with alternate S50M25(a) selections.

SSE values for the KSOM and BPNN are comparable. RSR SSE values are noticeably higher, in part because of RSR's tendency to predict all existing topologies. All three populations experienced difficulty with S50M50(b) predictions. RSR has particularly good performance for S25M50(b) and identifies more user preferences for this population than the other methods. RSR also has a strong performance for S50M25(b) with altered inputs. However, this performance must be balanced against the inability to discern predictions for all populations and limited number of predictions.

BPNN and KSOM have similar performance in terms of acceptable topology percentages. Populations S50M35(b) and S50M50(b) are points of particular difficulty, and S50M25(b) with altered inputs is a high point. The KSOM has a slight advantage over the BPNN for this criterion, but the BPNN has higher percentages of preferences predicted. This advantage indicates that the BPNN does a slightly better job of identifying a user's true preferences, rather than topologies that are merely acceptable. This added exploration, however, sometimes comes at the cost of an unacceptable prediction.

Figures 4.24 through 4.26 show relative percentages of preferable, acceptable, and unacceptable topologies for all three methods.

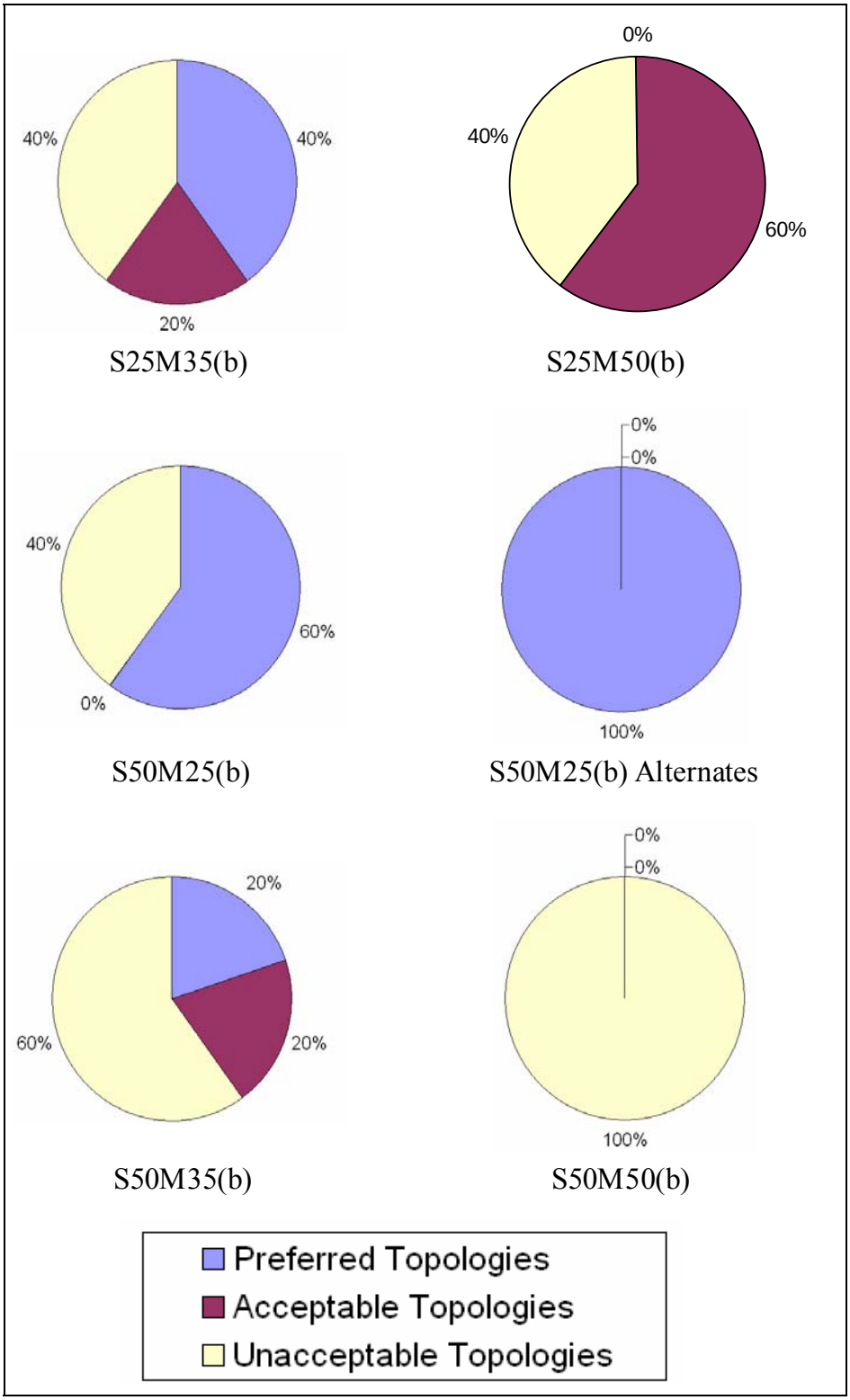


Figure 4.24. Summary of prediction accuracy using the KSOM.

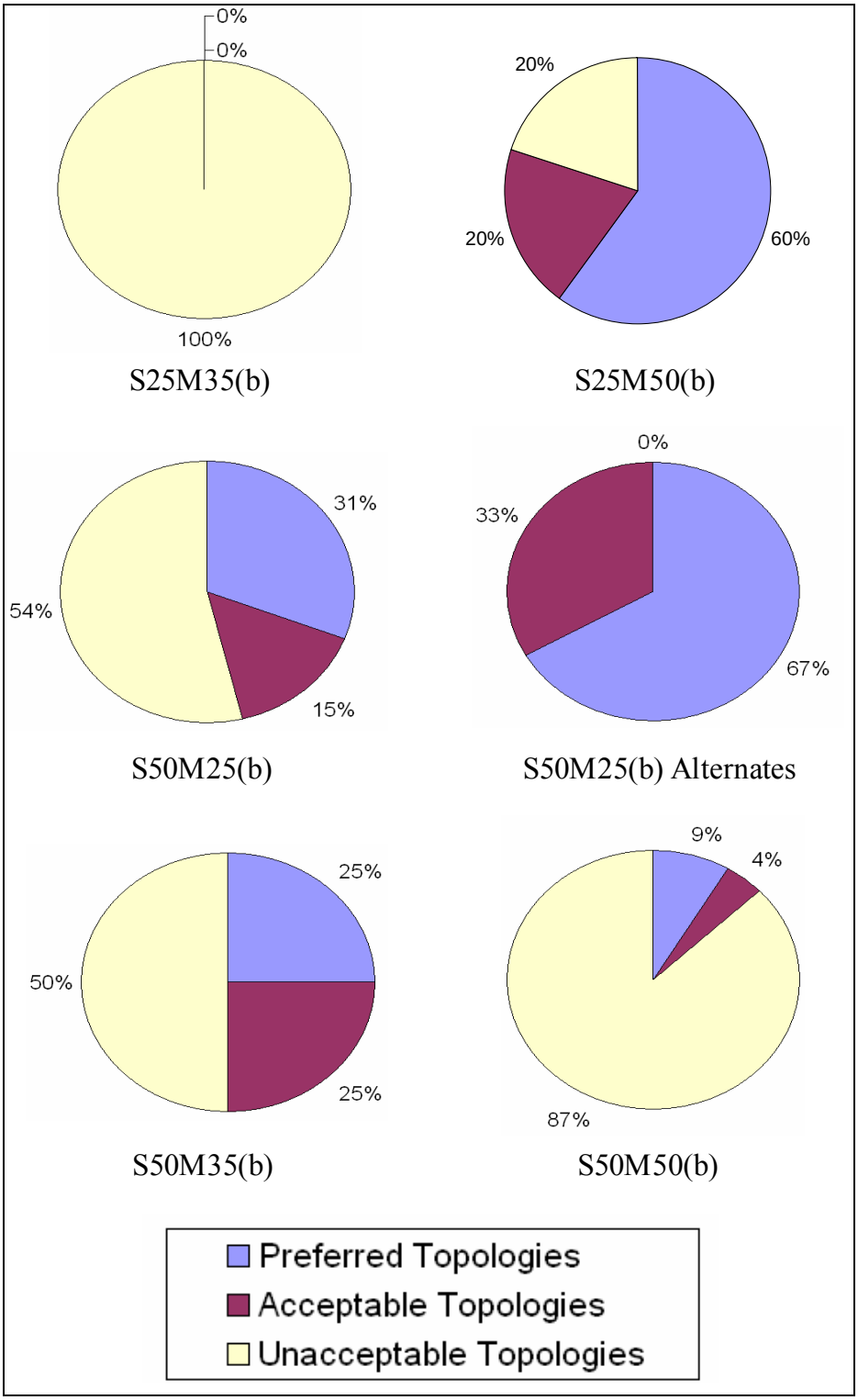


Figure 4.25. Summary of prediction accuracy using RSR.

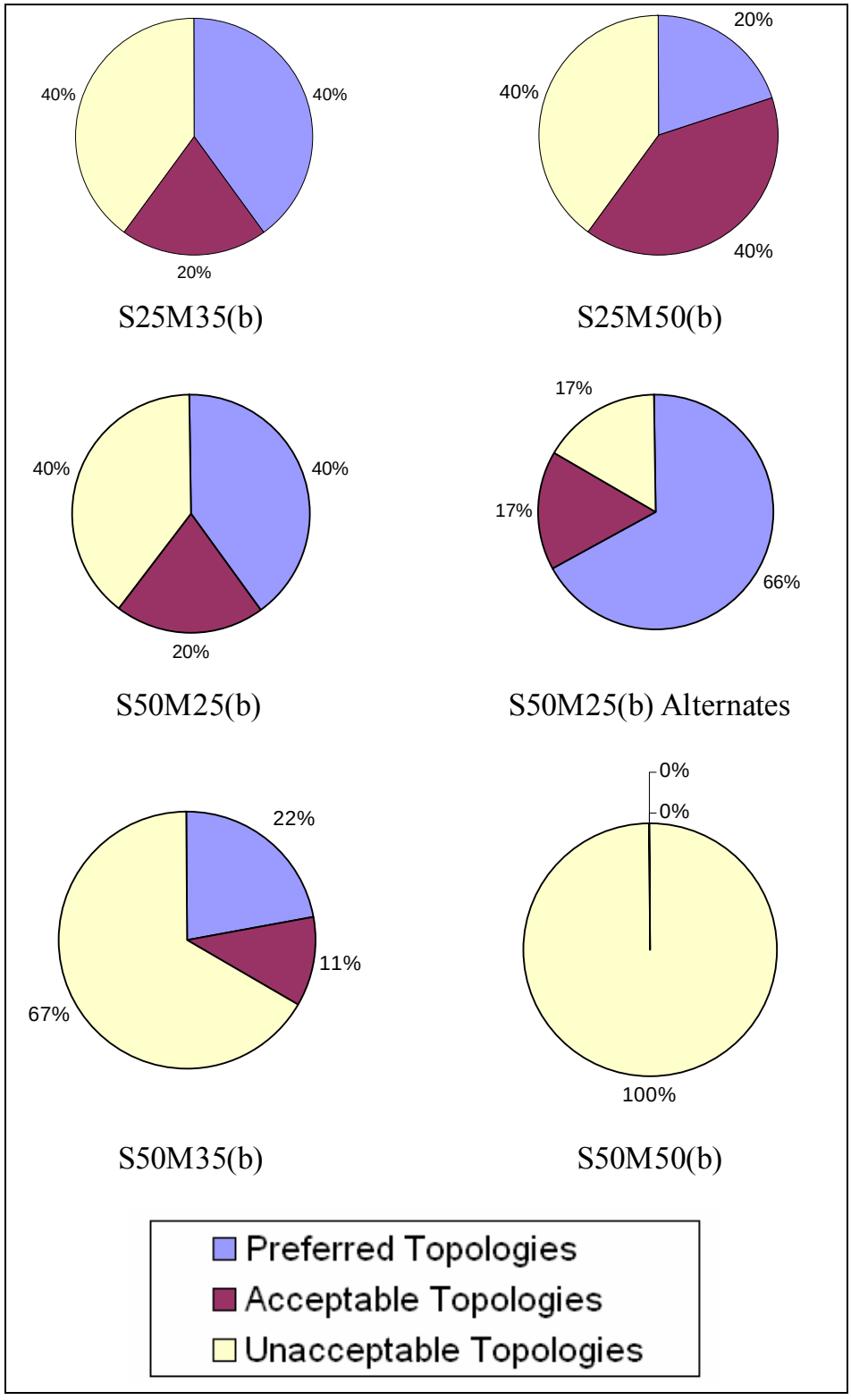


Figure 4.26. Summary of prediction accuracy using the BPNN.

Hybrid Back-Propagation with Rough Set Reduct Method

Results for both the input reproduction and preference detection trials indicate that both the KSOM and BPNN algorithms outperform RSR. The KSOM also appears to perform slightly better than the BPNN during preference detection. However, the ability of any of these methods to accurately and consistently predict a user's preferences has not been established by these trials.

For all three methods, the percentage of acceptable and preferred topologies among predictions is less than 50% for at least two populations. Moreover, although RSR cannot be selected as the final prediction mechanism because of its tendency to predict all topologies when no favored features have been identified, this algorithm has made some important predictions. For instance, RSR outperforms the other prediction methods for S50M35(b) and has 100% accuracy in predicting acceptable trusses with the alternate S50M25(a) inputs. RSR may not always discern what features a user favors, but the judgments it does make tend to be sound and occasionally identify relationships the other methods have missed.

The fact that prediction performance substantially improves when the user selections are altered for population S50M25(a) is important for two reasons. First, it reinforces the idea that a user's preferences can encompass divergent designs: for the most part, the angularity of designs in S50M25(a) drew the user's attention, but structure 48, though unique, was also interesting. Therefore, each group selected by a user may represent a completely different aspect of aesthetic appeal.

Additionally, the S50M25(a) population highlights a selection dilemma: Should a

group be selected in which one or more topologies are not favored? Structures 48 and 21 of population S50M25(a) are similar designs and look almost as if they are inverse images of each other. Yet, one of these images is more appealing than the other. Should the user select such a group, or ignore preferences that exist only in that group? Although the 1D KSOM successfully groups trusses based upon their appearance, the fact remains that there will be some groups of mixed preference.

Figure 4.27 illustrates this principle for population S50M35(b). In the 1D KSOM map created for this population, groups 2 and 5 contain only trusses that are preferences or second-choice selections. Groups 7 and 9, however, both display two topologies each, one of which is a user preference, and the other is among the unacceptable designs. The remaining preferences and acceptable topologies are placed in the overly large group 1, which contains even more unacceptable topologies.

Clearly, using the KSOM as the *only* method of analyzing a user's preference will lead to some erroneous judgments. In some cases, the topologies within a group must be considered on an individual basis, and the KSOM algorithm is not refined enough to handle this requirement. The BPNN offers a valuable advantage in addressing this problem in that previous training sets are available to supplement the user's current evaluations. That is, if desired, a user's prior aesthetic preferences can be used to increase the accuracy of the current predictions. Experiments with the BPNN indicated that reusing evaluations increased overall prediction accuracy, as did manually removing disliked trusses from the user's selections without regard for KSOM-defined groups.

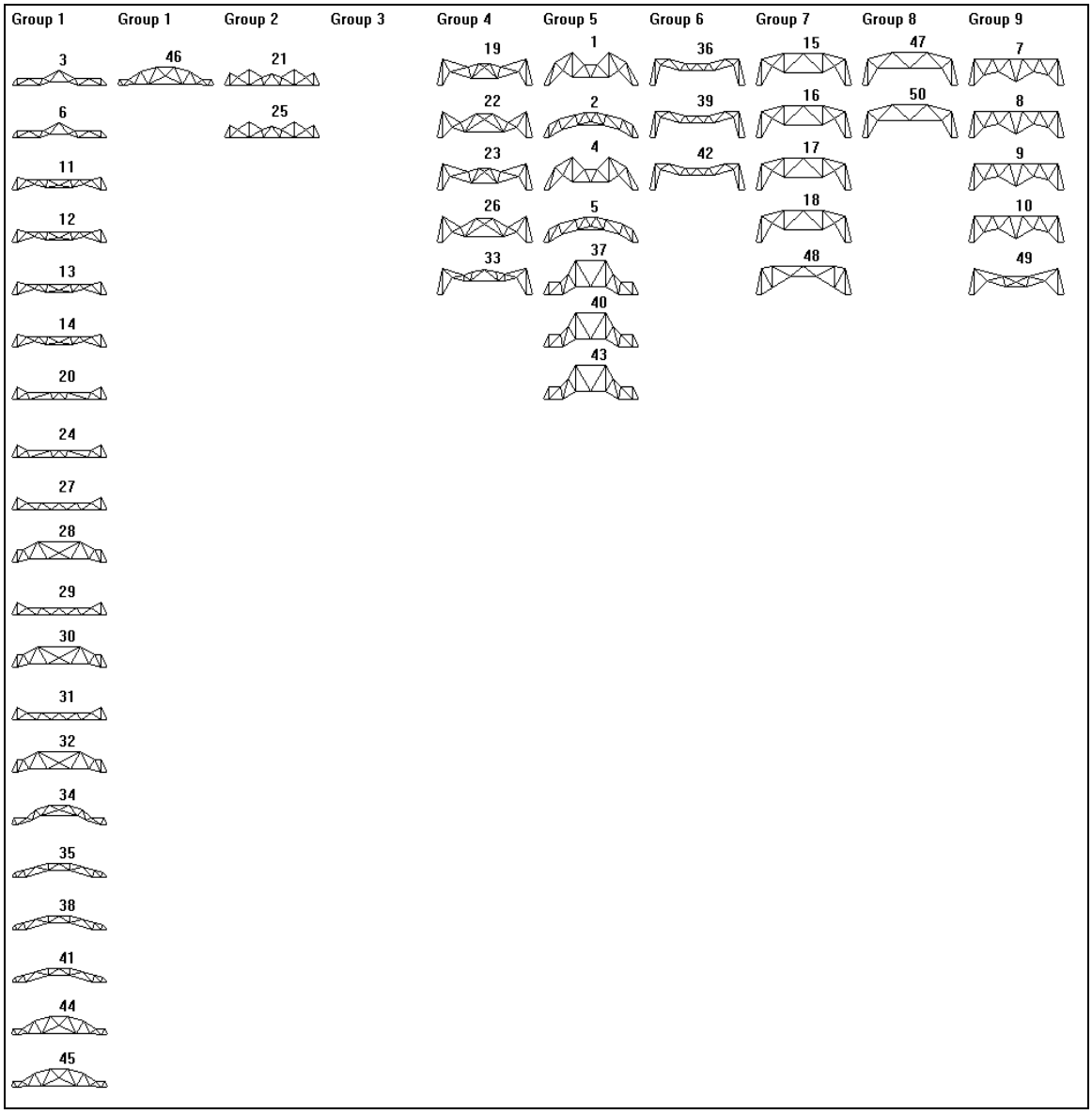


Figure 4.27. Clustering map for S50M35(b) using KSOM.

However, querying users to "prune" their selections counteracts the purpose of automatically grouping the trusses, which is to make the evaluation process as easy as possible on the user. Therefore, introducing an accurate prediction method that is also capable of evaluating the quality of input gathered from the KSOM selection process

seems advisable. To answer this need, a hybrid between the back-propagation neural network and the rough set reduct algorithm was created. This hybrid method attempts to combine the favored feature identification abilities of RSR with the BPNN's ability to model highly nonlinear relationships.

Description of the BP-RSR Method

The back-propagation, rough set reduct (BP-RSR) method begins by screening the user selections received from the KSOM classification. Instead of attempting to determine what distinguishes "good" trusses from "bad" trusses, the principle of rough set reduct is used to determine what distinguishes individual designs from all other selected trusses. For each truss selected by the user, the discernability for each characteristic of that truss is calculated by comparing values with all other good trusses; the smallest discernability for each feature is deemed characteristic of that particular design. This process leads to a vector describing the unusual characteristics of a truss in reference to the user's selections—that is, if a truss is abnormally tall compared to the user's other selections, it will have a high discernability for the height feature.

These vectors of unusual characteristics were then compared within KSOM groups. If all members of a group had a high discernability for a given feature, then it was assumed that the user had selected the group because of that feature. However, if a group member diverged significantly from other members in that group, it was eliminated from the user's selections. Eliminations occurred when a feature's discernability was more than 25% higher than either (a) all other members of the group, or (b) 75% of the group members, whichever was smaller.

Once these divergent topologies had been removed from the user selections, the characteristics of the remaining designs were used to create a training set for a BPNN. The back-prop network was trained, as before, to recognize user preferences.

This method was applied to the original and preference set populations and used to conduct additional input reproduction and preference detection trials. All parameters of the BPNN remained constant.

BP-RSR Results for the Input Reproduction Trials

Figure 4.28 shows the results for population S50M35(a) for the input reproduction trials using the BP-RSR algorithm. The topologies predicted by the BP-RSR method agree with the selections shown in Figure 4.10 almost perfectly. Structure 1, however, is missing from Figure 4.28. In the KSOM, structure 1 shares a group with multiple copies of structure 21. The user selected this group because structure 21 had favorable characteristics, even though structure 1 did not. Fortunately, structure 1 sufficiently deviated from structure 21 *as well as* the rest of the user's preferences to be eliminated.

This elimination was typical of BP-RSR eliminations in the user preferences. While not all of the trusses thus eliminated were non-pleasing designs (although many were), they all diverged significantly from the remaining trusses in their group. Therefore, it appears that rough set reduct is successfully choosing unusual or deviant topologies to eliminate. Table 4.8 presents SSE values and accuracy percentages to illustrate how these eliminations affected performance.

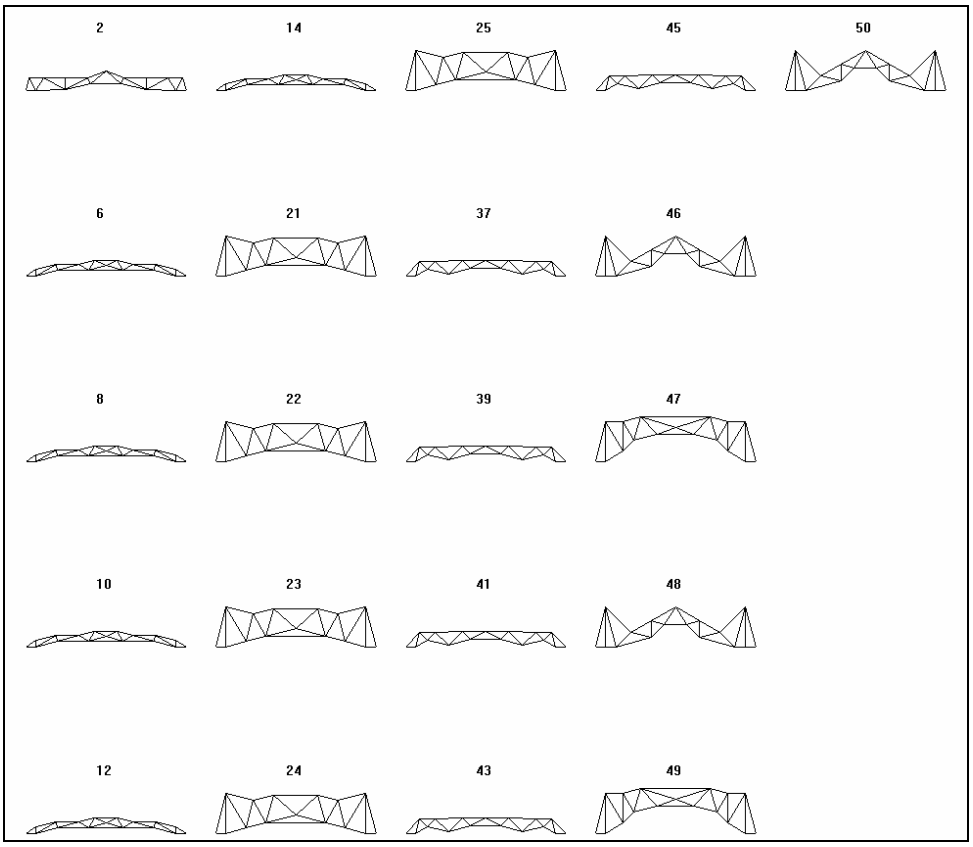


Figure 4.28. BP-RSR predictions for population S50M35(a) during input reproduction trials.

Table 4.8. BP-RSR Results for Input Reproduction Trials.

Population	SSE	Number of Selected Topologies	Number of Predicted Topologies	Number of Selections Predicted	Correct Predictions (%)	Selections Predicted (%)
S25M35(a)	0.13	8	6	6	100	75
S25M50(a)	0.07	11	10	10	100	91
S50M25(a)	0.03	5	4	4	100	80
S50M35(a)	0.06	7	6	6	100	86
S50M50(a)	0.15	5	5	5	100	100

By definition, the BP-RSR method analyses a user's inputs for irregular trusses

that may adversely influence its predictive ability. Therefore, it is not surprising to note that, while all predictions made were correct, not all of the selected topologies were predicted in these trials. For all populations except S50M50(a), one or more topologies were eliminated. As can be seen from the normalized-SSE errors, these eliminations improved the quality of results relative to the SSE values of KSOM and BPNN alone. This observation indicates that the remaining trusses lay closer to cluster centers, and the total error in the system decreased.

BP-RSR Results for the Preference Detection Trials

Figure 4.29 displays the BP-RSR preference predictions for population S50M35(b). Two of the user's preferences and one acceptable topology have been successfully identified, with three unacceptable topologies predicted. By contrast, the BPNN predicted six unacceptable topologies for the same number of preferred and acceptable designs. The KSOM found one more acceptable design at the cost of six total unacceptable trusses. Therefore, by removing structure 1 from the inputs sent to a back-propagation neural network, results for this population have substantially improved.

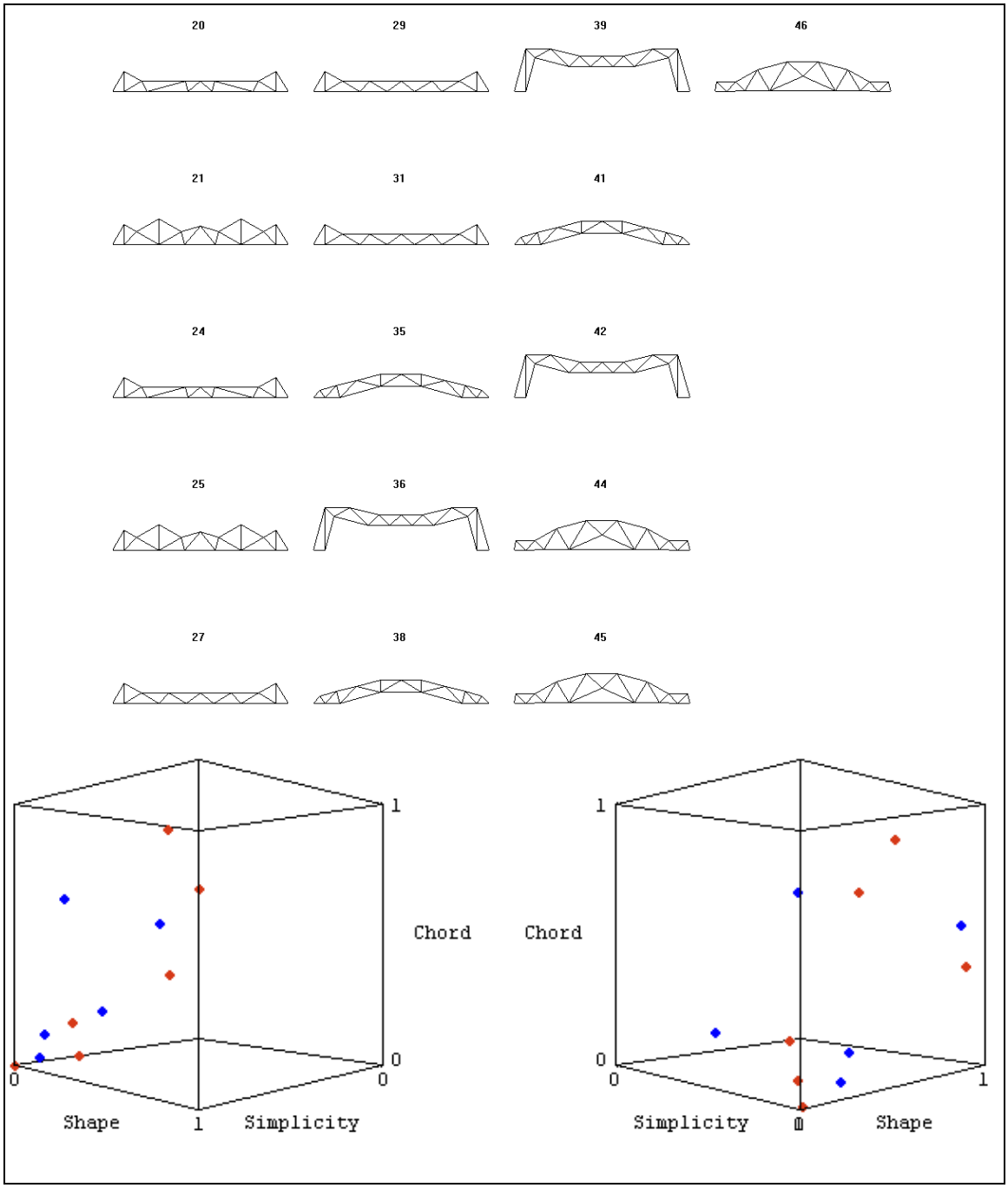


Figure 4.29. BP-RSR predictions for population S50M35(b) during preference detection trials.

Table 4.9 presents a numerical summary of the BP-RSR performance for all populations. The SSE errors for BP-RSR were larger than KSOM and BPNN errors for the 25-member populations and smaller for the 50-member populations. Unlike the previous two methods, BP-RSR was able to identify some preferences for population S50M50(b). The percentage of acceptable topologies in the BP-RSR predictions is always non-zero and is above 50% for all populations except S50M50(b), a problematic population for all techniques considered.

Table 4.9. BP-RSR Results for Preference Detection Trials.

Population	SSE	Number of Preferred Topologies	Number of Acceptable Topologies	Number of Unacceptable Topologies	Acceptable Topologies (%)	Preferences Predicted (%)
S25M35(b)	0.87	3	1	1	80	75
S25M50(b)	0.95	1	3	3	57	33
S50M25(b)	0.90	2	2	2	67	50
S50M25(b)*	0.75	2	2	1	80	50
S50M35(b)	0.35	2	1	3	50	40
S50M50(b)	0.17	1	0	4	20	50

* S50M25(b) analyzed with alternate S50M25(a) selections.

Overall, the BP-RSR method may not perform as well as one of the other methods considered for a given population. However, it does perform better, on average, across all populations. Figure 4.30 shows the ratios of acceptable, preferable, and unacceptable topologies.

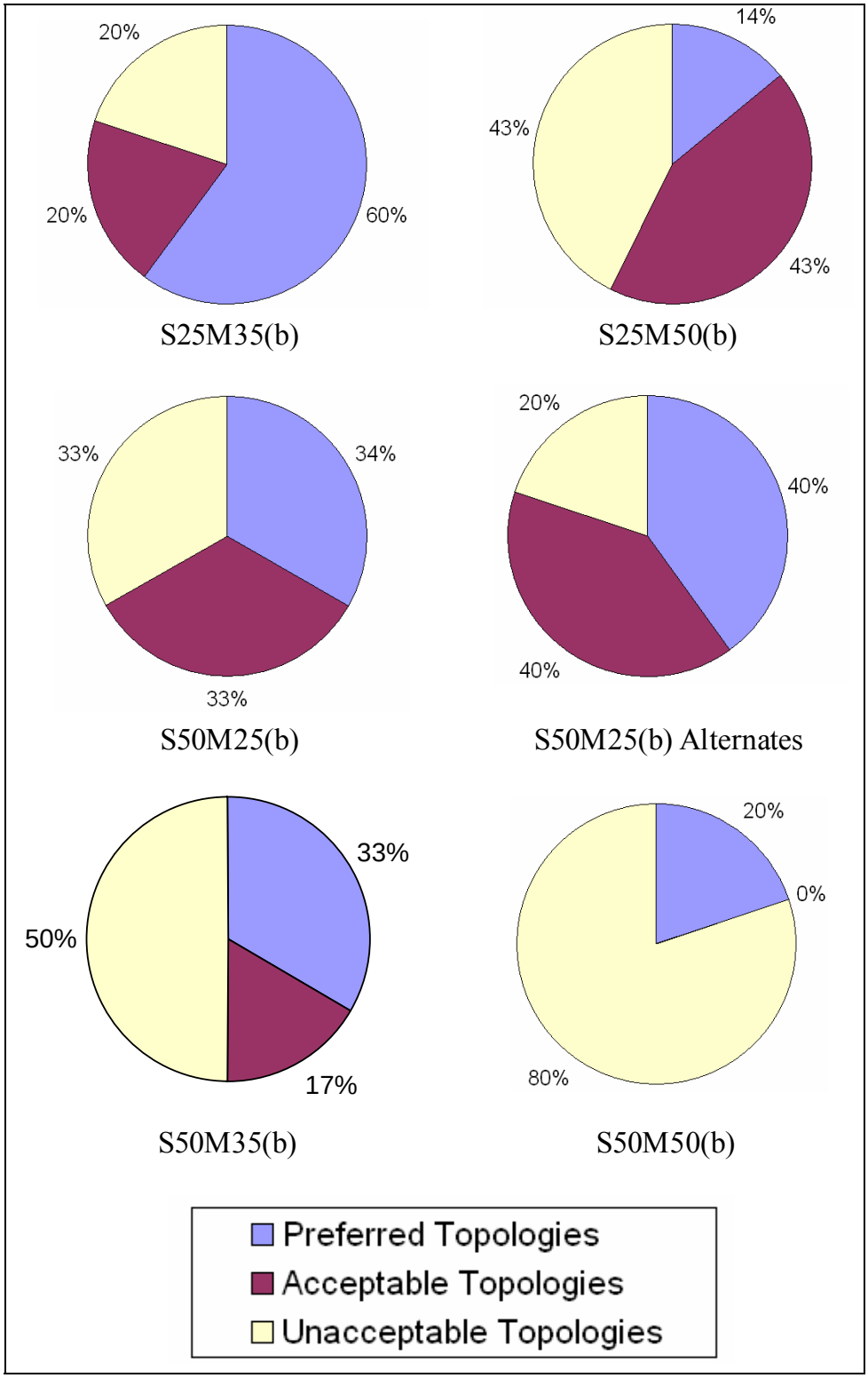


Figure 4.30. Summary of prediction accuracy using BP-RSR.

Selected Preference Detection Method

Kohonen's self-organizing map, rough set reduct, and back-propagation neural network algorithms were applied to the task of preference detection. These methods attempted to learn a user's aesthetic design preferences and apply this knowledge to predict preferences in previously unexplored populations. Input reproduction trials tested the ability of each method to identify user selections, and preference detection trials assessed the accuracy of preference predictions in unexplored populations.

These three algorithms had varying performance across the test populations, and each method exhibited different strengths. RSR had substantially better performance for several populations that presented difficulties for the KSOM and BPNN, which indicates that RSR was successful in determining "favored features" for some populations. However, RSR tended to output only a few predictions, thus giving a distorted view of user preferences, or to output all of the designs in a population, thus failing to detect preferences altogether.

Performance for the KSOM and BPNN were similar to each other, with the KSOM having slightly higher ratios of acceptable topologies. The BPNN countered this ability, however, by discovering more of a user's preferences. The BPNN was also able to evaluate truss appeal on an individual basis, while the KSOM was limited to evaluating truss groups as a whole. As the S50M25(b) analyses indicated, group-based predictions could lead to conflicting results.

Therefore, the hybrid back-propagation, rough set reduct technique was developed to combine the favored feature identification abilities of RSR and the pattern

recognition abilities of BPNN. This new method increased overall prediction accuracy to above 50% for all populations except S50M50(b). BP-RSR was able to identify one of the user preferences for this last population; RSR was the only other method to identify any of these preferences and did so only by outputting all possible topologies.

BP-RSR greatly increased prediction performance for those populations where the KSOM and BPNN struggled. In cases where it did not have the highest percentage of acceptable topologies, BP-RSR offered comparable results to most of the other methods. Therefore, the BP-RSR algorithm was identified as being an acceptably accurate method for detecting a user's preferences within a population.

CHAPTER V

PREFERENCE IMPLEMENTATION

The previous three chapters have detailed the creation of a method to detect and predict a user's design preferences among a population of potential roof truss topologies. The final phase of this investigation integrates this preference detection ability into an existing algorithm for multi-objective truss optimization. As mentioned in Chapter I, the truss optimization project seeks to create a comprehensive software tool for the conceptual design of large span roof trusses. Incorporating a user's preferences into the software will help to more fully represent the conceptual design process by modeling human judgments about the practicality, economic feasibility, and architectural desirability of a potential design.

This chapter explores different mechanisms for adding the user preference criterion to an algorithm concerned with the structural optimization of truss designs. The algorithm in question was developed in Paik (2005). Before presenting details of the preference incorporation, some background information about multi-objective optimization in general, and the current algorithm in specific, will be presented.

Background Information on Multi-Objective Optimization

The IRR GA implemented by Paik (2005) minimizes both truss weight and mid-span deflection. This multi-objective approach, where potentially conflicting goals must be weighed against each other, affects the way an optimization process is performed. Some authors have even gone so far as to say multi-objective optimization requires "...a

different definition of optimality, one that respects the integrity of each of our separate criteria" (Goldberg 1989).

Coello Coello (2001) provides a good definition of multi-objective optimization. The goal of any multi-objective process is to find a vector of solution variables that will optimize a vector function (comprised of individual objective functions) while satisfying problem constraints. Mathematically, this problem may be expressed as follows:

Find the vector $\mathbf{x^*} = (x^*_1, x^*_2, ...,x^*_n)^T$ of n decision variables which will satisfy m inequality constraints and p equality constraints, represented by Equations 5.1 and 5.2:

$$g_i(\mathbf{x}) \geq 0 \text{ for i=1, 2, ..., m} \quad (5.1)$$

$$h_i(\mathbf{x}) = 0 \text{ for i=1, 2, ..., p} \quad (5.2)$$

while optimizing the vector of k objective functions, $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), ... , f_k(\mathbf{x}))^T$ (Coello Coello 2001).

The vector **x*** which best satisfies the above equations will be the optimum solution. However, as pointed out earlier, the objective functions comprising the vector $\mathbf{f}(\mathbf{x})$ are often conflicting. This conflict means that it is unlikely that the optimization will converge to a single vector of design variables; rather, it is more likely that several **x*** vectors will be found. The notion of a Pareto optimum provides a "rational" way to redefine the best solution for situations where more than one solution vector is discovered (Goldberg 1989).

A vector **x*** is considered to be Pareto optimal if there is not another vector **x** in the feasible region (defined by $g_i(\mathbf{x})$ and $h_i(\mathbf{x})$) such that $f_i(\mathbf{x}) \leq f_i(\mathbf{x^*})$ for all k objectives *and* $f_i(\mathbf{x}) < f_i(\mathbf{x^*})$ for at least one i. In other words, a vector **x** becomes part of the Pareto

optimal set if it provides an overall solution to the problem at least as good as all other potential vectors *and* provides a better solution to at least one specific objective (Coello Coello 2001). Solutions in the Pareto set are called non-dominated; graphically, they represent points on the Pareto surface. Figure 5.1 illustrates this definition.

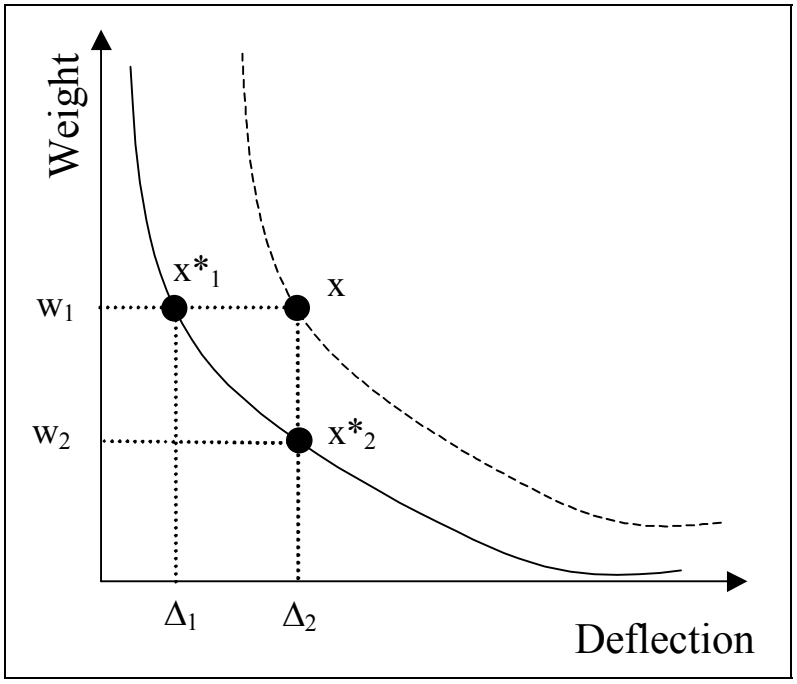


Figure 5.1. Graphic representation of Pareto optimality.

Three potential solution vectors (**x**, $\mathbf{x^*_1}$, and $\mathbf{x^*_2}$) are shown in Figure 5.1 for an optimization problem that seeks to minimize both weight and deflection. The non-dominated solutions for this population may be discovered by comparing values of the two objective functions. Solution **x** has a weight value of w_1 and a deflection value of Δ_2. By comparison, solution $\mathbf{x^*_1}$ has a weight of w_1, which is equal to **x**, and a deflection

of Δ_1, which is smaller than **x**. Therefore, $\mathbf{x^*_1}$ dominates solution **x**. Similarly, solution $\mathbf{x^*_2}$'s weight is w_2, which is less than **x**, and deflection is Δ_2, which is equal to **x**. Clearly, solution $\mathbf{x^*_2}$ dominates solution **x**, as well, because it satisfies all objective functions with equal or smaller values. Neither solution $\mathbf{x^*_1}$ nor solution $\mathbf{x^*_2}$ can be said to dominate the other because both present the best solution to at least one objective: $\mathbf{x^*_1}$ best satisfies deflection, and $\mathbf{x^*_2}$ best satisfies weight. These solutions are taken to be non-dominated, or Pareto optimal.

As indicated in Figure 5.1, curves may be drawn through the non-dominated solutions. These curves, or Pareto fronts, may be used to illustrate the trade offs inherent in the selection of one solution over another. Figure 1.4 represented one such curve for the truss optimization problem. However, multiple Pareto fronts may also be identified and used to indicate the relative ranking of subordinate solutions. This process is at the core of the multi-objective genetic algorithm (MOGA).

Once the non-dominated solutions have been identified in a population, the comparison process may be continued. The second-round comparisons seek to determine which of the dominated solutions would be Pareto optimal *if* the non-dominated solutions were removed from the population. The purpose of this and subsequent comparisons is to rank solutions within a population. The solutions originally identified as non-dominated receive a rank of one, while solutions selected by ignoring these original "optimal" points receive a rank of two, and so on until the population is completely ranked. Different algorithms exist for efficient ranking (Deb 2001), and the overall fitness of a solution will correspond directly to its assigned rank

(Coello Coello 2001).

Finding a set of Pareto optimal, or near optimal, solutions is, of course, the first priority of any multi-objective optimization method. However, solution diversity is a desirable secondary goal (Deb 2001). Diversity endeavors not only to find optimal solutions but solutions which differ significantly from each other, thus offering different glimpses into the trade off regions of the Pareto curve (Deb 2001). This solution variety is particularly important for conceptual design applications. The primary goal of the envisioned software is not to offer users a single, good alternative, but to allow selection of a final design from among a set of equals. Users may then explore how the design alternatives have changed geometrically or topologically in response to the conflicting objectives and select the design(s) best suited to their needs.

However, uniform fitness assessments, based solely on solution rank as outlined above, may lead to premature convergence of a population (Coello Coello 2001). Several techniques have been proposed to help foster diversity within the population of a genetic algorithm. Paik (2005) used two of these tools in the current algorithm: fitness sharing as proposed by Goldberg (1989), and the Strength Pareto Evolutionary Algorithm proposed by Zitzler and Thiele (Deb 2001).

Fitness Sharing

Goldberg (1989) proposed degrading an individual's fitness according to the number of similar individuals in its neighborhood. Similarity is measured using pre-selected "sharing functions," with the distance between solutions being calculated for given functions. This process is similar to the distance calculations used to classify

trusses based upon the characteristic feature vector developed in Chapter I. The closer two solutions lie to each other, the higher their sharing value, with self-associations having a sharing value of 1.

The fitness of an individual is altered as shown in Equation 5.3 (Goldberg 1989). The original fitness of individual x_i (assigned using rank or another method) is $f(x_i)$. The sharing function, s, values are calculated based upon the distance between x_i and all other members of the population, x_j. This summation is then used to define the reduced shared fitness, $f_s(x_i)$, as shown.

$$f_s(x_i) = \frac{f(x_i)}{\sum_{j=1}^{n} s(d(x_i, x_j))} \tag{5.3}$$

Fitness sharing adds to population diversity by encouraging the even spread of non-dominated solutions along the Pareto front. If many of the Pareto optimal solutions lie tightly clustered to each other, then the fitness of individuals in this cluster will be degraded. These solutions then become less advantageous when compared to other non-dominated solutions that lie in unexplored regions of the Pareto front.

Strength Pareto Evolutionary Algorithm

The Strength Pareto Evolutionary Algorithm (SPEA) was originally proposed by Zitzler and Thiele in 1998 and is briefly summarized in Deb (2001). The SPEA uses an external population to store non-dominated solutions. That is, this algorithm uses two populations: the mating population is allowed to evolve through normal GA processes, and the external population acts as a location to save any solutions that might lie upon the Pareto front. While the mating population will be replaced at each generation,

changes are only made to the external population if a newly discovered solution is non-dominated, in which case it is added to the external set, or dominates a solution in the external set, in which case the dominated solution is removed.

By preserving non-dominated solutions, the SPEA attempts to develop a robust set of individuals that are as close to Pareto optimality as possible. However, the external population also provides a way to encourage solution diversity. The number of population members allowed to remain in the external set must, of necessity, be limited. Solutions are spaced out according to groups identified by an agglomerative clustering mechanism that is similar to nearest neighbors but takes cluster averages into account.

In selecting parents for the next generation of solutions, individuals from both the external and mating pool may be considered. Fitness assessment in the SPEA relies upon the concept of a solution's "strength." First, the number of individuals in the mating pool that an individual in the external population dominates is tallied. This tally (n_i) is used to calculate the strength of the external solution (S_i) as shown in Equation 5.4 (Deb 2001). N refers to the total number of solutions in the mating population, so that strength is always less than 1.

$$S_i = \frac{n_i}{N+1} \tag{5.4}$$

For the mating population, an individual's fitness is calculated as one more than the summation of strengths for all external solutions that dominate the mating individual. This algorithm minimizes fitness. Therefore, solutions in the mating pool are, by definition, less fit than those in the external pool (Deb 2001).

Details of the Truss Optimization Algorithm

As mentioned earlier, Paik (2005) outlines the development of software for the optimization of large span roof trusses. This algorithm minimized truss weight and mid-span deflection, while accounting for kinematic stability, constructability limits on length, and allowable member stresses. Several fitness formulations were considered to accomplish this multivariate optimization. In order to place into its proper context the incorporation of user preference into this software, relevant details about Paik's (2005) work are presented here.

Design Variables

This algorithm makes use of the IRR GA as described in Chapter I. Therefore, the chromosome string describing an individual solution will contain sequences of design variables as well as redundant sections. No pre-defined grid of nodes or ground structure was used to influence the truss proposed by the GA. Truss designs were specified using randomly selected nodal coordinates. The design variable sequences contained information about these nodes and the members they connected and included the following parameters:

- X-coordinate of the node
- Y-coordinate of the node
- Number of members connected to the node
- Flags indicating steel W-shape selection for members connected to the node
- Priority value

Opposite end nodes for the specified members were selected by proximity; the

nearest two nodes with an open "member" slot were connected. In order to reduce the amount of memory needed to store an individual, symmetry was exploited. Each chromosome represented nodal information for half the desired truss span. During decoding, the full truss would be created by mirroring the nodes and members described in the chromosome string. Occasionally, nodal coordinates would overlap, and an individual would specify two or more nodes at the same location. In these cases, the priority value encoded in the chromosome was used to determine which node remained. For more details, see Paik (2005).

Architecture: Processes and Pools

Maintaining population diversity proved to be a difficult challenge for this application, and this concern led to the development of a multiple-process architecture.

The main process of the IRR GA is performed for a predefined number of generations. During the first of these generations, a population is created, and the stable trusses of that population are clustered according to perceived topological similarities. Paik (2005) used the number of members, number of nodes, and the total member length of a truss to pair it with similar designs. Each group of designs specified in this way is then moved into a sub-process of its own.

All of the GA processes of selection, crossover, and mutation occur in the sub-processes. However, instead of a randomly created first generation of designs, the group of similar topologies identified by the main process is used to "seed" the GA. Once the desired number of generations for this sub-process is complete, results for the final population are sent back to the main process, and a new sub-process begins for the next

design group determined by the main process.

A generation of the main process completes when each of the groups identified at the beginning of the generation have been subjected to analysis in a sub-process. Groups are then created based upon the final generation topologies from the sub-processes. This architecture helps ensure that different topological types are considered throughout the analysis. Even if a group of trusses performs poorly in an early generation of the main process, performance will likely improve during the sub-process, thus allowing the group to eventually contend for a higher rank among the total population.

As with the SPEA, Paik (2005) used external populations in both the main and sub-process to keep track of non-dominated solutions. Paik (2005) also added one further population, which kept track of all unique topologies discovered during the course of the IRR GA. As before, the inclusion of these extra populations was intended to increase overall population diversity and prevent premature convergence of the Pareto front to a few topologies. Descriptions of the populations used in the main and sub-processes are shown below:

- Rank Pool: Used in the **main process** to save non-dominated individuals; based upon the SPEA.
- Save Pool: Used in the **main process** to accumulate individuals placed in the rank pool of the sub-processes. These designs were the non-dominated individuals found for specific topological groups and were used to seed the sub-processes.
- Backup Pool: Used in the **main process** to save unique topologies.

- mating pool: Used in the **sub-process** for GA operations. Seeded by Save Pool to consider a specific topological group.
- rank pool: Used in the **sub-process** to track the non-dominated solutions for the topological group. Based upon the SPEA.
- backup pool: Used in the **sub-process** to save unique topologies.

Fitness Assessment and Selection

Selection occurs during the sub-processes and begins by computing rank. A point-by-point comparison is made between the individuals in the rank and mating pools using the **dominate( )** function. When called, **dominate( )** compares two individuals based upon values for the following objectives:

- Deflection
- Weight
- Stability (**penalty2**; degree to which kinematic determinancy is violated)
- Short members (**penalty3**; summation of member lengths less than eight feet)
- Long members (**penalty4**; summation of member lengths more than twenty feet)
- Stress (**penalty5**; summation of member stress violations)

The function returns a decision about whether one individual may be said to dominate the other. The number of solutions an individual dominates, as determined in this manner, is counted. All individuals begin with a rank of one, and their rank is increased each time they are dominated by another solution. Once all individual pairs have been examined, the individuals are sorted according to rank. The variable **rankFitness** is used to sort individuals within rank and will be defined shortly.

The above procedure describes the action of the function **computeRank( )** in the sub-processes. This sub-process function has an analog in the main process. Both functions are used to rank individuals in the solution pools. In cases where two individuals submitted to **dominate( )** are *tied* for all objective values, then a composite fitness function is used as a "tie breaker." Composite fitness functions are sometimes used to collapse multi-objective optimization criteria into a single objective. Two variables, **fitness** and **rankFitness**, apply this principle as shown in Equations 5.5 and 5.6.

$$\text{fitness} = \frac{100 - \dfrac{\text{weight}}{\text{max weight}} - \dfrac{\text{deflection}}{\text{max deflection}}}{1 + 10\left(\dfrac{\text{penalty2}}{\text{max2}} + 10\left(\dfrac{\text{penalty3}}{\text{max3}} + \dfrac{\text{penalty4}}{\text{max4}}\right) + 5\left(\dfrac{\text{penalty5}}{\text{max5}}\right)\right)} \tag{5.5}$$

$$\text{rankFitness} = \frac{100 - \alpha\left(\dfrac{\text{weight}}{\text{max weight}}\right) - \beta\left(\dfrac{\text{deflection}}{\text{max deflection}}\right)}{1 + 10\left(\dfrac{\text{penalty2}}{\text{max2}} + 10\left(\dfrac{\text{penalty3}}{\text{max3}} + \dfrac{\text{penalty4}}{\text{max4}}\right) + 5\left(\dfrac{\text{penalty5}}{\text{max5}}\right)\right)} \tag{5.6}$$

In these composite forms, the two major structural objectives (weight and deflection) are separated from the constraints (penalty2 through penalty5, which correspond to the stability, length, and stress considerations listed previously). Each objective and penalty is normalized by the population corresponding maximum. The constants used in these expressions were determined empirically (Paik 2005). Note that **fitness** and **rankFitness** differ only by the inclusion of α and β in the **rankFitness** definition. These variables fluctuate randomly throughout the IRR GA and are used to alter the ratio between the weight and deflection objectives. This alteration is intended

to help explore different regions of the Pareto front.

The **computeRank( )** function in the main process uses **fitness** as a tie-breaker in the assignment of rank. The **fitness** variable is also used to sort the Rank and Backup pools in the main process. When individuals need to be removed from either pool to conserve space, the lower **fitness** individuals will be removed first. In the sub-processes, **rankFitness** is used as both tie-breaker and sorting criteria. Sorting the Save pool in the main process also relies upon **rankFitness**.

Once all population members in the sub-process pools have been ranked, tournament selection begins. The number of individuals involved in the tournament may be altered by user input, but the default value is four. Two individuals are selected at random, and **dominate( )** is used to determine dominance. The dominant individual is then compared to the next solution in the tournament using **dominate( )**. This process continues until the desired number of comparisons is reached, and the winning solution is selected to become a parent of the next generation of truss designs.

If **dominate( )** does not produce a clear decision between two potential solutions, then a "total fitness" value, calculated using Equation 5.7, is used to make the decision. The member with the higher total fitness, F_{TOT}, value is selected. This fitness calculation relies upon the fitness sharing and strength concepts discussed earlier in the chapter, as well as the penalty/constraint terms.

$$F_{TOT} = \frac{1}{F_{share}\ P_{TOT}\ F_{strength}} \tag{5.7}$$

The sharing fitness, F_{share}, is calculated in accordance with the fitness sharing

proposed by Goldberg and expressed in Equation 5.3. The sharing value for a pair of individuals depends on the number of elements, number of nodes, and total member length in both designs. The total penalty term, P_{TOT}, is the combination of constraint terms shown in the denominator of Equations 5.5 and 5.6. This variable is named **compositePen**. Lastly, the strength fitness, $F_{strength}$, is calculated, in accordance with Equation 5.4, as the ratio between the number of individuals a solution dominates and the total number of individuals in the relevant pools.

Methodology for Selecting a Preference Implementation Method

Several mechanisms were considered to embed user preferences (as modeled with the tools described in Chapters II through IV) into the optimization software outlined above. The functions and variables that comprise the fitness assessment and selection process for Paik's (2005) original algorithm were altered in order to accommodate this new criterion. The format for and location of these alterations were selected during the mechanism trials, while the preference emphasis trials examined how user preferences altered the population.

Although user preferences were applied to the IRR GA using different mechanisms, the method for determining these preferences remained constant. When the first generation of the IRR GA's main process was complete, an input file was created listing 100 stable and unique truss topologies. A separate program was run to classify and record user selections of these topologies. Control then returned to the main program, where the user's inputs were analyzed using the back-propagation with rough set reduct algorithm developed in Chapter IV. The weights for the trained network were

recorded. For each new generation created during the sub-processes, the neural network was reinitialized with these weights and used to predict whether or not the user would prefer a given individual. The results of these predictions were stored as the **preferred** attribute for an individual. A **preferred** value of two correlated to a prediction that the user would not like the truss, and a value of one indicated that the user would.

Most IRR GA simulations used in the present work consisted of ten generations for the main process and 100 generations for each sub-process. This combination of generations was the smallest allowed by Paik's (2005) algorithm, with 100 main process and 300 sub-process generations being more typical. The simulations for this work were deliberately shortened to return results in less than one day.

Mechanism Trials

Three unique truss populations were created during early runs of the IRR GA trials. Topology files for each population type were saved when first created so that they could be reused in later simulations. The search space and genetic parameters were varied across these populations in order to create truss groups that illustrated different levels of complexity and topological variation. The populations may be summarized as follows:

- **60by15**: 60-foot span trusses with a maximum height of 15 feet; default crossover (0.3) and mutation (0.005) parameters.
- **60by15Par**: 60-foot span trusses with a maximum height of 15 feet; increased crossover (0.6) and mutation (0.010) parameters.
- **40by15**: 40-foot span trusses with a maximum height of 15 feet; default

crossover (0.3) and mutation (0.005) parameters.

Paik's (2005) thesis considered test cases of 40, 60, and 80-foot spans with maximum 15-foot heights, and these test cases inspired the selection of spans and height used in the present trials. Preference selections for the three populations remained constant throughout the mechanism trials and are shown in Figures 5.2 through 5.4.

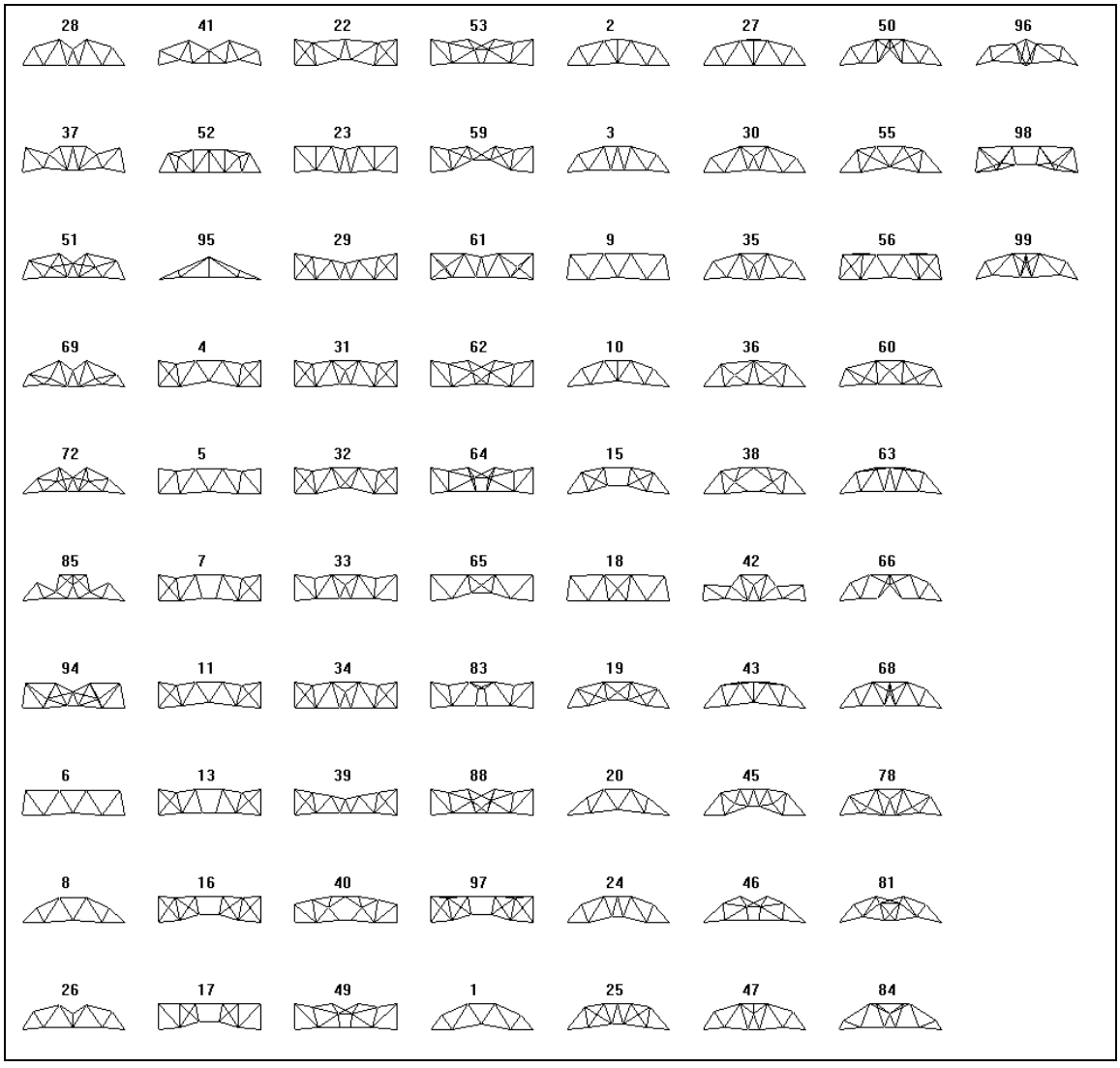


Figure 5.2. User preferences for population 60by15 during mechanism trials.

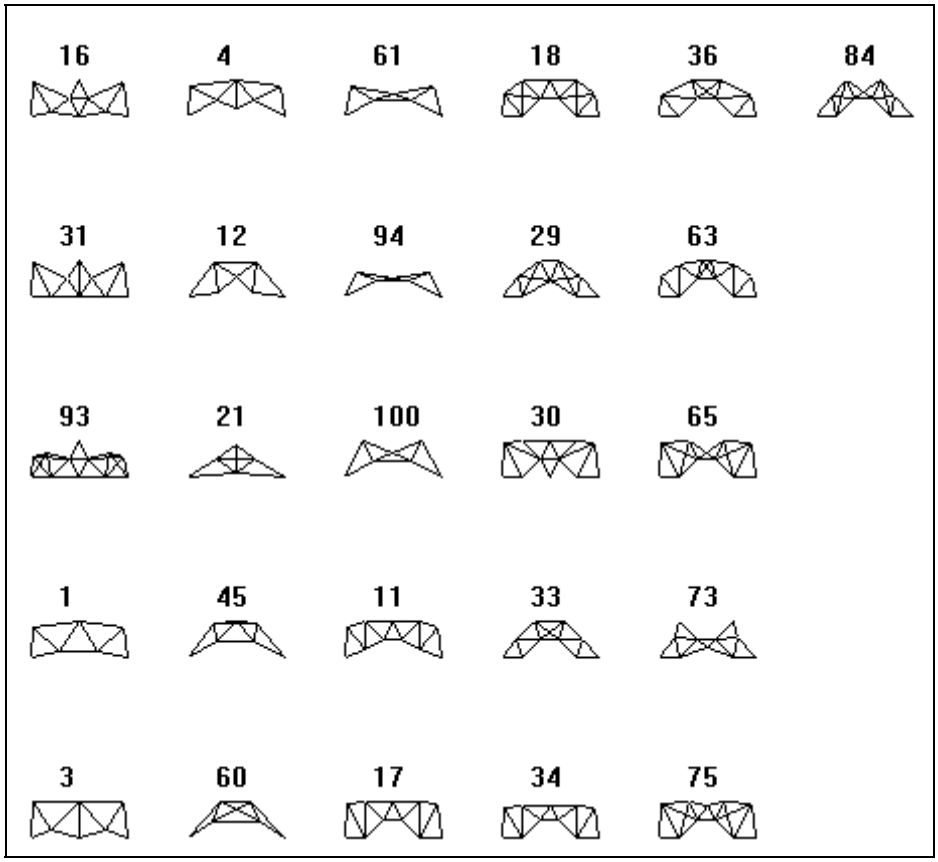


Figure 5.3. User preferences for population 40by15 during mechanism trials.

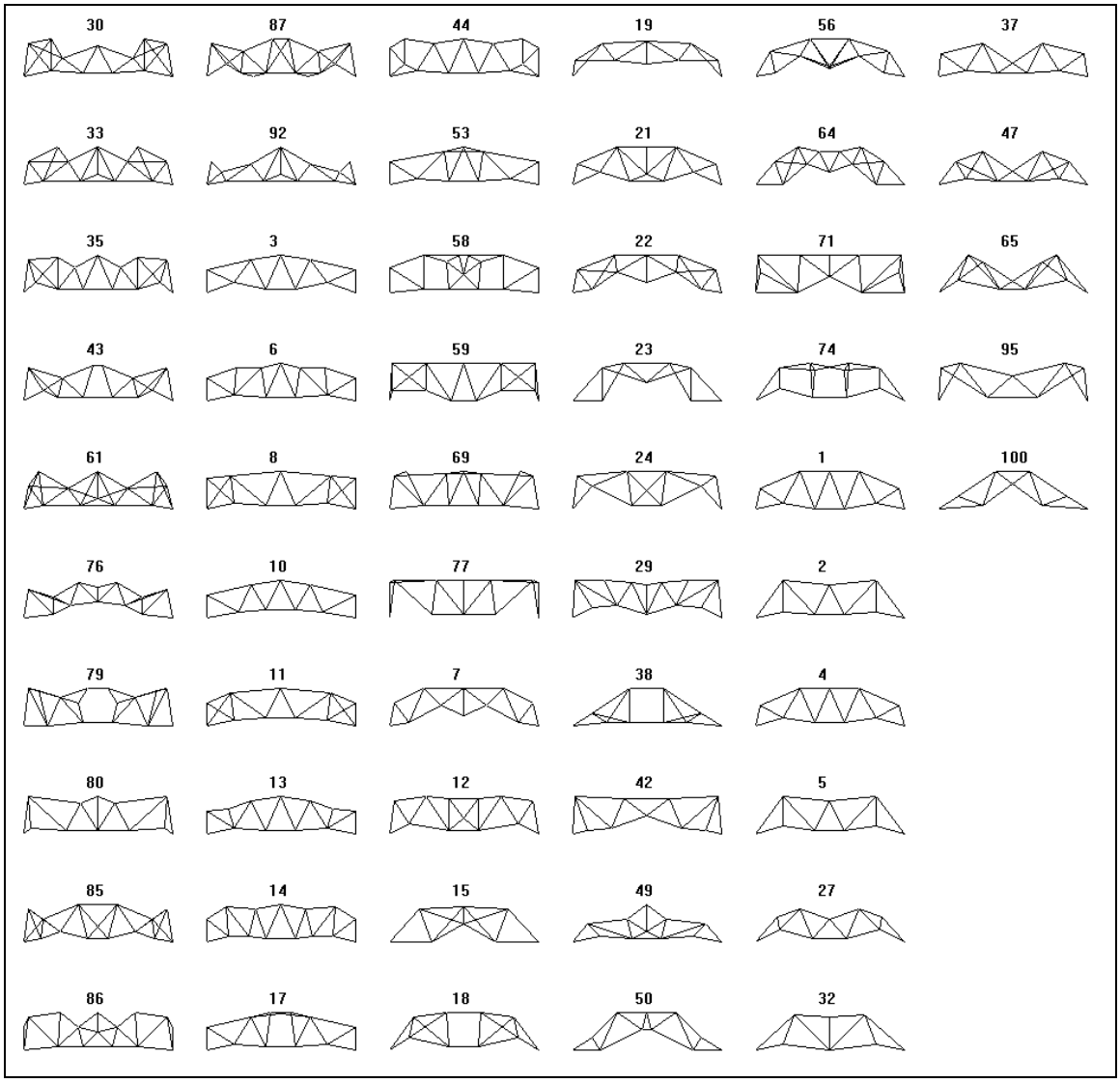


Figure 5.4. User preferences for population 60by15Par during mechanism trials.

A review of the fitness and selection procedures of Paik (2005) suggests three key places where the addition of user preference information might affect the optimization of the IRR GA.

1. The **preferred** attribute could be added to the **dominate( )** functions of both the main and sub-processes. This addition would then include the preference

objective in determining solution dominance, thus affecting both the ranking and selection processes.

2. User preference could be included in the definitions of **fitness** and **rankFitness**. Recall that these variables are used to break ties during ranking of the main and sub-processes as well as to sort solutions in the external populations for removal.
3. The **preferred** attribute could also be incorporated into the **compositePen** variable. This variable formed part of the total fitness assessment for a truss, which was used to break ties during the selection process.

When incorporating user preference into one of the "composite fitness" formulations, i.e. **fitness**, **rankFitness**, or **compositePen**, the **preferred** attribute was treated as an additional penalty. This treatment reflected a decision not to treat preference with equal emphasis as weight and deflection. In part, this decision was made so as to preserve the rigorous structural requirements placed upon the problem. However, treating preference as a penalty also allowed for a consistent formulation for all three variables. The denominator of Equations 5.5 and 5.6 modeled how to incorporate preference into these terms using the P_{TOT} definition shown in Equation 5.8. The variable C represents an integer multiplier to be determined experimentally. The **preferred** attribute is divided by three because all trusses in a population are originally set to a preference value of three, so this denominator has a normalizing effect.

$$P_{TOT} = 1 + 10\left(\frac{\text{penalty2}}{\text{max2}} + C\left(\frac{\text{preferred}}{3}\right) + 10\left(\frac{\text{penalty3}}{\text{max3}} + \frac{\text{penalty4}}{\text{max4}}\right) + 5\left(\frac{\text{penalty5}}{\text{max5}}\right)\right) \quad (5.8)$$

Since **dominate( )** and **compositePen** both directly contribute to the selection

process, these locations were considered the most important areas to include user preferences. Most of the mechanism trials, therefore, sought to empirically evaluate different variable formulations for these locations. In total, six IRR GA simulations were run for each of the three populations. These simulations investigated the following methods for incorporating user preference:

- **Baseline**: The IRR GA was run without any user preference information.
- **Dominate**: User preferences were incorporated through **dominate( )** only.
- **CompositeOnly**: User preferences were incorporated only through the **compositePen** definition; a multiplier of 10 (C=1) was used.
- **Composite10**: User preferences were incorporated through both **dominate( )** and **compositePen** with a multiplier of 10 (C=1).
- **Composite50**: User preferences were incorporated through both **dominate( )** and **compositePen** with a multiplier of 50 (C=5).
- **Fitness**: User preferences were incorporated through **dominate( )**, **compositePen** with a multiplier of 10 (C=1), and **fitness/rankFitness** definitions.

Preference Emphasis Trials

The mechanism trials sought to determine *where* and *how* a user's preferences could be connected to the truss optimization program. In the preference emphasis trials, the goal was to determine how *effective* this connection was in guiding the IRR GA towards designs that were more aesthetically pleasing.

At the end of the mechanism trials, it was determined that final generation rank one trusses for population 60by15 were extremely similar to user preferences as well as

to the final generation rank one trusses for this population when no user preferences were considered. The reason for this similarity was believed to be that user preferences corresponded well with structurally optimal designs for this population. To better examine the IRR GA's ability to model a user's preferences, a second set of 60by15 preferences were selected for additional trials. These more "radical" user preferences, shown in Figure 5.5, were also used in a longer, 25 main-generation run to see how preferences affected the IRR GA over time.

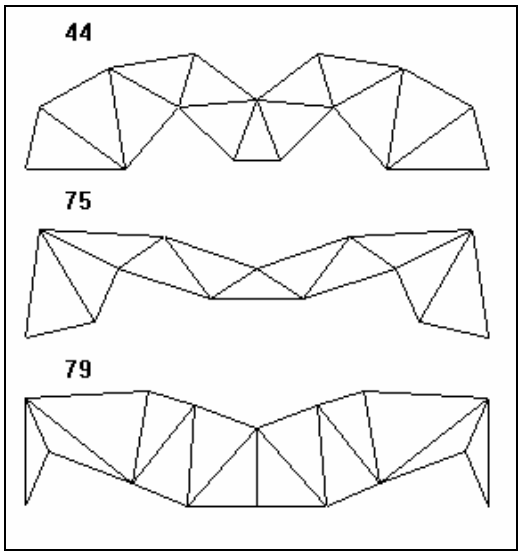


Figure 5.5. Radical user preferences for population 60by15 during preference emphasis trials.

In the last stage of these trials, the final form of preference incorporation was tested using the selections of four volunteers. All volunteers were structural engineering graduate students. The volunteers identified design preferences using KSOM generated groups for population 60by15. Simulations were run using these preferences, and the

10^{th} generation rank-one truss topologies were returned to the volunteers for examination and commentary.

Decision Criteria

In determining a preference incorporation format, several questions had to be addressed. Decision criteria used to select the final mechanism reflected these questions.

Determining whether or not the addition of the preference criteria degraded the overall structural efficiency of the truss optimization was crucial. While modeling user preferences is important to the *conceptual* design of a roof truss system, determining highly performing *structural* systems is equally important. Therefore, records were kept of the average fitness of individuals in the top three ranks of each main generation. The weight and deflection values were also recorded for the individual with the highest fitness. These records, as well as a series of Pareto curves, provide information about the relative structural performance of simulations with and without user preferences. This information helped indicate whether the preference objective was overriding the structural criteria. Pareto curves were generated using a separate program developed by Paik (2005) and displayed truss weight and deflection.

In addition to ensuring that structural objectives were optimized, the algorithm's ability to optimize preference information was considered. For every generation of the main algorithm, records were kept of the average **preferred** value for individuals in the first three ranks. These values were used to show whether the different mechanisms successfully encouraged the selection of preferred trusses. They also provided information to determine which mechanism performed this task most effectively.

Lastly, the larger question of whether or not user preferences were being successfully reflected by the IRR GA had to be addressed. This question was more than determining whether or not preference, weight, or deflection was optimized. It concerned the overall effects of preference integration on the algorithm. Again, comparisons of Pareto fronts illustrate different patterns of exploration when preferences are included in the analysis. The unique topologies of the first rank solutions were identified from points on these graphs and compared to user selections. Similarities were noted and helped address this question as well as determine what mechanism led to the most significant resemblances. Volunteer surveys and commentary provided a final method of examining how accurately preferences were reflected.

Presentation and Discussion of Results

This section presents results for the mechanism and preference emphasis trials of the preference implementation investigation. As previously, graphic results will be displayed for only one of the populations used in this study. Pareto curves and pictures of final generation topologies, etc., will focus on population 60by15, since this population was used in both the mechanism and preference emphasis trials. When available, parallel figures for populations 40by15 and 60by15Par appear in Appendix D.

Mechanism Trials

The mechanism trials explored six different locations and/or formulations for including predictions of user preferences into the IRR GA. These trials were concerned with determining a method that would not unduly influence structural criteria while successfully optimizing and reflecting preference information. In Figures 5.6 through

5.11, the topologically or geometrically unique rank one trusses from the 10th generation of the 60by15 trials are displayed. The trusses in Figure 5.6 were discovered during the Baseline trial and do not reflect user preference information. The remaining trials made use of the selections shown in Figure 5.2. Appendix D presents similar pictures for the 40by15 and 60by15Par populations.

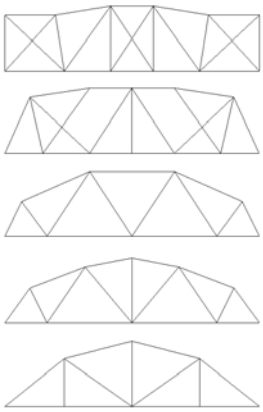


Figure 5.6. 10th generation, rank one trusses for population 60by15 discovered during the Baseline trial.

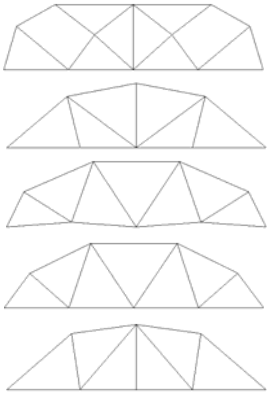


Figure 5.7. 10th generation, rank one trusses for population 60by15 discovered during the Dominate trial.

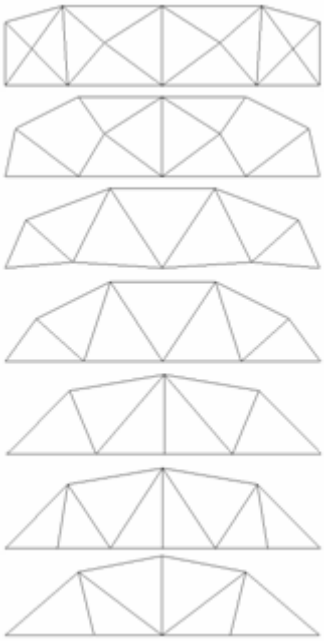


Figure 5.8. 10th generation, rank one trusses for population 60by15 discovered during the CompositeOnly trial.

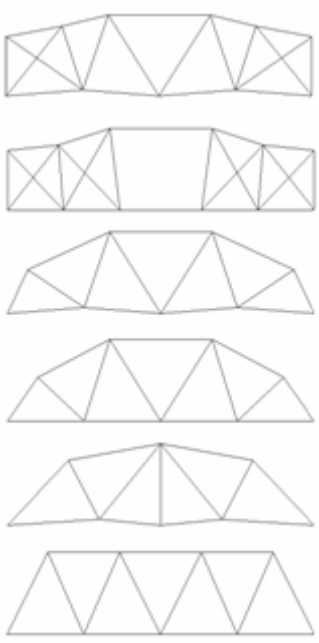


Figure 5.9. 10th generation, rank one trusses for population 60by15 discovered during the Composite10 trial.

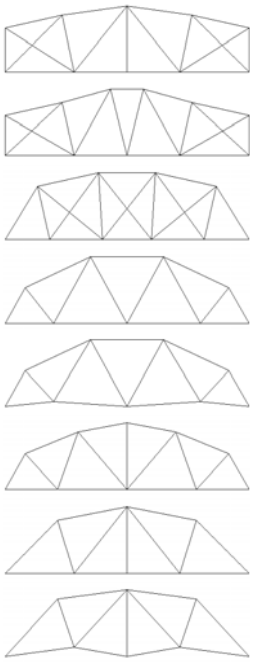


Figure 5.10. 10th generation, rank one trusses for population 60by15 discovered during the Composite50 trial.

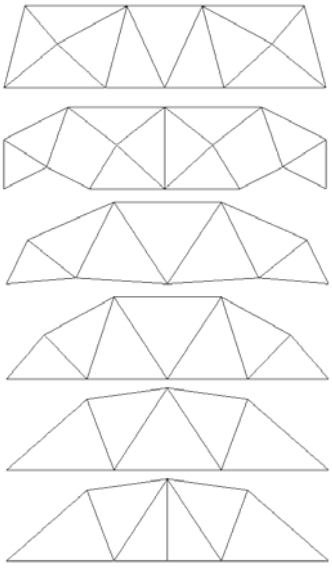


Figure 5.11. 10th generation, rank one trusses for population 60by15 discovered during the Fitness trial.

The truss designs in Figures 5.6 through 5.11 highlight variations in final design alternatives that occur when different preference mechanisms are used. However, the IRR GA is fundamentally a *random* process and is expected to generate different optimal solutions in each analysis. An investigation is required to determine whether or not these variations are meaningful. The Baseline trials for both the 60by15 and 40by15 trials bear a strong resemblance to the final results in Paik (2005). The main difference is that Paik's results, obtained for significantly longer simulations, have fewer truss designs than are obtained with user preferences. This observation reflects the IRR GA's tendency to eliminate designs as optimization progresses. Therefore, while some differences between Figures 5.6 through 5.11, and their Appendix D counterparts, may be attributed to the IRR GA's random processes, they also offer significant variations in terms of the number of trusses proposed and the degree to which these trusses reflect user selections. These variations are representative of the mechanisms themselves. To determine which mechanism best accomplishes the preference implementation goals of this project, these figures were one of the many factors taken into consideration.

Analysis of the mechanisms' performance began by considering whether any or all of these algorithms successfully promoted user preferences during the course of the IRR GA. The **preferred** attribute values for highly ranking solutions were averaged and recorded in each main process generation. Figure 5.12 shows generational variations in the average preference for top ranked individuals in the 40by15 population. Of all populations, 40by15 most clearly illustrates that preference *was* optimized over time. Baseline data is omitted because preferences were not included in this analysis.

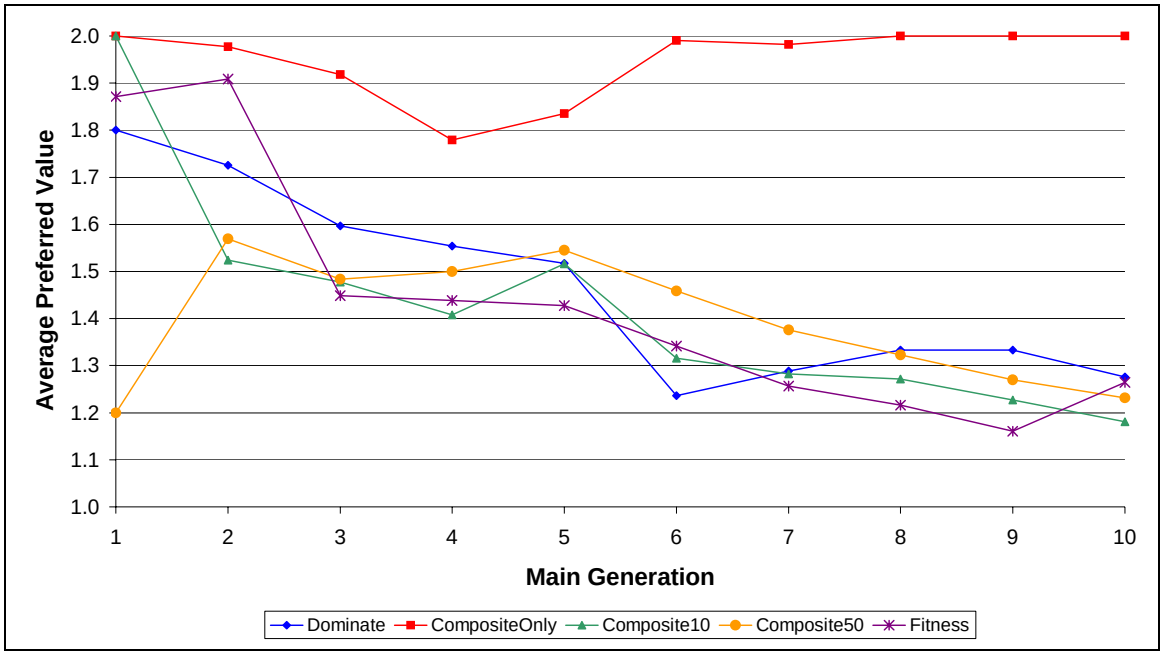


Figure 5.12. Preference optimization for 40by15.

Figure 5.12 shows that all preference methods, except CompositeOnly, reduced the average preference value of top ranked individuals. The CompositeOnly trial performed better for 60by15par, as shown in Figure 5.13. In Figure 5.12, the Composite50 trial began with an initially low average preference, spiked to a larger value, and recovered by the end of the trial. Although performing extremely well for the 40by15 population, the Composite10 and Fitness methods exhibit similar behavior for the 60by15Par population. The Composite50 trial, however, shows a steadily increasing average preference for this second population. Based on Figures 5.12 and 5.13, the Fitness, Composite10, and Dominate formulations perform the preference optimization best and most consistently. Composite10 has a slight advantage over the other methods in that it generates the lowest preference value for population 40by15 and shows slightly

better convergence (though a higher final value) in population 60by15Par.

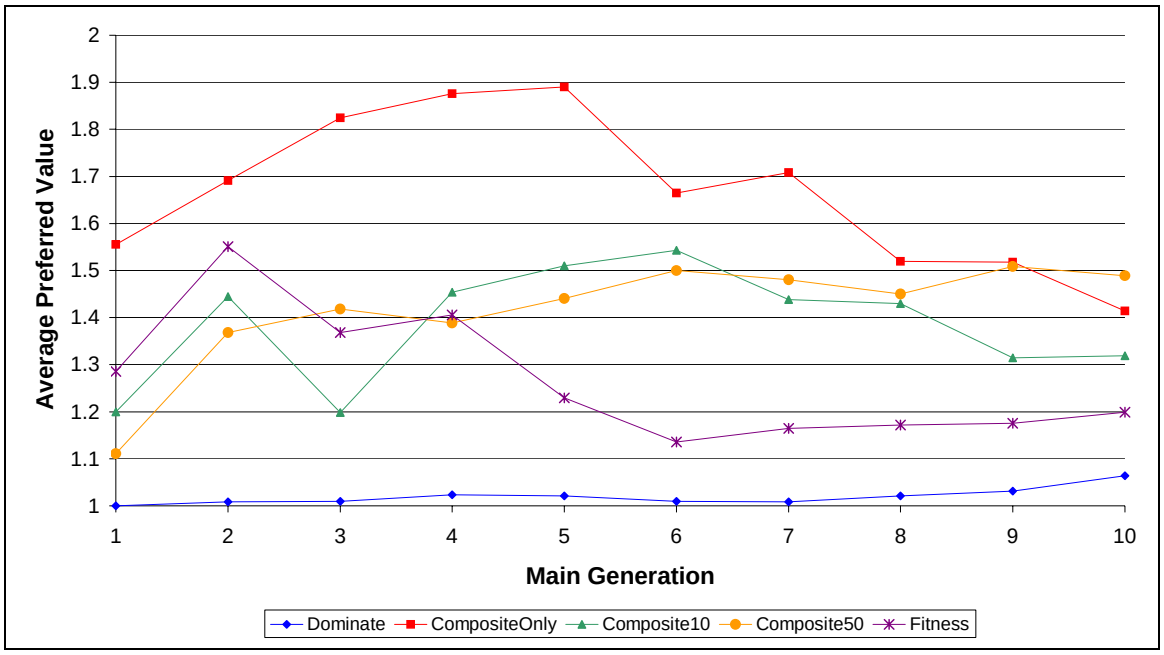


Figure 5.13. Preference optimization for 60by15Par.

Table 5.1 summarizes the average preference values for the final generation of all applicable mechanism trials. Except for the CompositeOnly 40by15 trial, the average preference value for each case was below 1.50, which indicates that topologies reflecting the user's preferences comprised at least half of the final, rank one designs in all but one simulation. Therefore, user preference was successfully optimized. As before, the Dominate, Composite10, and Fitness trials appear to be the best performing formulations. Note that the average preference value for 60by15 is nearly perfect (1.00 or marginally higher) for all trials, an indication that structural and preference criteria were not in conflict for this population. That is, that the user preferences of Figure 5.2

accorded well with the topologies of the Baseline trial, shown in Figure 5.6.

Table 5.1. 10th Generation Average Preference Values for Mechanism Trials.

Trial	60by15	40by15	60by15Par
Dominate	1.00	1.28	1.06
CompositeOnly	1.00	2.00	1.41
Composite10	1.00	1.18	1.32
Composite50	1.00	1.23	1.49
Fitness	1.01	1.26	1.20

The effect of preference incorporation on the structural optimization process was another factor that needed to be considered. By considering a user's aesthetic design preferences, was the IRR GA creating designs that were heavier or less flexible? Did structural performance suffer when aesthetics were considered? To address this concern, the average fitness values for the top ranking solutions were recorded at each generation of the main process. Additional records were kept of the fitness, weight, and deflection data for the rank's most fit individual. Tables 5.2 through 5.4 present the information collected for rank-one individuals in the final generations of each simulation.

Table 5.2. Summary of Structural Behavior for 60by15 Population.

Trial	Avg. Fitness	Max Fitness	Weight (lbs)	Deflection (in)
Baseline	99.43	99.58	7036.9	0.414
Dominate	99.42	99.63	7798.7	0.382
CompositeOnly	99.42	99.62	7719.6	0.376
Composite10	99.46	99.62	7733.9	0.380
Composite50	99.45	99.61	7451.7	0.393
Fitness	99.39	99.59	7809.4	0.393

Table 5.3. Summary of Structural Behavior for 40by15 Population.

Trial	Avg. Fitness	Max Fitness	Weight (lbs)	Deflection (in)
Baseline	99.66	99.78	3106.4	0.171
Dominate	99.24	99.77	3319.4	0.158
CompositeOnly	99.66	99.78	3439.4	0.149
Composite10	99.19	99.74	3219.4	0.168
Composite50	99.30	99.76	2786.4	0.190
Fitness	99.30	99.77	2706.4	0.195

Table 5.4. Summary of Structural Behavior for 60by15Par Population.

Trial	Avg. Fitness	Max Fitness	Weight (lbs)	Deflection (in)
Baseline	99.46	99.62	7835.0	0.375
Dominate	99.49	99.61	8060.1	0.372
CompositeOnly	99.41	99.62	6866.9	0.415
Composite10	99.42	99.62	7138.3	0.396
Composite50	99.42	99.62	8325.6	0.342
Fitness	99.41	99.61	6897.0	0.474

Recall that fitness values account for the weight and deflection of individual trusses as well as the degree to which individuals violate kinematic stability, constructability requirements, and allowable member stresses. Tables 5.2 through 5.4 give an overall view of how the inclusion of user preferences changes the optimization process from a structural standpoint.

The fittest member in any given generation may lie at a different point along the Pareto curve than the fittest member of a previous or succeeding generation. Therefore, the different weight-deflection ratios represented in these tables do not indicate either better or poorer optimization when preferences are incorporated. What is shown is that, by and large, the fitnesses obtained in the final generation of the GA are in the same

region for all trials, regardless of whether user preferences are included. Moreover, the average fitness values do not change substantially when preferences are incorporated. This leads to the general impression that the inclusion of user preferences does not substantially degrade structural performance.

The Pareto front displayed in Figure 5.14 reinforces this impression. This curve shows the relative weight and deflection values for the rank one individuals in the 10^{th} generation of the 60by15 Composite10 and Baseline trials. Results for the Baseline simulation are in red, and the yellow points represent Composite10 solutions. As can be seen, the Composite10 results obscure most of the Baseline values, and results for most user preference mechanisms are equally hard to distinguish. This fact indicates that the simulations proposed final generation designs that were nearly identical from a structural viewpoint. Results for populations 40by15 and 60by15Par exhibit similar behavior and are shown in Appendix D. Overlapping points on these Pareto curves correspond to proposed truss designs with the same weight and maximum deflection; however, these points do not necessarily share identical geometries, member sizes, or even topologies.

Examining Figure 5.14, the Composite10 trial slightly outperforms the Baseline trial in some locations, especially the crucial middle region where the weight and deflection objectives are most balanced. The Baseline simulation, however, has a small advantage in the low weight, high deflection region. Figure 5.14 also shows that the Composite10 simulation created a *fuller* Pareto front than the Baseline. The weight-deflection values in yellow are more evenly spread than those in red. Comparing the Pareto fronts of the five simulations, which incorporated user preferences, to the

Baseline simulation, which did not, revealed that the preference addition greatly increased the IRR GA's explorative abilities.

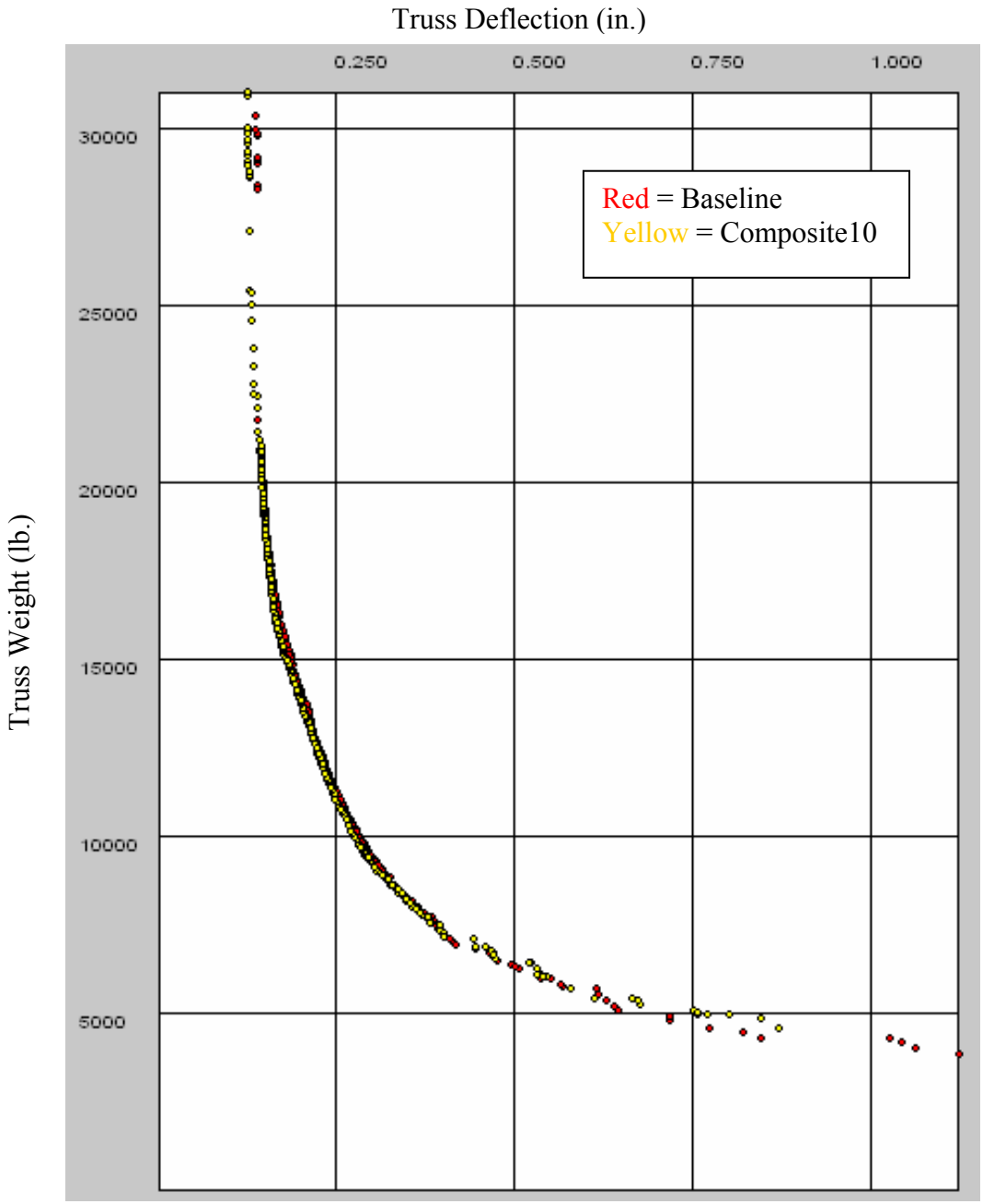


Figure 5.14. 60by15 Pareto fronts for the Baseline (red) and Composite10 (yellow) trials.

Figure 5.15 on the next page presents such a comparison. Top-ranking designs for the 10th generation of the 60by15 main processes are plotted in the Pareto fronts of this figure. The curves for each mechanism trial are laid side-by-side to create a more complete picture of the trials' behavior as well as avoid difficulties distinguishing between the analyses. Figure 5.15, as well as Figures D.15 and D.16 of Appendix D, illustrates that the user preference mechanism trials explored different regions of the Pareto curve. Units for these figures are in inches and pounds.

In each population, the Pareto fronts of the Baseline trials contain significant gaps. These gaps generally occur in the high weight/low deflection and low weight/high deflection regions. In creating a conceptual design tool, however, it is important to create as complete a picture of the Pareto front as possible. Incorporating user preferences has successfully fleshed out the curves, thus giving a more complete picture of optimal truss alternatives. This completion does not always correspond to the generation of a new truss topology, as can be seen from Figures 5.6 through 5.11. Rather, including user preferences appears to have encouraged more geometric and shape optimization of preferred topologies in the top truss ranks.

Once again, the Dominate, Composite10, and Fitness trials show strong performances by offering the most complete curves in all three populations. Of the three mechanisms, the Composite10 curve appears to encompass more designs than the other trials for populations 40by15 and 60by15Par. There is a slight truncation of low weight/high deflection trusses occurring for 60by15.

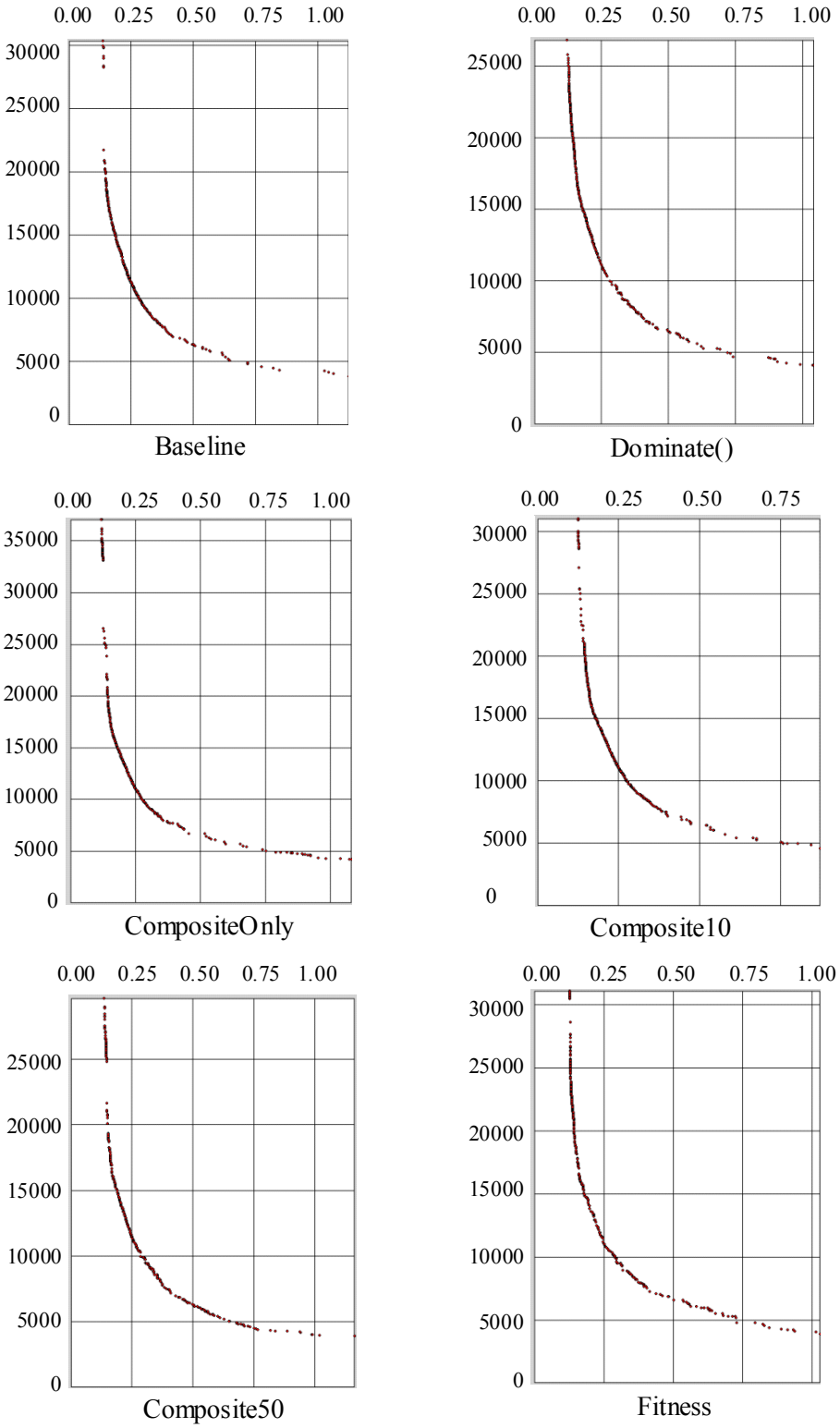


Figure 5.15. 60by15 10th generation, rank one Pareto fronts for all mechanism trials.

Another method for visualizing the effect of preference implementation on the IRR GA was to directly compare the truss designs in the Pareto fronts of the trials' final generations to those selected by the user. Trusses with unique topologies, or recurring topologies with significant geometric differences, were evaluated according to perceived similarity to the user's selections of Figures 5.2 through 5.4. In some cases, designs predicted by the mechanisms were identical to user preferences. In others, topologies differed by the placement of one or two members or the amount of top/bottom chord inclination, etc. These similarities were recorded and summarized in Tables 5.5 through 5.7. The similar preferences column indicates the percentage of final generation trusses that reflected, without replicating, user preferences. Information is also presented about the number of unique truss designs in the final Pareto curve, so that IRR GA diversity may be judged. Data for the Baseline trials are included to illustrate how preferences affected the selection and generation of topologies.

Table 5.5. Truss Types in Final Pareto Fronts for Population 60by15.

Trial	Topologies	Identical Preferences (%)	Similar Preferences (%)
Baseline	5	40.0	60.0
Dominate	5	20.0	80.0
CompositeOnly	7	14.3	85.7
Composite10	6	33.3	66.7
Composite50	8	37.5	62.5
Fitness	6	33.3	66.7

Table 5.6. Truss Types in Final Pareto Fronts for Population 40by15.

Trial	Topologies	Identical Preferences (%)	Similar Preferences (%)
Baseline	5	0.0	60.0
Dominate	6	0.0	50.0
CompositeOnly	4	0.0	75.0
Composite10	6	0.0	50.0
Composite50	6	0.0	50.0
Fitness	3	0.0	33.3

Table 5.7. Truss Types in Final Pareto Fronts for Population 60by15Par.

Trial	Topologies	Identical Preferences (%)	Similar Preferences (%)
Baseline	4	0.0	75.0
Dominate	3	0.0	100.0
CompositeOnly	6	0.0	66.7
Composite10	5	0.0	80.0
Composite50	4	0.0	80.0
Fitness	6	0.0	100.0

These tables highlight some interesting behaviors. First, 100% of the final truss designs in the 60by15 population were either identical to or reflections of user preferences, regardless of how, or even *if*, these preferences were incorporated into the optimization process. As suggested earlier, the user preferences were closely aligned to the final outcome of the Baseline simulation; that is, the user tended to prefer truss designs that would prove to be structurally optimal. For this reason, the 60by15 population was selected for additional preference emphasis trials using the "radical" user selections of Figure 5.5.

Largely because trusses in population 60by15 tended to be homogenous, populations 60by15Par and 40by15 were included in this study in the hopes of seeing the

effects of user preferences on more diverse initial populations. Although none of the user selections were replicated in the final Pareto front for either of these populations, incorporating these preferences tended to encourage exploration. More truss forms were generated, and, for population 60by15Par, these new trusses tended to align well with user preferences.

In population 40by15, the user preferences were aimed at exploring the height portion of the search space. Preferences were tall and angular (see Figure 5.3) in the hopes of creating some unusual truss designs for this population. However, generated designs fell far short of this goal. The similarities represented in Table 5.6 reflect some resemblance between trusses 1, 3, and 12 and the three topologies shown in Figure 5.16. These similarities were the weakest of any used in this analysis.

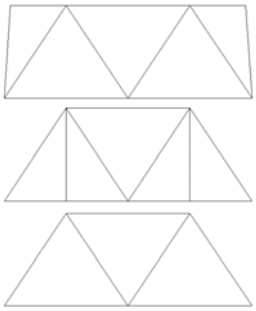


Figure 5.16. Topologies considered similar to 40by15 preferences.

Performance for these methods considered not only how many identical or similar preferences were discovered but also how many unique topologies or geometric forms were proposed. The Composite50 analysis generated the largest number of truss

types for population 60by15 and tied with Composite10 and Dominate for this distinction in the 40by15 population. Similarity percentages were also high for Composite50 for all three populations, but very few trusses were proposed for 60by15Par. The Dominate mechanism also proposed disappointingly few truss designs for populations 60by15 and 40by15, although similarity percentages were relatively high. Fitness performed well for both 60-foot spans but suggested only three topologies for the 40-foot span, only one of which proved to be similar to user selections.

The Composite10 formulation, on the other hand, offered fairly balanced results for all three populations. Not only did the Composite10 trials propose a significant number of truss designs, but these designs also had relatively high preference similarities. Since Composite10 also showed advantages both in optimizing preferences and in expanding the Pareto front, this mechanism was selected for additional study during the preference emphasis trials.

Preference Emphasis Trials

The preference emphasis trials sought to further explore the question of whether the IRR GA successfully reflected user preferences. Figure 5.15 and Tables 5.5 through 5.7 began to address this issue by showing that the inclusion of user preferences led to more exploration and that many final-generation trusses did reflect at least a portion of the user's preferences. However, since the original user preferences for population 60by15 corresponded so closely to the structurally optimal designs in the Baseline trial, it seemed advisable to reconsider this population to determine if unusual, non-optimal selections would have a more dramatic effect.

The "radical" user preferences of Figure 5.5 all have an arched top chord that dips at midspan. Additionally, the supports tend to be sharp and the bottom chords bowed. Using the Composite10 formulation, these preferences were used to create the 10th generation Pareto front illustrated in Figure 5.17.

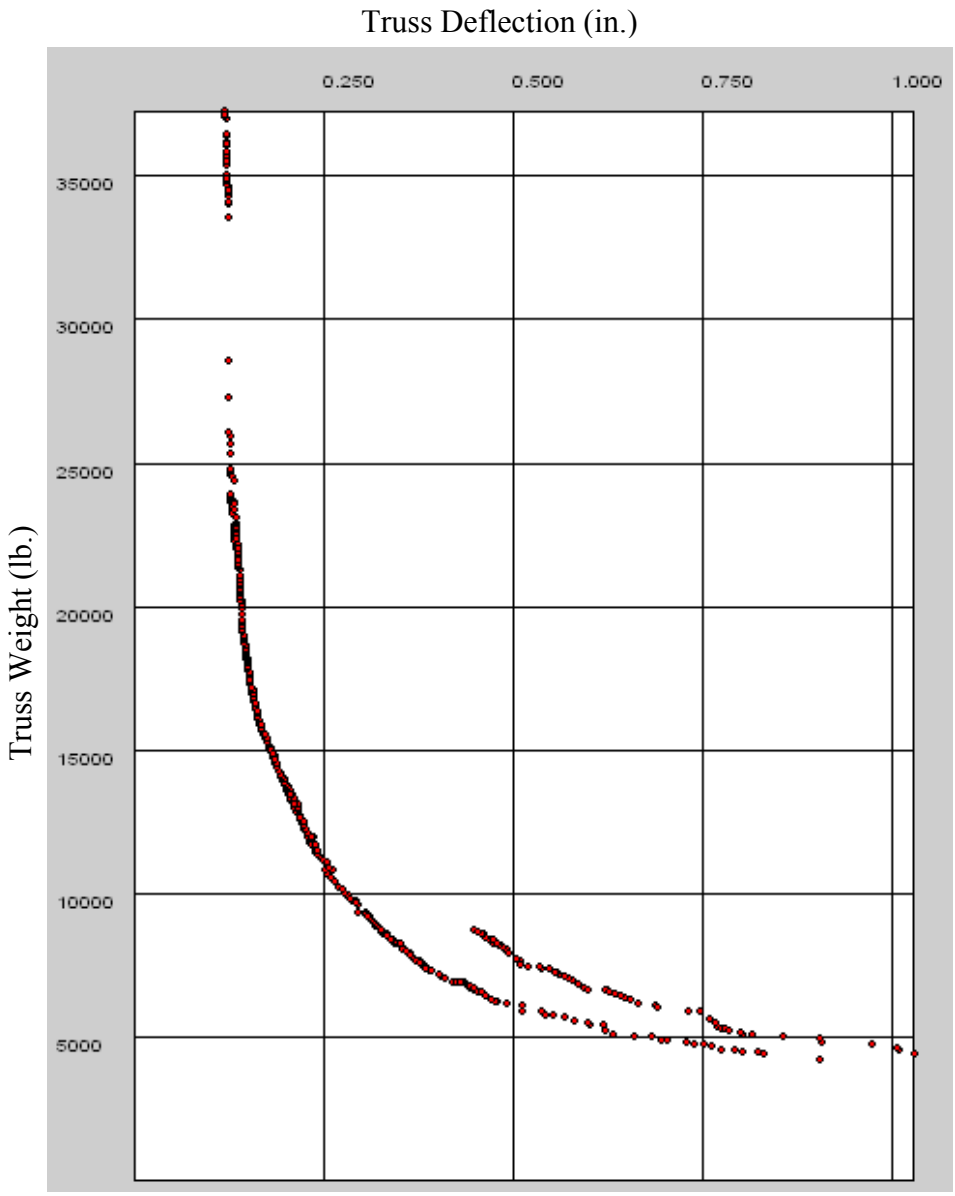


Figure 5.17. 10th generation Pareto front(s) for 60by15 with radical user selections.

In addition to the expected Pareto front that matches the structural performance of the Baseline and other mechanism trials, Figure 5.17 reveals the presence of a second "front." Typically, non-dominated solutions would lie along only one Pareto curve. In this instance, however, three major objectives (weight, deflection, and user preferences) are being plotted in only two dimensions (weight and deflection). This fact accounts for why a secondary front may develop. Figure 5.18 presents the geometrically or topologically unique truss designs that lie along both curves. The topologies in the left column of Figure 5.18 were located along the left-most Pareto front and are common among the previous 60by15 trials. The right column designs appeared in the second Pareto front and represent geometric variations new to this analysis.

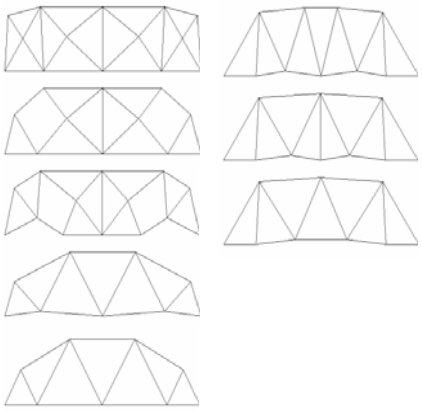


Figure 5.18. 10th generation designs for 60by15 with radical user selections.

The right column of Figure 5.18 shows truss designs that correspond more closely to the radical user selections than previously seen. The top chords are slightly arched and supports are pointed. This trend indicates that the IRR GA is attempting to

accommodate user preferences even though they are not structurally optimal designs. Overall, Figures 5.17 and 5.18 indicate that user preferences encourage discovery of new topologies while maintaining topologies that perform well from a purely structural standpoint. A longer, 25 main-generation simulation was run to determine how this behavior altered over time. Results are presented in Figures 5.19 and 5.20.

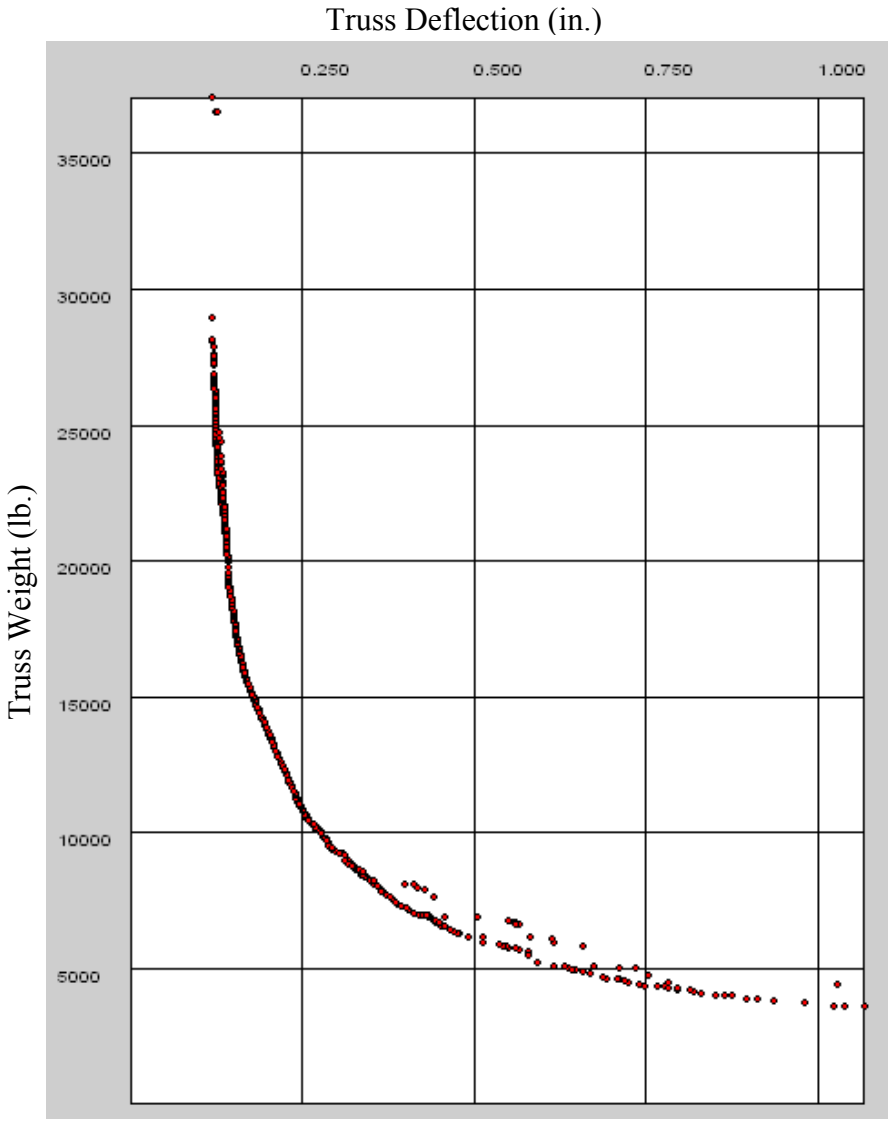


Figure 5.19. 25th generation Pareto front(s) for 60by15 with radical user selections.

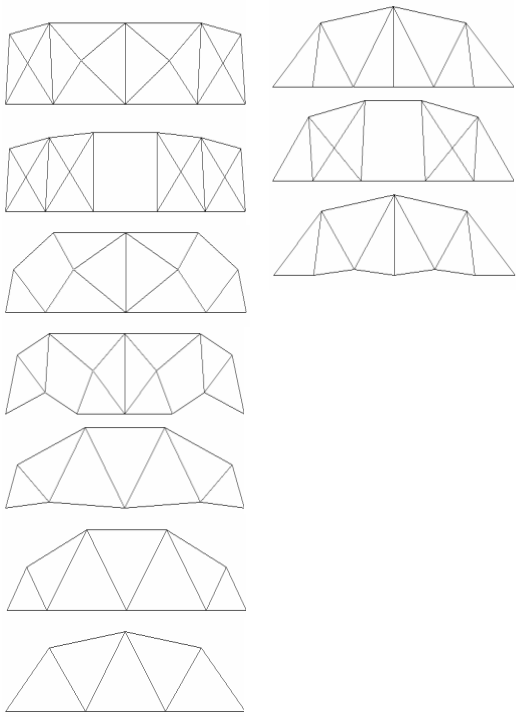


Figure 5.20. 25th generation designs for 60by15 with radical user selections.

In Figure 5.19, it is apparent that the two Pareto fronts are merging as the IRR GA progresses in its search. The capacity of the GA to preserve structurally rigorous topologies while continuing to explore new topologies of interest to the user appears to be degrading with time. Figure 5.20, which illustrates the rank-one truss forms present in the 25th generations of this analysis, reinforces this observation. Although more topologies are present in the 25th generation, it is obvious that the new topologies do not coincide with the user preferences illustrated in Figure 5.5. Only three of Figure 5.20 topologies have the bowed bottom chord and sharply pointed supports characteristic of these preferences; instead, more flat bottom chords are present.

This observation led to a re-examination of the manner in which user preferences

are emphasized as a criterion in the IRR GA. In the original **compositePen** definition (see Equation 5.8), the preference variable is normalized by a maximum value of three, the value to which preferences are initialized. However, except for the brief time before user preferences are applied to the analysis, the maximum preference value will be two. Altering the normalization variable appears to positively affect the ability of the IRR GA to incorporate user preferences, as shown in Figure 5.21 on the next page.

Figure 5.21 superimposes Pareto fronts obtained for the 60by15 trial when the Composite10 formulation normalized preference values by three (yellow) and by two (red). These curves are for rank-one individuals the 10^{th} main generation. The normalization-by-two trial, shown in red, has a much stronger secondary curve than the normalization-by-three trial, shown in yellow. This stronger front corresponds to a larger number of truss designs in the normalization-by-two trial, as shown in Figure 5.22.

From a structural standpoint, the normalization-by-two trial proposes fewer solutions in the high weight/low deflection region, with the majority of designs in the Pareto curve weighing less than 25,000 pounds. This lack may actually enhance the structural performance, however, as the Pareto front in this region tends to be asymptotic. That is, increasingly heavier designs are proposed with marginal gains in stiffness. In all other respects, the curves overlap to such a degree that the structural performance of solutions in the normalization-by-two trial appears equal to that of the normalization-by-three simulation when the first, "traditional" Pareto front is considered.

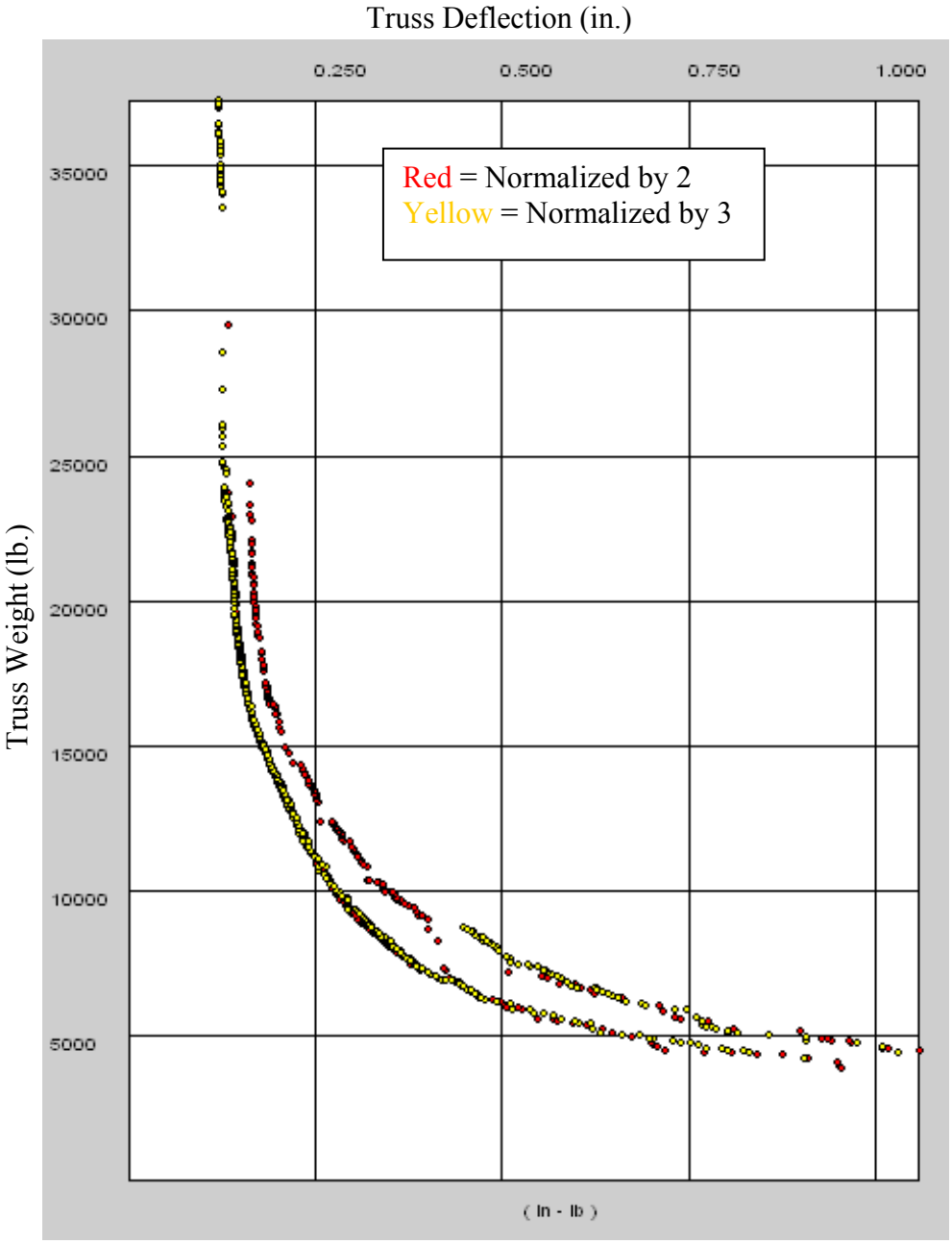


Figure 5.21. Pareto fronts for 60by15 population with radical selections using the normalization-by-two (red) and normalization-by-three (yellow) trials.

Figure 5.22 shows the truss designs present in the 10th generation of the 60by15 population when the normalization-by-two Composite10 formulation is used. The

Pareto front for this trial is also shown to give a more complete view than Figure 5.21. Figure 5.23 presents similar information for the 25th generation of this analysis. As before, designs in the left column correspond to designs along the left Pareto curve.

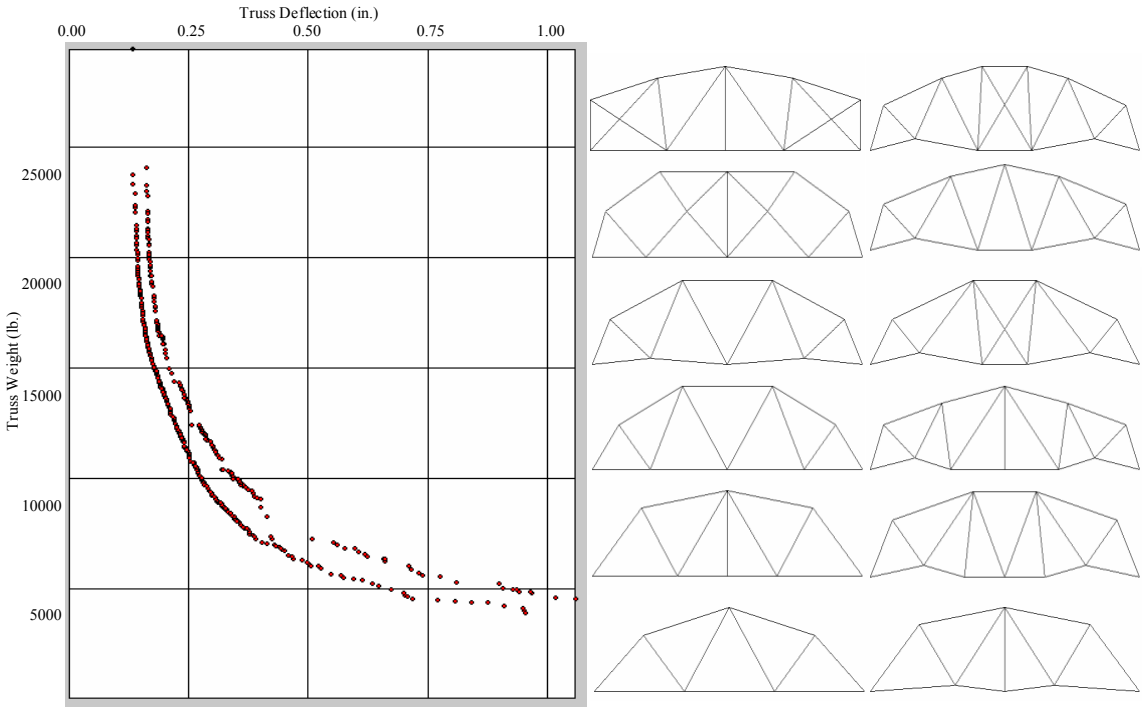


Figure 5.22. 10th generation Pareto front and corresponding truss designs for 60by15 population with radical user selections using normalization-by-two.

The normalization-by-two trial proposed significantly more truss designs than the normalization-by-three trial by the end of the 10th generation. All of the designs in the right-hand column of Figure 5.22 have bowed bottom chords with sharp points. Although none of these designs show dipped top chords, there is more top chord angularity than in previous designs. These characteristics show that the user preferences

of Figure 5.5 are being more clearly reflected for this analysis.

By the end of the 25th generation, fewer designs are present for the normalization-by-two trial than for the normalization-by-three trial (see Figure 5.23). However, these eight designs contain five with bowed bottom chords and tipped supports. The normalization-by-three trial had only three designs, of ten proposed, with these characteristics by the 25th generation. The secondary Pareto front of Figure 5.23 is also significantly more pronounced than in Figure 5.19, and higher weight trusses are beginning to be more fully explored.

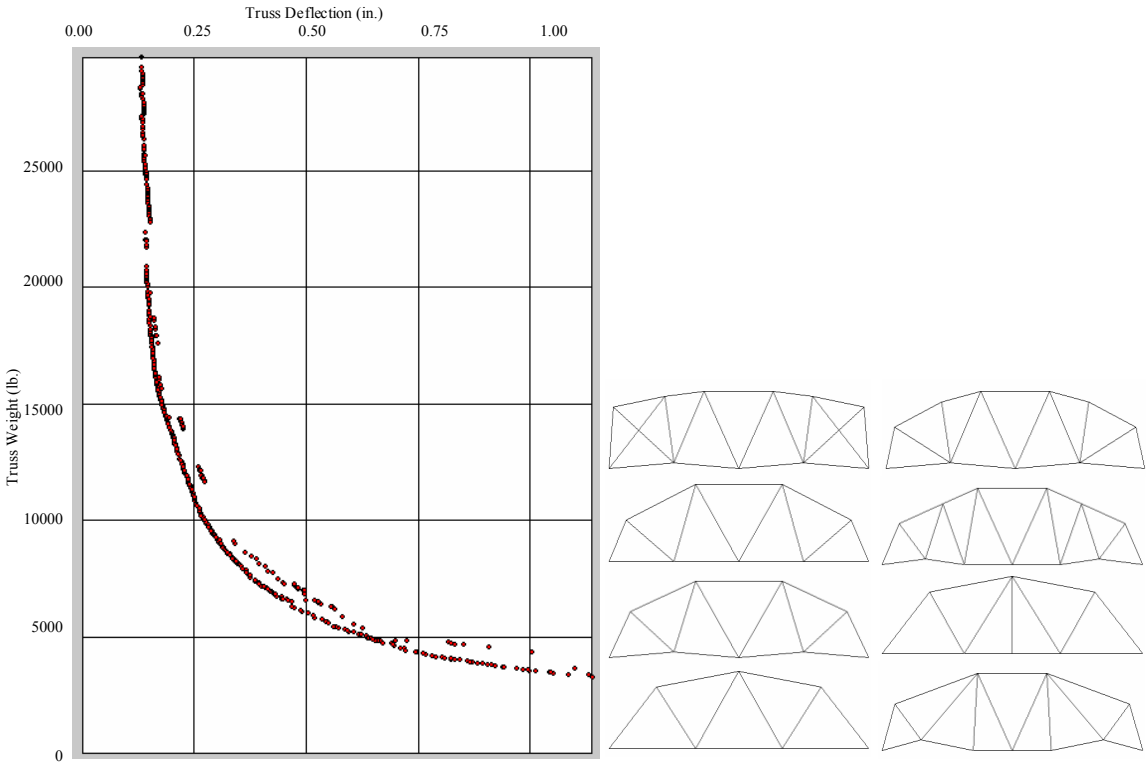


Figure 5.23. 25th generation Pareto front and corresponding truss designs for 60by15 population with radical user selections using normalization-by-two.

Overall, this figure shows that the IRR GA is converging to structurally optimal designs by bringing the two "fronts" closer together as the simulation progresses. At the same time, exploration has led to a fuller Pareto curve and to the discovery of several new and interesting designs that reflect user preferences. The IRR GA is successfully balancing the architectural and structural requirements placed upon it.

Additional analyses were conducted using the original user selections for populations 60by15, 40by15, and 60by15Par. These simulations were used to verify that the normalization-by-two approach to the **compositePen** definition improved the identification of user preferences. Table 5.8 summarizes similarities between 10^{th} generation truss designs and the user preferences of Figures 5.2 through 5.4. Figures of the Pareto curves and the truss designs themselves appear in Appendix D.

Table 5.8. Final Truss Forms for All Populations When Normalized by Two.

Population	Total Types	Identical Preferences (%)	Similar Preferences (%)
60by15	5	40.0	60.0
60by15Par	7	14.3	85.7
40by15	8	0.0	75.0

Table 5.8 illustrates that the normalization-by-two trials encouraged exploration for the 60by15Par and 40by15 populations; the number of truss forms proposed for these populations was greater than in any of the previous mechanism trials. Both populations also showed significant improvement in discovering trusses similar to user preferences, and the 60by15Par simulation proposed a truss design *identical* to user preferences for the first time. Performance for 60by15 is similar to that of the Baseline trial and

proposes one less design than the Composite10 trial when normalized by three.

Results for the 40by15 population were especially encouraging. Earlier, the user's preference for angular, almost cathedral-looking top chords was completely ignored. When the **preferred** attribute is normalized by two, however, this population forms a "second" Pareto front as in the radical user selection analyses for 60by15. As shown in Figure 5.24, the designs along both this second curve and the dominant front have begun to mirror the user's preferences. Top chords are sharper for some designs, and many of the angles comprising the trusses are narrower. The Pareto curve also shows that highly performing structural alternatives are still present despite this additional exploration.

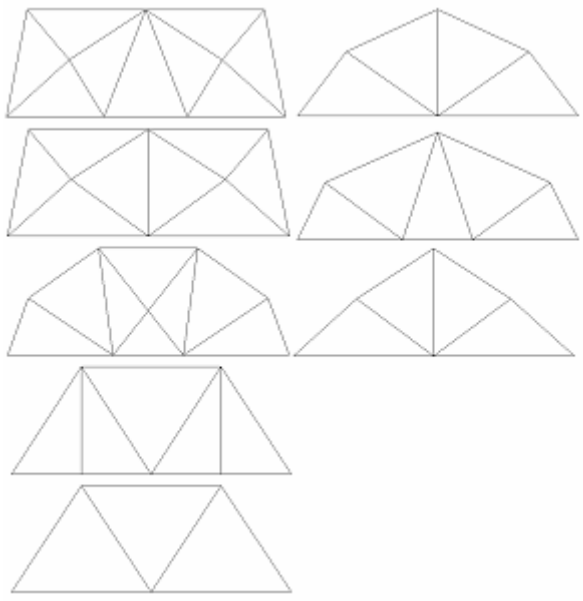


Figure 5.24. 10^{th} generation design alternatives for the 40by15 population when preferences normalized by two.

The behavior of the Composite10 formulation, when **preferred** is normalized by two, appears to offer the most effective modeling of user preferences while simultaneously ensuring that the structural criteria are being met. At this point, the ability of the IRR GA to respond to user preferences has received significant consideration. However, a final, more subjective measure of the efficacy of preference implementation was desired. To that end, four structural engineering graduate students were asked to participate in this investigation.

The 60by15 population was selected for these final trials. As before, the 1D KSOM generated truss design groups, and volunteers were asked to identify their personal preferences from among these groups. The IRR GA incorporated these selections using the Composite10, normalization-by-two formulation. The main process consisted of ten generations. The topologically or geometrically unique truss designs along the 10^{th} generation, rank-one Pareto front were returned to the volunteers at the end of each simulation.

Volunteers offered feedback about their satisfaction with these designs. They were asked the following questions:

- To what degree did you feel the trusses suggested to meet your design preferences reflected the set of design features or concepts you used to make your initial truss suggestions?
- Regardless of your answer to the above question, how satisfied were you with the trusses suggested to meet your design preferences? Were the suggested trusses aesthetically interesting to you?

Users selected answers to these questions from among responses expressing different degrees of satisfaction.

Three of the four volunteers felt the proposed designs reflected some of their preferences, as opposed to reflecting all, most, or none of their preferences. All three reported that they were mostly satisfied (as opposed to very, somewhat, slightly, or not satisfied or disappointed) with the proposed designs from a purely aesthetic viewpoint. One of these three volunteers, who will be referred to as Volunteer 1, selected two groups, each of which exhibited a different characteristic that was desirable. Volunteer 1's selections are shown in Figure 5.25. Trusses 12, 14, 21, 71, and 74 were particularly appealing because of the steep slopes at mid-span.

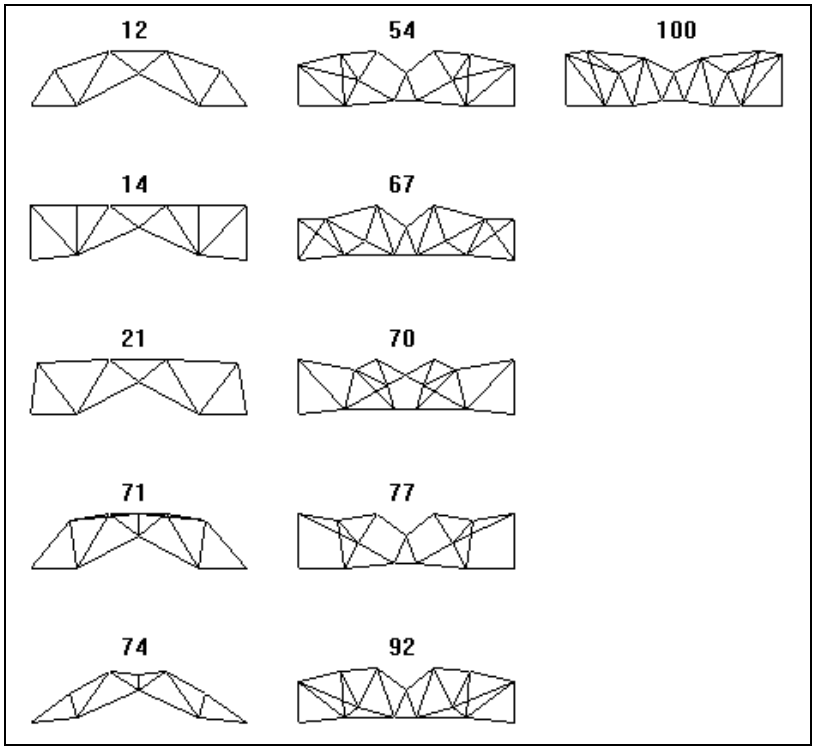


Figure 5.25. 60by15 topologies selected by Volunteer 1.

As shown in Figure 5.26, none of the truss designs proposed by the IRR GA reflected Volunteer 1's preference for steep bottom chords at mid-span. However, of the four volunteers whose preferences were recorded, this user's simulation was the only one where a second Pareto front forms (see Figure 5.26). Results for Volunteer 1's selections are shown in red; Baseline results are superimposed in yellow.

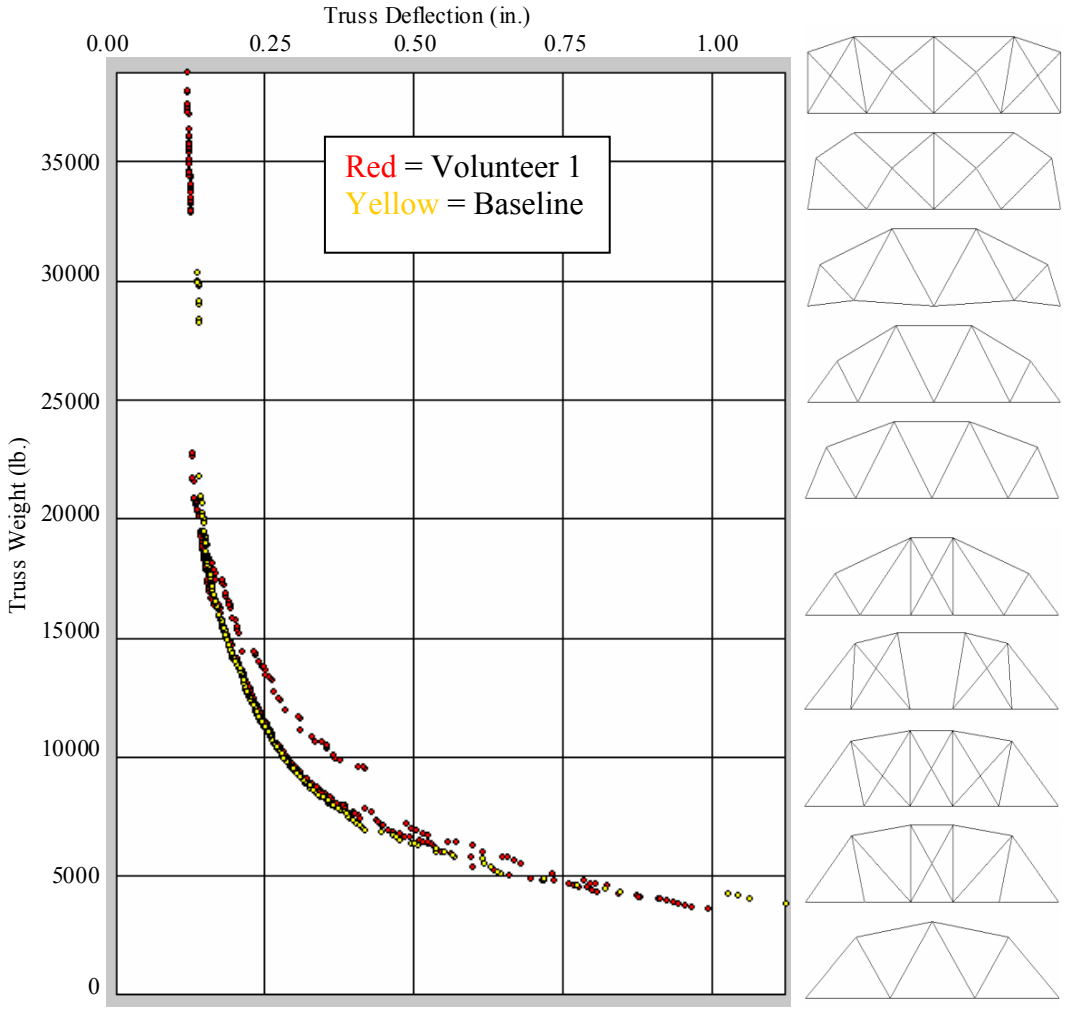


Figure 5.26. 60by15 truss designs and Pareto front for Volunteer 1 (red); Pareto front without user preferences (yellow).

As usual, the simulation incorporating user preferences shows significantly more exploration than the Baseline trial. Truss designs that lie along both the original and the secondary Pareto curves created for Volunteer 1 are separated by a gap in Figure 5.26. The top five designs of Figure 5.26 represent the topologies of the dominant Pareto curve and have been seen frequently in previous analyses. However, the bottom five trusses, which lie along the secondary front, contain many unusual designs. Four of these designs appear similar to trusses 54, 67, 70, 77, 92, and 100. The fact that ten truss designs were proposed at the end of ten generations itself indicates that the IRR GA is continuing to explore the search space for designs that more closely align to Volunteer 1's preferences.

The remaining three volunteers made fairly conventional selections from among the 60by15 topologies, and their results do not differ significantly from those explored elsewhere in this work. Selections, Pareto fronts, and design alternatives for Volunteers 3 and 4 are shown in Appendix D. Volunteer 2's results will be discussed as representative of these three analyses.

Volunteer 2 deliberately made truss selections that reflected simple, frequently occurring truss topologies. These selections are presented in Figure 5.27. With the exception of trusses 12, 14, 21, 71, and 74, these selections tend to reflect those shown in Figure 5.2. These truss topologies are also similar to design alternatives previously proposed by the IRR GA.

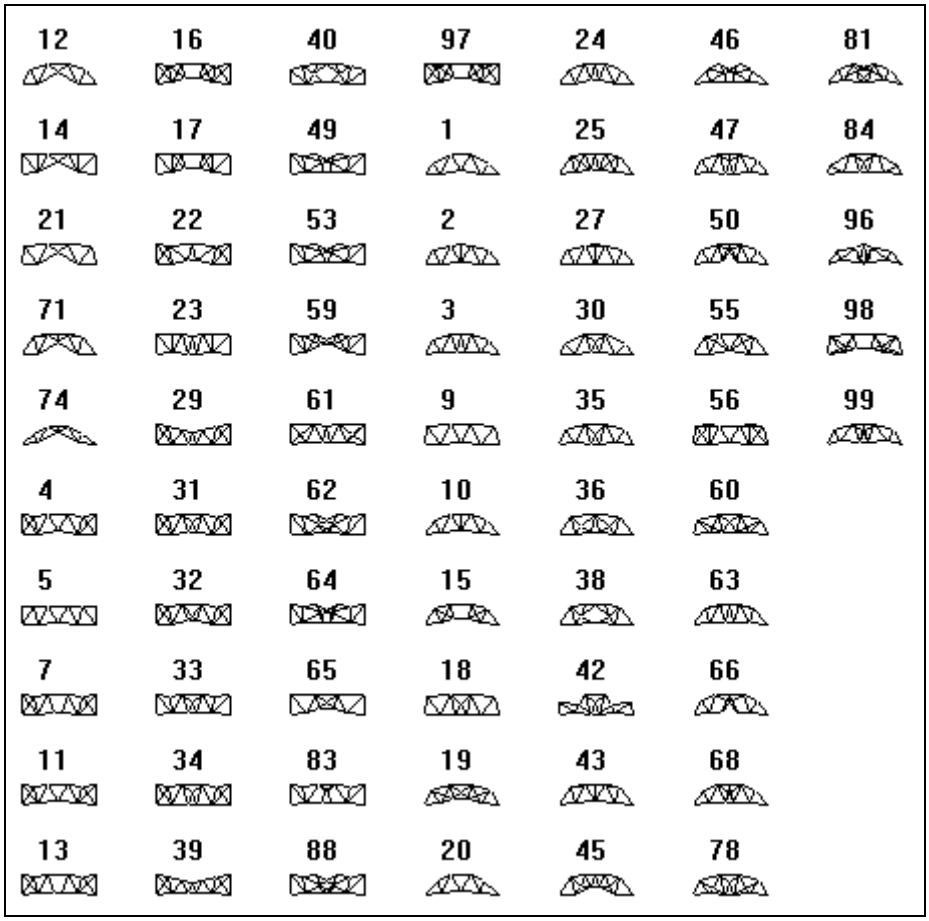


Figure 5.27. 60by15 topologies selected by Volunteer 2.

Truss topologies proposed in response to Volunteer 2's selections are presented in Figure 5.28. Volunteer 2 reported that the IRR GA's design alternatives exactly reflected the input preferences and was very satisfied with the aesthetic appearance of these trusses as a whole. Although many of the designs have occurred elsewhere in this work, the fact that eight designs remain in the Pareto front at the 10th generation indicates that exploration has been encouraged in this analysis, even though the user's preferences were easily satisfied. In Figure 5.28, the Pareto front created for Volunteer 2

is shown in red, while the Baseline analysis is shown in yellow. It is virtually impossible to distinguish between these two analyses, except for the fact that the red curve is more complete. Therefore, when user preferences closely align to structural optimums, preference implementation tends close gaps in the Pareto front.

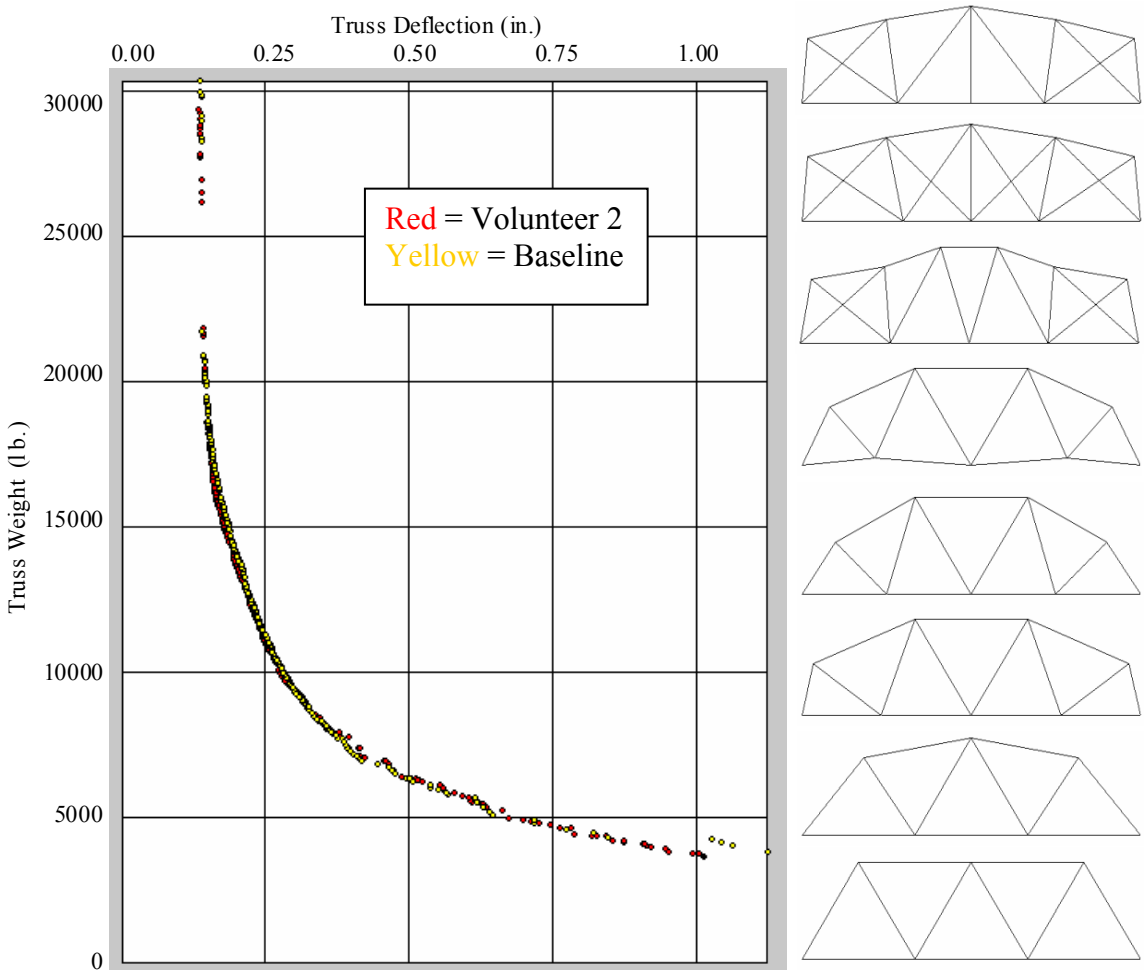


Figure 5.28. 60by15 truss designs and Pareto front for Volunteer 2 (red); Pareto front without user preferences (yellow).

Selected Preference Implementation Method

Five proposals for incorporating user preference into the IRR GA program developed in Paik (2005) were considered in this chapter. Each of these mechanisms made use of the preference prediction algorithm developed in Chapters II through IV to identify designs that coincided with a user's initial selections. This prediction was stored in the **preferred** attribute of each population member.

The **dominate( )** function and the **compositePen**, **fitness**, and **rankFitness** variables were considered as locations where user preference could affect the IRR GA outcome. These locations placed the **preferred** attribute into either tournament selection or ranking decisions. A series of mechanism trials conducted on the 60by15, 40by15, and 60by15Par populations identified the Composite10 formulation as the optimal method for including user preferences.

The Composite10 formulation used the value of the **preferred** attribute in the **dominate( )** function, which performs a comparison between two designs to determine if either may be said to dominate the other. This information is used in both ranking and selection procedures. The **preferred** attribute was also used in the calculation of a total penalty term, **compositePen**, described by Equation 5.8, with an integer multiplier of ten. The **compositePen** variable was used during the selection process when the **dominate( )** function failed to provide a clear decision between two designs.

The Composite10 formulation was shown to successfully optimize the average preference of individuals in the IRR GA populations. Comparisons to Pareto fronts proved that structural criteria were equally satisfied with and without user preferences,

which answered concerns that structural performance might be impaired by the inclusion of user input. In fact, rather than hampering the IRR GA, it was determined that the inclusion of the user preference criterion encouraged exploration. Pareto fronts for these analyses contained fewer gaps and more design alternatives.

This last result was surprising and significant. The objective formulation proposed by Paik (2005) was already highly constrained, and the maintenance of sufficient genetic diversity in the populations was a major problem in this study (Paik 2005). Adding an additional objective was expected to further reduce the feasibility region for solutions and therefore reduce the overall number of unique truss topologies discovered by the GA. It was thought that some of the constraints placed upon member length or stability might need to be relaxed in order to accommodate this addition. The preference criterion did reduce the feasibility region, and a smaller number of sub-processes were run for simulations that included this objective. However, although the sub-processes identified fewer groups of "similar" truss designs, more exploration within these identified types appears to have occurred.

With the ability of the Composite10 mechanism to optimize both preference and structural criteria settled, the preference emphasis trials explored how closely the final generation design alternatives matched user preferences. This exploration, based upon radical selections for population 60by15, concluded that the **preferred** attribute should be normalized by two (versus three) in the **compositePen** definition. This alteration improved the number and quality of design alternatives.

The resulting Composite10 formulation allowed the IRR GA to accommodate

even unusual user preferences by creating a second Pareto front (when plotting weight and deflection) in the population's top rank. Designs along this secondary front represented attempts to explore topologies more in keeping with the user's selections. It was shown that this second curve would be absorbed into the dominant front over time. The final front would contain designs that balanced both the user's selections and strong structural performance. Therefore, this formulation was successful in altering the overall architectural concept for a truss while meeting the necessary structural requirements.

Four structural engineering graduate students evaluated the final mechanism. Results for these students reinforced the conclusion that the IRR GA would seek a balance between preference and structural criteria. Although none of the students' inputs drove the IRR GA to create new designs based entirely on their selections, the students were mostly satisfied with the aesthetic appeal of the proposed design alternatives.

CHAPTER VI

CONCLUSIONS AND FUTURE WORK

This research effort focused on the development of computational methods to recognize a human designer's aesthetic preferences and embed these preferences into the multi-objective optimization of large-span roof trusses. The optimization program developed in Paik (2005) focused on optimizing the weight and deflection of trusses while constraining member lengths, kinematic stability, and allowable stresses. However, the truss designs obtained did not always accord with engineering judgment (Paik 2005). The current work extends this previous research to create a more complete conceptual design tool that provides structurally optimal truss systems and allows a human user to influence the final form of these systems.

Summary of Objectives

The first step towards incorporating user preferences into the conceptual design system was to identify quantifiable characteristics that uniquely described the aesthetic features of a truss design. This task was discussed in Chapter II, where a series of twenty potential features were evaluated based upon their ability to visually partition truss populations of varying size and complexity. A characteristic feature vector was created using the nine most effective truss descriptors. The feature vector described basic information about the simplicity, general shape, and chord shape of a potential truss design and included the following characteristics:

- Number of joints

- Number of members
- Average joint connectivity
- Maximum truss height
- Truss depth
- Mid-span clearance
- Top chord flatness
- Bottom chord flatness
- Number of direction changes along the top chord

In Chapter III, an unsupervised clustering algorithm was sought to classify a population of potential truss designs based upon the above characteristics. Earlier research indicated that partitioning a population into groups of similar individuals reduced the burden on users asked to visually assess designs (Ohsaki and Takagi 1998; Takagi and Ohya 1996). Both classic and heuristic clustering mechanisms were considered for the present application. The one-dimensional Kohonen's self-organizing map (KSOM) best matched judgments made by human users. Numerically, the KSOM formed tight groups, as described by the sum-of-squares error and a comparison of standard deviations, with relatively few single-topology clusters. Therefore, the one-dimensional KSOM provided the clustering ability for the user input process.

Using the KSOM, truss designs are placed into groups according to perceived similarities in their characteristic feature vectors. A designer is presented with the best truss design (i.e., the truss closest to the cluster center) for each group and asked to select design preferences from among these trusses. These selections provide feedback about a

user's likes and dislikes in the present population. Chapter IV considered different mechanisms to convert this input into predictions of a user's likes and dislikes in *future* populations.

Preference algorithms were used to reproduce a series of user inputs as well as extrapolate these inputs to new populations. Information contained in the characteristic feature vectors of both selected and unselected designs were used in these decisions. After an initial investigation failed to provide sufficiently accurate results, a hybrid back-propagation with rough set reduct (BP-RSR) algorithm was developed. The BP-RSR algorithm relies on rough set reduct to strategically reduce user selections that vary significantly from the population as a whole and from corresponding group members. This reduction supplements the KSOM clustering ability by identifying and eliminating unwanted designs placed alongside desirable ones. Once inputs are analyzed for consistency, a back propagation neural network is trained to recognize the user's preferences. Overall, the BP-RSR outperformed all other preference algorithms considered in this study and proved to be an effective method for predicting user preferences.

The final stage of this investigation connected the preference detection method to the implicit redundant representation genetic algorithm (IRR GA) designed previously. Chapter V describes the integration process. Potential formulations focused on adding the user preference criterion to the ranking and selection procedures. The Composite10 mechanism was selected as the most effective way to promote user preferences. This method used preference to determine solution dominance and break ties during

tournament selection.

Incorporating user preferences into the IRR GA significantly increased exploration. The IRR GA is extremely useful in undefined search spaces because of the flexibility with which it encodes design variables. However, this flexibility proved to be a deficit in a multi-criteria environment with a small feasible region (Paik 2005). Preserving unique truss topologies and encouraging the discovery of more was a priority in Paik's (2005) work and led to the creation of separate sub-processes to analyze groups of similar design. These multiple processes corresponded to high computation costs and a slow run time.

When the feasibility region was further narrowed by addition of the preference criterion, fewer sub-processes were required, thus reducing the computation effort to complete the simulations. Contrary to expectations, narrowing the feasible region had a beneficial effect on exploration. The sub-processes focused on designs interesting to the user and provided locations for new truss forms to develop. Two key results indicate this improved explorative ability. First, in most cases, a greater number of geometrically or topologically unique trusses were proposed in the final generation of the IRR GA when user preferences were included. Second, the Pareto fronts on which these trusses lie were more fully developed.

Overall, preference implementation enabled the IRR GA to identify designs more likely to be acceptable to the user while simultaneously improving the algorithm's ability to discover additional design alternatives. Fitness information and Pareto fronts from analyses with and without user preferences reveal that this improved exploration did not

detract from the optimization of the structural criteria. In instances where user preferences aligned with structural optima, the IRR GA still received the benefit of added exploration. When user preferences promoted non-optimal designs, the IRR GA strove to incorporate these preferences while retaining the most structurally efficient individuals.

The results presented in this work show that preference incorporation into the automated design of structural components is not only possible but an area with a high potential for improving the conceptual design process.

Future Research Directions

The preference implementation study created an effective method for promoting a designer's aesthetic preferences during truss optimization. In order to be truly efficient as a conceptual design tool, however, the IRR GA must propose still more design alternatives. The algorithm must also become more responsive to the designer.

One way to increase responsiveness is to re-examine the characteristic feature vector that forms the basis for evaluating user inputs. The feature vector was developed for trusses automatically generated according to the algorithm described in (Agarwal and Raich 2005). This algorithm constructed trusses from triangular building blocks. The grammar used in the current conceptual design system, however, is nodal in nature (Paik 2005). While most features will not be affected by this difference, the trusses look different from those used to identify the descriptors.

For instance, the algorithm does not necessarily create triangular panels. That is, vertical and horizontal members could be used to connect four nodes without using

diagonal members to create the triangular shapes characteristic of trusses. As it currently stands, the characteristic feature vector cannot determine when these open panels occur. User surveys from the classification analysis, however, indicate that users respond to the number of diagonal members in a panel, and it is unlikely that many users would willingly select designs with possible instabilities under lateral loads. Perhaps an additional feature could determine how many diagonal members were in each rectangular truss panel and use this information to better accommodate user preference. In addition, a horizontal load can be imposed to identify unstable trusses during the design process.

Another aspect of the algorithm that might be reconsidered is the current method of creating sub-processes of similar truss designs. The IRR GA with preference information tends to find fewer groups, and therefore fewer sub-processes are run. Since this reduction led to beneficial results in terms of exploration, it may be possible to eliminate the use of sub-processes altogether. The purpose of the sub-process architecture was to help prevent premature convergence of the IRR GA to a single topology (Paik 2005). However, the inclusion of user preference might be strong enough to entirely replace the sub-processes.

If the use of sub-processes is still required, then the method for creating truss groups for these sub-processes should be reconsidered. Currently, the number of nodes, number of members, and total member lengths are being used to determine whether or not trusses are similar to each other. However, total member length was considered during feature identification as a potential indicator of truss simplicity. This

characteristic was rejected as containing no significant information about truss appearance. Sub-process grouping could easily be replaced with a KSOM using the characteristic feature vector described in this study. A full replacement may not be necessary, however, and further investigation into this matter is needed. In any case, the total length criterion should be removed and replaced with one or more elements of the feature vector.

On a larger scale, the effect of human-GA interaction on the conceptual design of trusses has not been fully explored. For instance, preference predictions were based on user selections made at the end of the first GA generation. In addition, users evaluated only 100 truss designs. Further research is needed to consider both the timing and extent of user interactions during run time. Issues to consider include whether users should be asked to evaluate more than 100 trusses; whether user evaluation would have a greater effect on the algorithm if it occurred later in the optimization process; and whether users should be queried for input more than once during the course of the algorithm.

REFERENCES

Adelson, P. (1998). "Smart, unexpected, and beautiful: intelligent aesthetics + smart engineering." *Intelligent Systems Through Artificial Neural Networks*, 8, 881-886.

Agarwal, P. and Raich, A. M. (2005). "Optimal design of bridge and roof trusses using multi-objective genetic algorithms." *Proc., 2005 ASCE International Conference on Computing in Civil Engineering*, Cancun, Mexico.

Anderberg, M. R. (1973). *Cluster Analysis for Applications*, Academic Press, New York.

Becker, S. (1996). "Unsupervised neural network learning procedures for feature extraction and classification." *Applied Intelligence*, 6(3), 185-203.

Camp, C., Pezeshk, S., Cao, G. (1998). "Optimized design of two-dimensional structures using a genetic algorithm." *Journal of Structural Engineering*, 124(5), 551-559.

Cheng, F. Y. and Li, D. (1997). "Multiobjective optimization design with Pareto genetic algorithm." *Journal of Structural Engineering*, 123(9), 1252-1261.

Chikata, Y., Yasuda, N., Matsushima, M., and Kobori, T. (1998) “Inverse analysis of aesthetic evaluation of planned concrete structures by neural networks.” *Computer-Aided Civil and Infrastructure Engineering*, 13, 255-264.

Coello Coello, C. A. (2001). "A short tutorial on evolutionary, multiobjective optimization." *Evolutionary Multi-Criteria Optimization: First International Conference: EMO 2001*, Eckart Zitzler et al., eds, Springer, Berlin.

Cottrell, M., de Bodt, E., Verleysen, M. (2001). "A Statistical Tool to Assess the Reliability of Self-Organizing Maps." *Advances in Self-Organising Maps*, Nigel

Allinson, Hujun Yil, Lesly Allinson, and Jon Slack, eds, Springer, London.

Deb, K. (2001). *Multi-Objective Optimization Using Evolutionary Algorithms*, John Wiley and Sons, Chichester, England.

Duda, R. O. and Hart, P. E. (1973). *Pattern Classification and Scene Analysis*, John Wiley and Sons, New York.

Everitt, B. (1980). *Cluster Analysis*, 2nd ed, Halsted Press, John Wiley and Sons, New York.

Everitt, B. S., Landau, S., and Leese, M. (2001). *Cluster Analysis*, 4th ed, Arnold, London.

Furuta, H., Maeda, K., and Watanabe, E. (1995). "Application of genetic algorithm to aesthetic design of bridge structures." *Microcomputers in Civil Engineering*, 10(6), 415-421.

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Boston.

Goodacre, R., Kaderbhai, N., McGovern, A. C., and Goodacre, E. A. (1999). "Chemometric analyses with self organizing feature maps: a worked example of the analysis of cosmetics using Raman spectrosocopy." *Kohonen Maps*, Amsterdam.

Graf, J. and Banzhaf, W. (1995). "Interactive evolution of images." *Proc., 4th Annual Conference on Evolutionary Programming*, San Diego, California.

Grierson, D. E., and Pak, W. H. (1993). “Optimal sizing, geometrical, and topological design using a genetic algorithm.” *Structural Optimization*, 6, 151-159.

Hajela, P. and Lee, E. (1995). "Genetic algorithms in truss topological optimization."

International Journal of Solids and Structures, 32(22), 3341-3357.

Hassoun, M. H. (1995). *Fundamentals of Artificial Neural Networks*, MIT Press, Cambridge, Massachusetts.

Hayashida, N., and Takagi, H. (2000). "Visualized IEC: Interactive evolutionary computation with multidimensional data visualization." *Proc., IECON 2000: IEEE International Conference on Industrial Electronics, Control, and Instrumentation*, Nagoya, Aichi, Japan.

Haykin, S. (1999). *Neural Networks: A Comprehensive Foundation*, 2nd Ed., Prentice Hall, Upper Saddle River, New Jersey.

Hibbeler, R. C. (2006). *Structural Analysis*, 6th ed., Pearson Prentice Hall, Upper Saddle River, New Jersey.

Keyes, L., Winstanley, A., and Healey, P. (2003). "Comparing learning strategies for topographic object classification." *Proc., International Geoscience and Remote Sensing Symposium*, Toulouse, France.

Kohonen, T. (1982). "Clustering, taxonomy, and topological maps of patterns." *Proc., Sixth International Conference on Pattern Recognition*, Munich, Germany.

Kohonen, T. (1988a). "The 'neural' phonetic typewriter." *IEEE Computer*, 21(3), 11-22.

Kohonen, T. (1988b). *Self-Organization and Associative Memory*, 2nd ed., Springer-Verlag, Berlin.

Kohonen, T. (2001). *Self-Organizing Maps*, 3rd ed., Springer-Verlag, Berlin.

Kulasuriya, C., Dias, W. P. S., and Hettiarachchi, M. T. P. (2002). "The aesthetics of proportion in structural form." *The Structural Engineer*, 80, 22-27.

Ohsaki, M. and Takagi, H. (1998). “Improvement of presenting interface by predicting the evaluation order to reduce the burden of human interactive EC operators.” *Proc., IEEE International Conference on Systems, Man, and Cybernetics,* San Diego, California.

Paik, S. (2005). "Multi-objective optimal design of steel trusses in unstructured design domains." M.S. thesis, Texas A&M University, College Station, Texas.

Parmee, I. C. (1993). “The concrete arch dam: an evolutionary model of the design process.” *Proc., International Conference on Artificial Neural Nets and Genetic Algorithms*, Innsbruck, Austria.

Pawlak, Z. (1981). "Information systems theoretical foundations." *Information Systems*, 6(3), 205-218.

Pawlak, Z. (1991). *Rough Sets: Theoretical Aspects of Reasoning about Data*, Kluwer Academic Publishers, Dordrecht, Netherlands.

Pawlak, Z., Grzymala-Busse, J., Slowinski, R., Ziarko, W. "Rough sets." (1995). *Communications of the ACM*, 38(11), 88-95.

Petiot, J.-F. and Grognet, S. (2002). "A multidimensional scaling approach for product design and preference modeling." *Proc., 2002 IEEE International Conference on Systems, Man and Cybernetics*, Yasmine Hammamet, Tunisia.

Pothos, E. M. and Chater, N. (2002). "A simplicity principle in unsupervised human categorization." *Cognitive Science*, 26, 303-343.

Rafiq, M. Y., Mathews, J. D., and Bullock, G. N. (2003). "Conceptual building design--evolutionary approach." *Journal of Computing in Civil Engineering*, 17(3), 150-158.

Raich, A. M. and Ghaboussi, J. (1997). "Implicit representation in genetic algorithms using redundancy." *Evolutionary Computation*, 5(3), 277-302.

Raich, A. M. and Ghaboussi, J. (2000). "Evolving structural design solutions using an implicit redundant genetic algorithm." *Structural and Multidisciplinary Optimization*, 20(3), 222-231.

Rajan, S. D. (1995). "Sizing, shape, and topology design optimization of trusses using genetic algorithm." *Journal of Structural Engineering*, 121(10), 1480-1487.

Rajeev, S. and Krishnamoorthy, C. S. (1997). "Genetic algorithms-based methodologies for design optimization of trusses." *Journal of Structural Engineering*, 123(3), 350-358.

Ramasamy, J. V. and Rajasekaran, S. (1996). "Artificial neural network and genetic algorithm for the design optimization of industrial roofs-- a comparison." *Computers and Structures*, 58(4), 747-755.

Reeke, G. N., Jr., Finkel, L. H., and Edelman, G. M. (1990). "Selective recognition automata." *An Introduction to Neural and Electronic Networks*. Steven F. Zornetzer, Joel L. Davis, and Clifford Lau, eds., Academic Press Inc., San Diego, CA, 203-226.

Reekie, R. F. (1972). *Design in the Built Environment*. Crane, Russak, and Company, Incorporated, New York.

Ritter, H., Thomas M., and Klaus S. (1992). *Neural Computation and Self-Organizing Maps*, Addison-Wesley Publishing Company, Reading, Massachusetts.

Ruy, W.-S., Young-Soon, Y., Kim, G.-H., and Yeun, Y.-S. (2001). "Topology design of truss structures in a multicriteria environment." *Computer-Aided Civil and*

Infrastructure Engineering, 16, 246-258.

Schyns, P. G. (1991). "A modular neural network model of concept acquisition." *Cognitive Science*, 15, 461-508.

Sims, K. (1993). "Interactive evolution of equations for procedural models." *Visual Computer*, 9 (8), 466-476.

Sisk, G. M., Miles, J. C., and Moore, C. J. (2003). "Designer centered development of GA-based DSS for conceptual design of buildings." *Journal of Computing in Civil Engineering*, 17(3), 159-166.

Smyth, S. N. and D. R. Wallace. (2000). "Toward the synthesis of aesthetic product form." *Proc., 12th International Conference on Design Theory and Methodology*, Baltimore, Maryland.

Spath, H. (1980). *Cluster Analysis Algorithms for Data Reduction and Classification of Objects*, Ellis Horwood Limited, Chichester, England.

Takagi, H. (1998). "Interactive evolutionary computing – Cooperation of computational intelligence and human KANSEI." *Proc., 5th International Conference on Soft Computing and Information/Intelligent Systems*, Fukuoka, Japan.

Takagi, H. and Ohya, K. (1996). "Discrete fitness values for improving the human interface in an interactive GA." *Proc., IEEE Conference on Evolutionary Computation*, Nagoya, Japan.

Taner, M. T., Berge, T., Walls, J. D., Smith, M., Taylor, G., Dumas, D., Carr, M. B. (2001). "Well log calibration of Kohonen-classified seismic attributes using Bayesian logic." *Journal of Petroleum Geology*, 24(4), 405-416.

von der Malsburg, C. (1990). "Network self-organization." *An Introduction to Neural and Electronic Networks*. Steven F. Zornetzer, Joel L. Davis, and Clifford Lau, eds., Academic Press Inc., San Diego, California, 421-432.

Wagner, R. (2001). "Aesthetic design improvement." *Proc., 13th International Conference on Engineering Design*, Glasgow, Scotland.

Wemmert, C., Gancarski, P., and Korczak, J. (1999). "An unsupervised collaborative learning method to refine classification hierarchies." *Proc., 11th International Conference on Tools with Artificial Intelligence*, Chicago, Illinois.

West, H. H. and Geschwindner, L. F. (2002). *Fundamentals of Structural Analysis*, 2nd ed., John Wiley and Sons, Incorporated, New York.

Willshaw, D. J., and von der Malsburg, C. (1976). "How patterned neural connections can be set up by self-organization." *Proceedings of the Royal Society London*, B194, 431-445.

Wong, W. Y. and Sneed, M. (2004). "Capturing user preferences in structural design." Undergraduate research project summary, Texas A&M University, College Station, Texas.

Yanagisawa, H. and Fukada, S. (2003). "Interactive reduct evolutional computation for aesthetic design." *Proc., ASME Design Engineering Technical Conference*, Chicago, Illinois.

Yanagisawa, H. and Fukada, S. (2004). "Interactive reduct evolutional computation for aesthetic design." *Proc., ASME Design Engineering Technical Conference*, Salt Lake City, Utah.

Yanagisawa, H. and Fukada, S. (2005). "Interactive reduct evolutional computation for aesthetic design." *Journal of Computing and Information Science in Engineering*, 5(1), 1-7.

Yeomans, D. T. (1992). *The Trussed Roof: Its History and Development*, Scolar Press, Aldershot, England.

APPENDIX A

This Appendix contains additional 1D KSOM feature maps used in the selection of truss descriptors for inclusion in a characteristic feature vector. For each characteristic considered in this investigation, five feature maps were created. The maps for population S50M50 were shown in Chapter II of the text. The remaining population maps are shown here.

Maximum Truss Height

Figures A.1 through A.4 show feature maps when maximum height was used as input for the 1D KSOM. The maps shown are for populations S25M50, S100M50, S100M25, and S100M100, respectively.

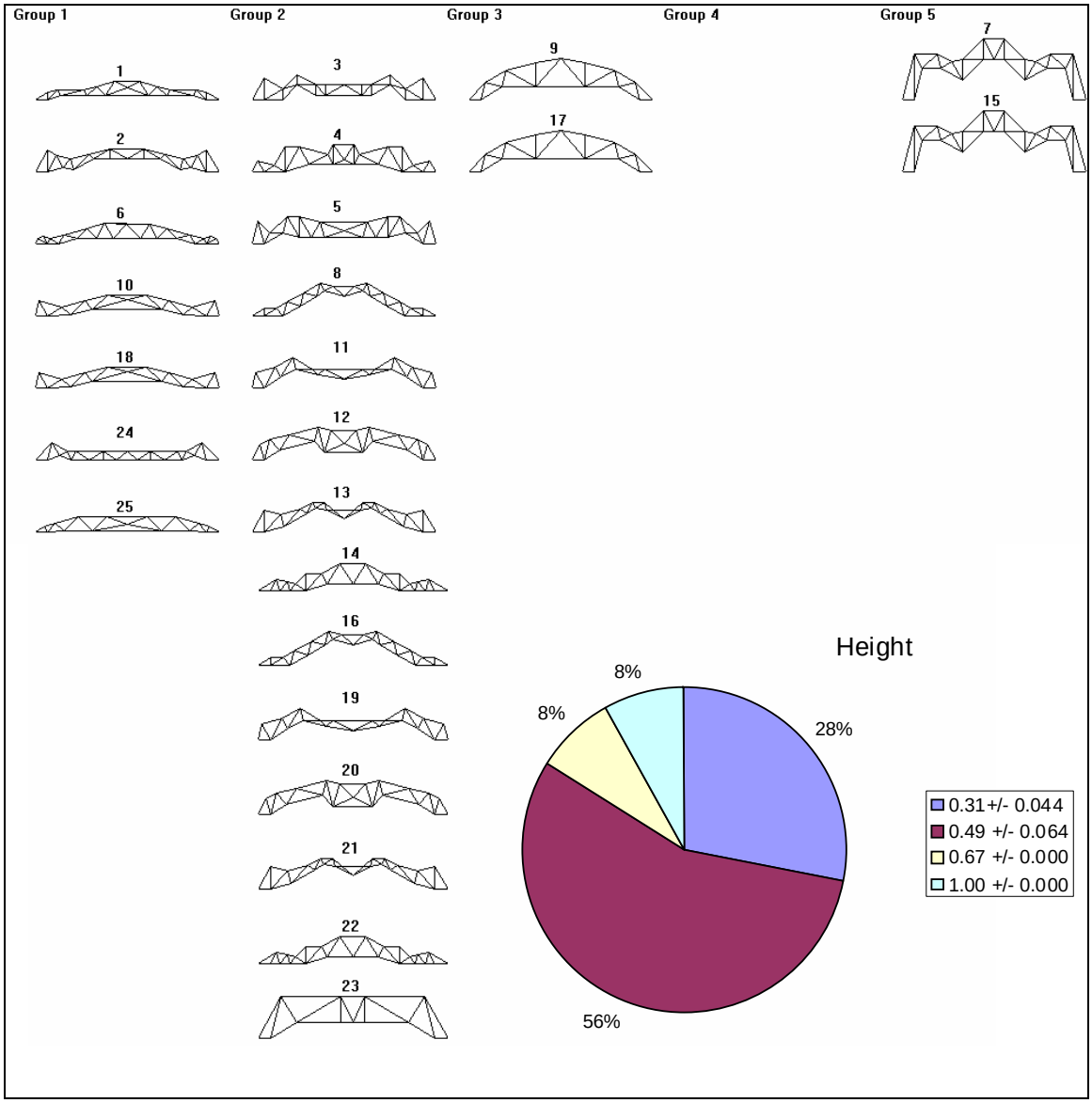


Figure A.1. Feature map created for S25M50 with maximum height as input.

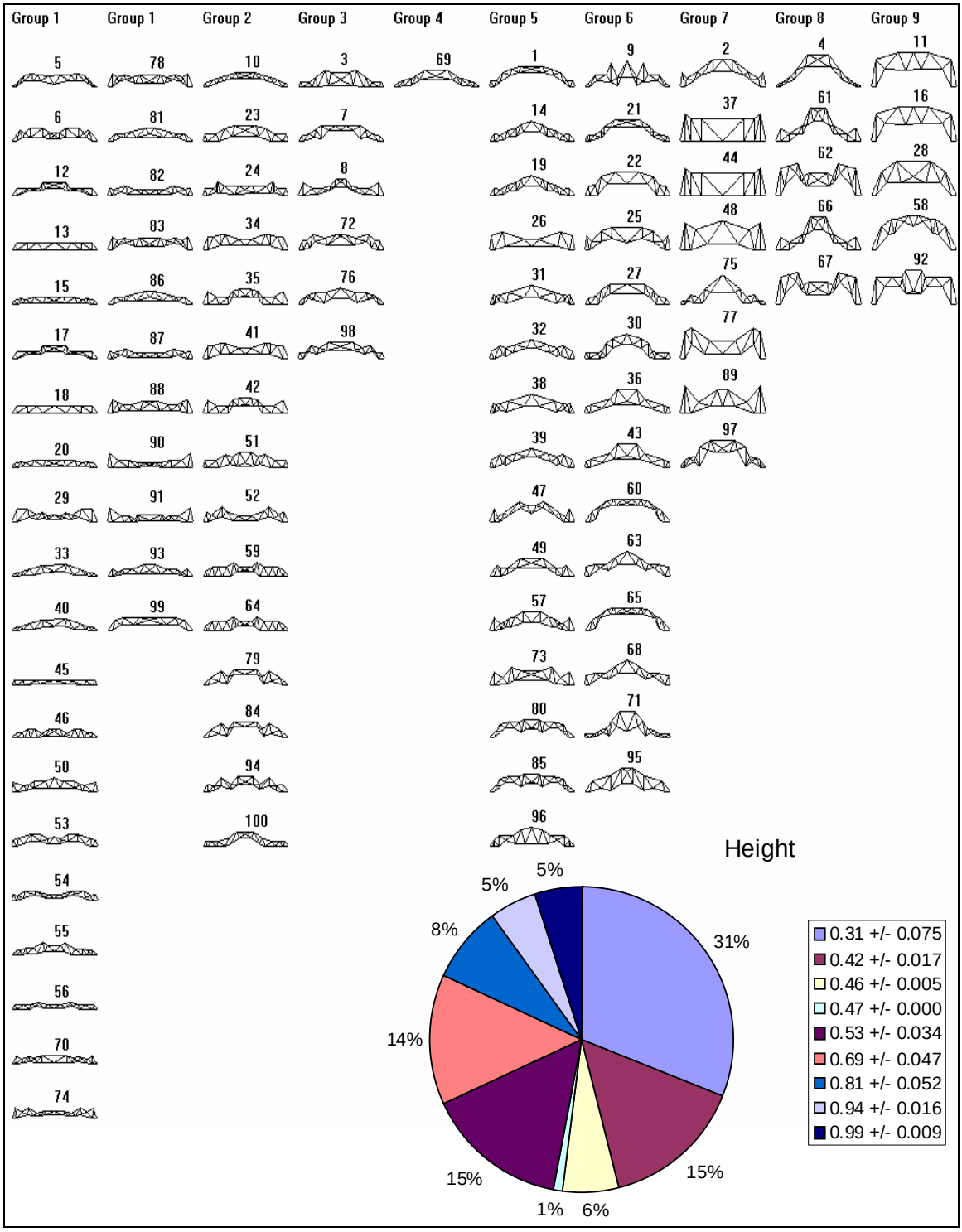


Figure A.2. Feature map created for S100M50 with maximum height as input.

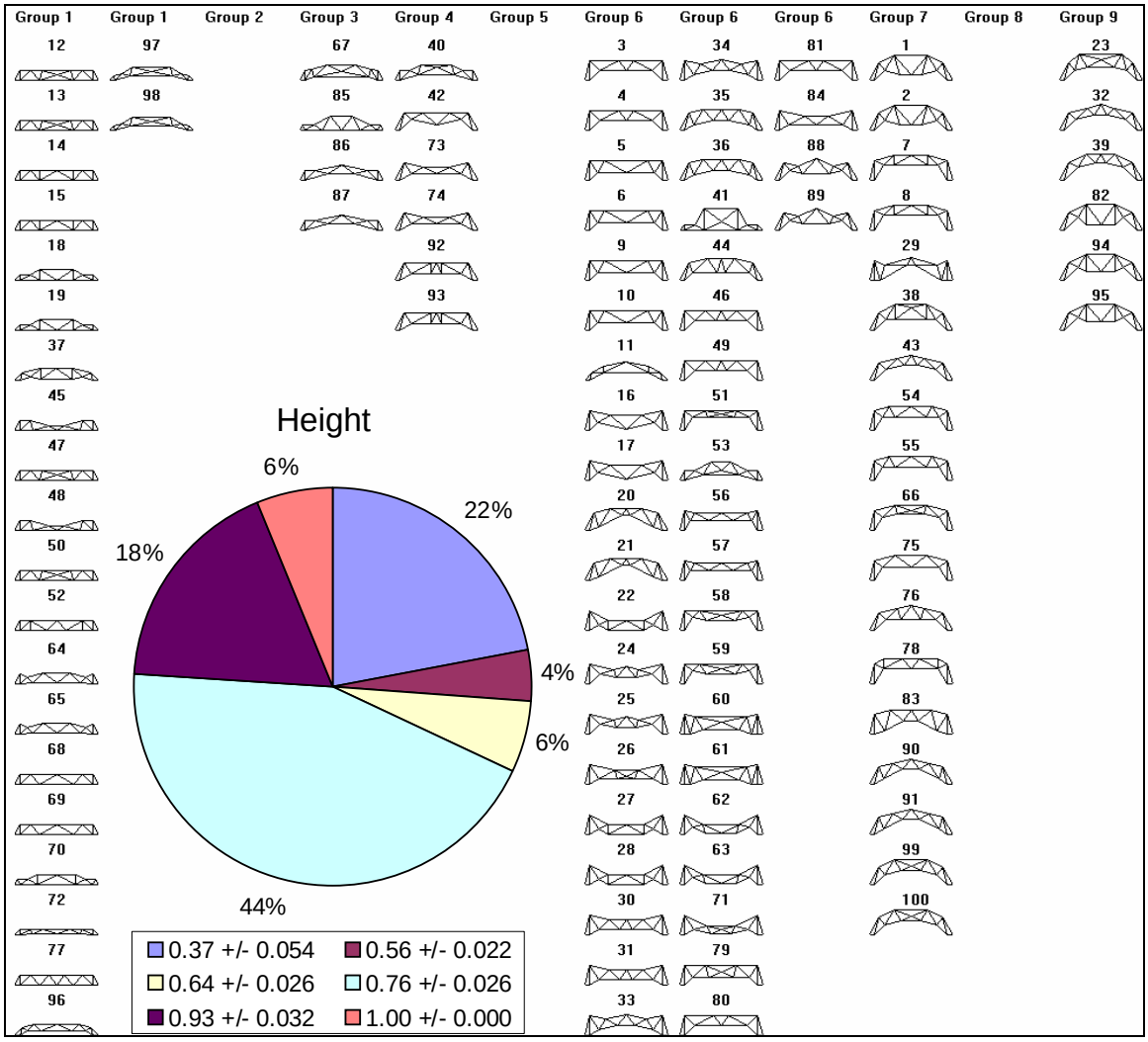


Figure A.3. Feature map created for S100M25 with maximum height as input.

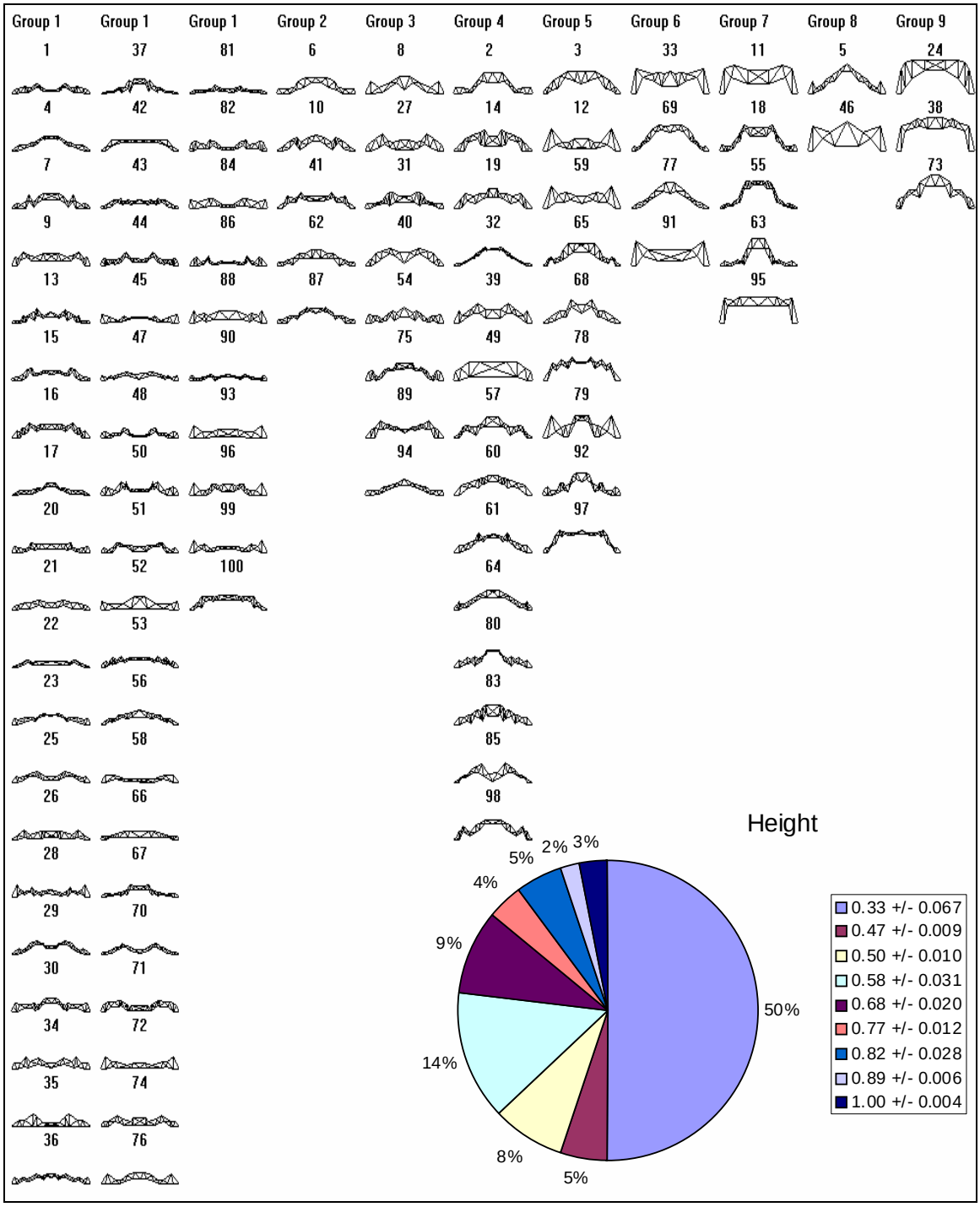


Figure A.4. Feature map created for S100M100 with maximum height as input.

Maximum Joint Connectivity

Figures A.5 through A.8 show feature maps for populations S25M50, S100M50, S100M25, and S100M100, respectively, for maximum joint connectivity.

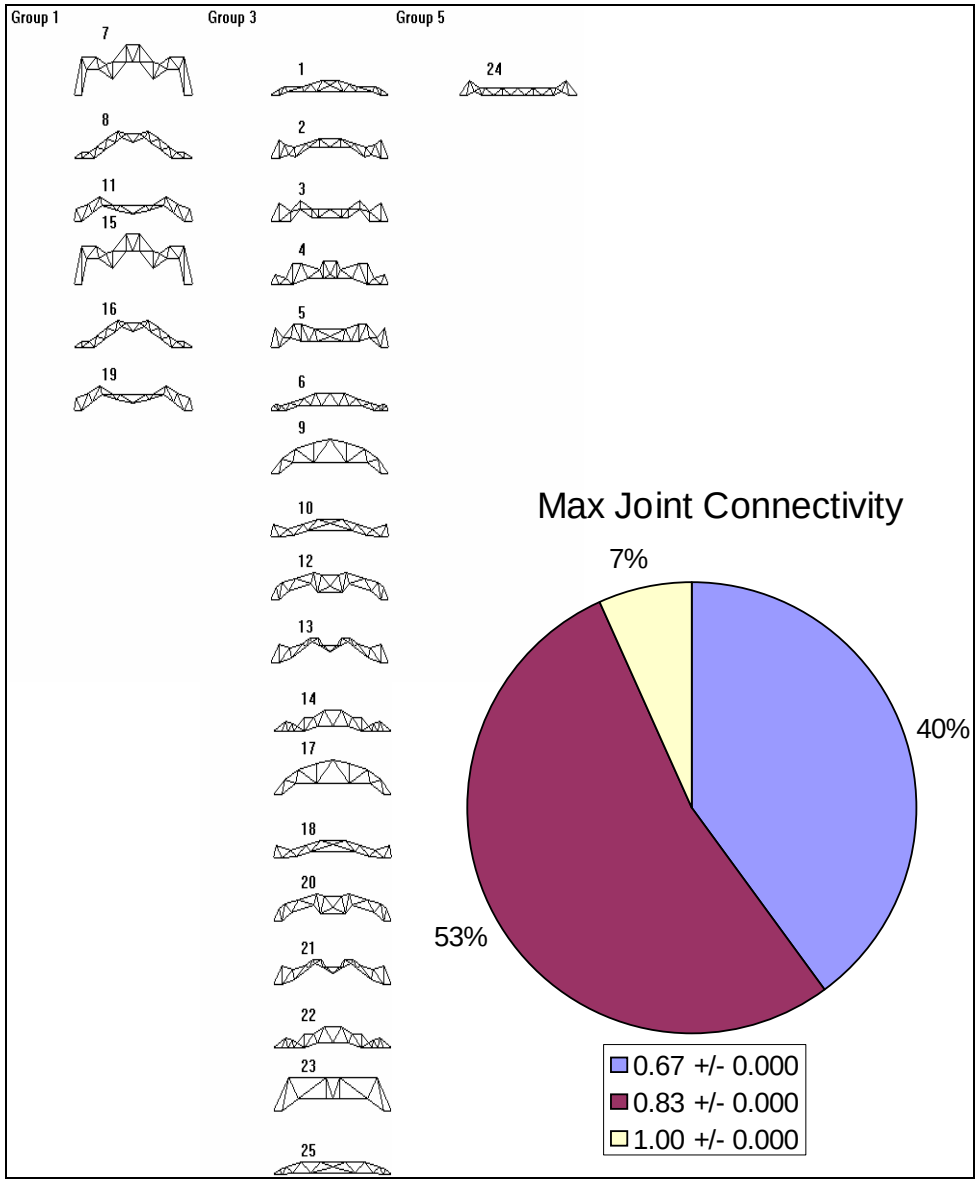


Figure A.5. Feature map created for S25M50 with maximum joint connectivity as input.

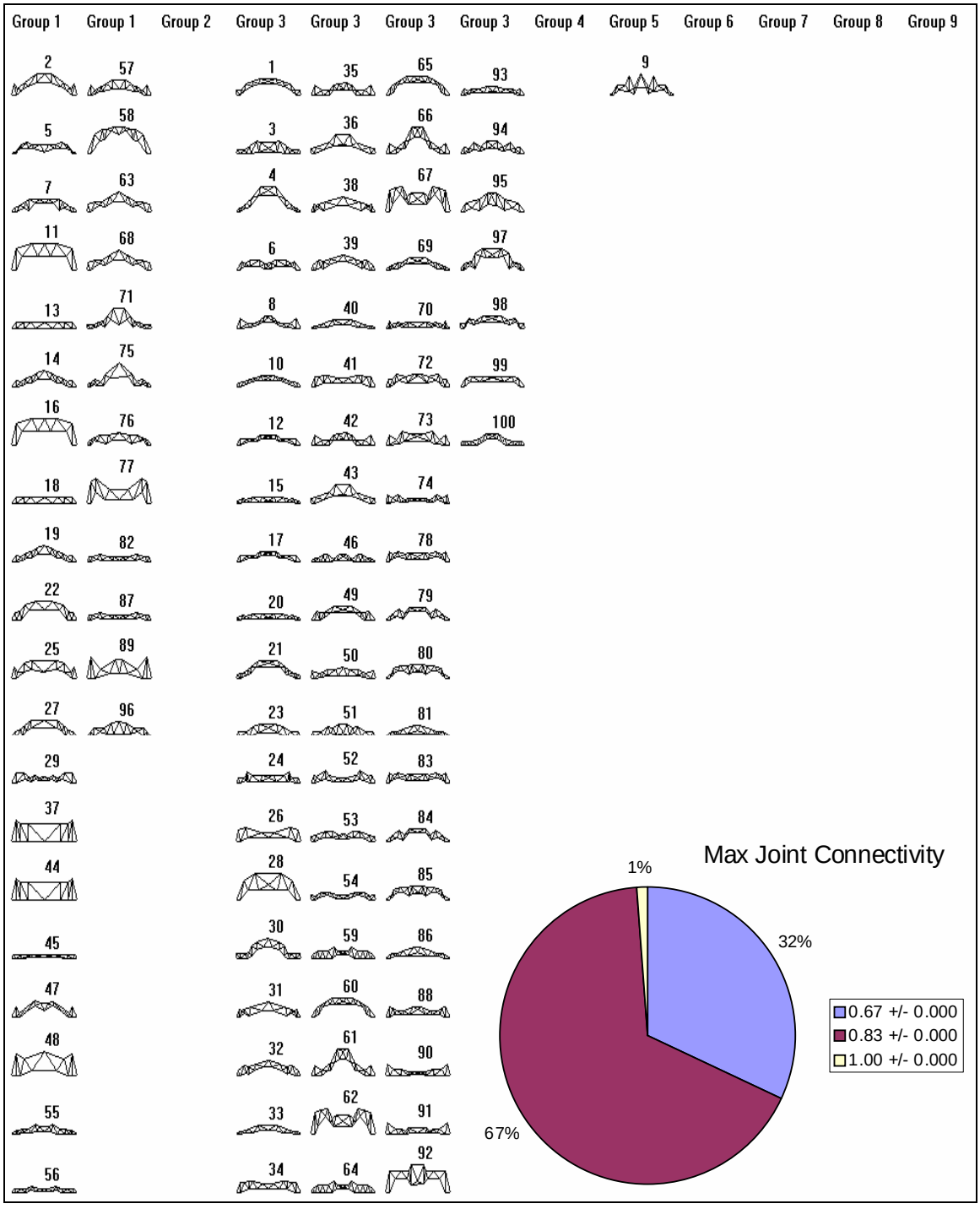


Figure A.6. Feature map created for S100M50 with maximum joint connectivity as input.

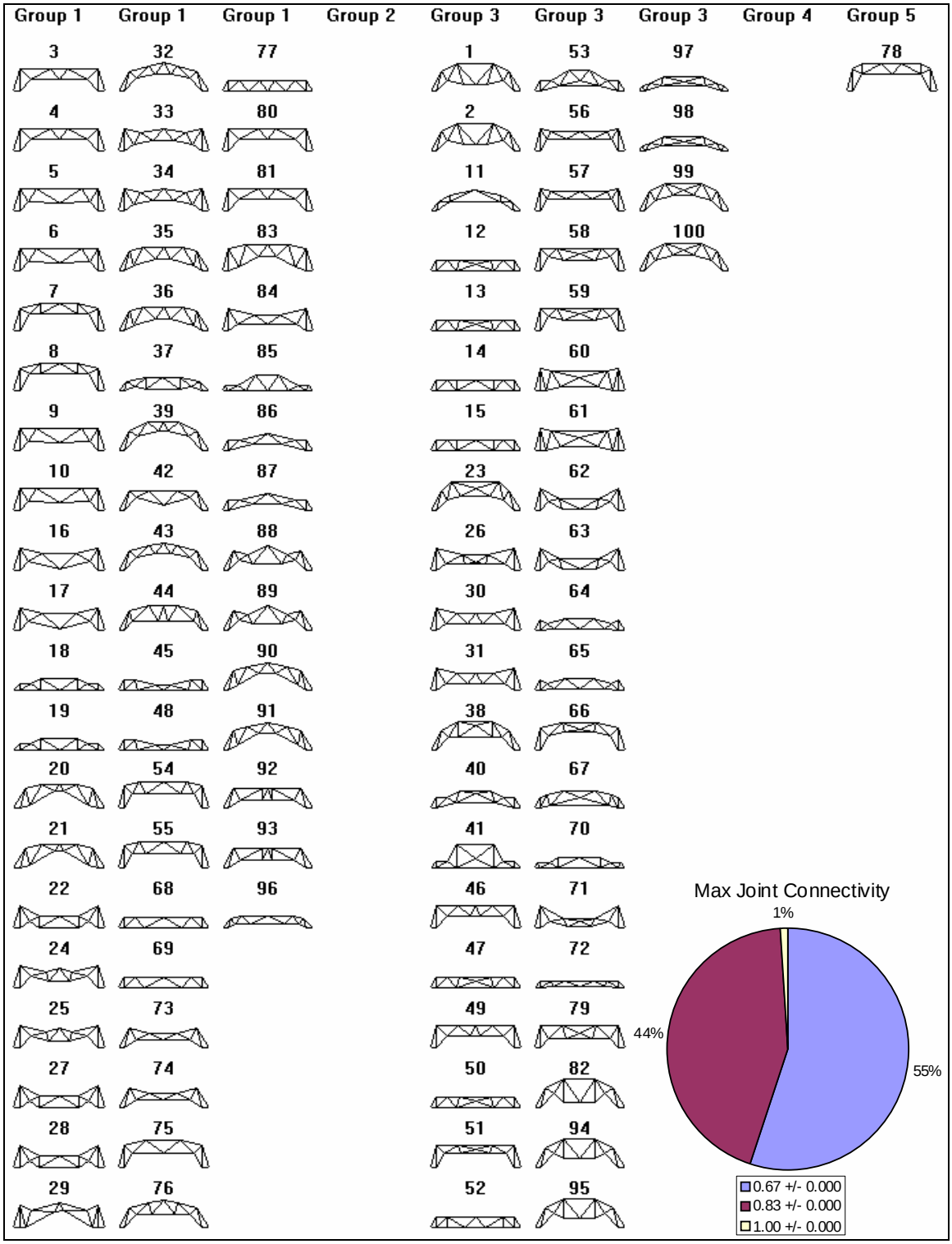


Figure A.7. Feature map created for S100M25 with maximum joint connectivity as input.

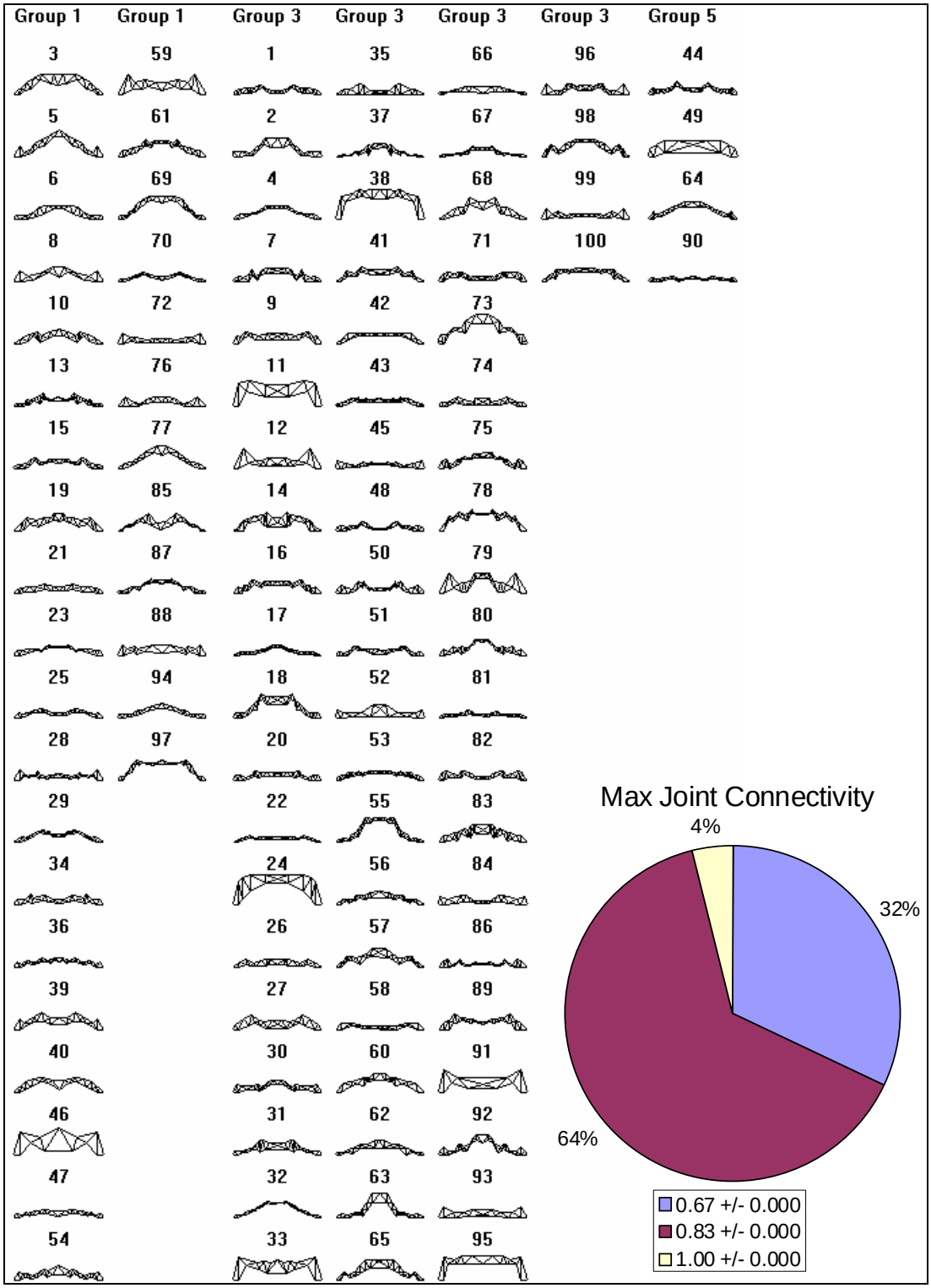


Figure A.8. Feature map created for S100M100 with maximum joint connectivity as input.

Average Joint Connectivity

Figures A.9 through A.12 show feature maps for populations S25M50, S100M50, S100M25, and S100M100, respectively, for average joint connectivity.

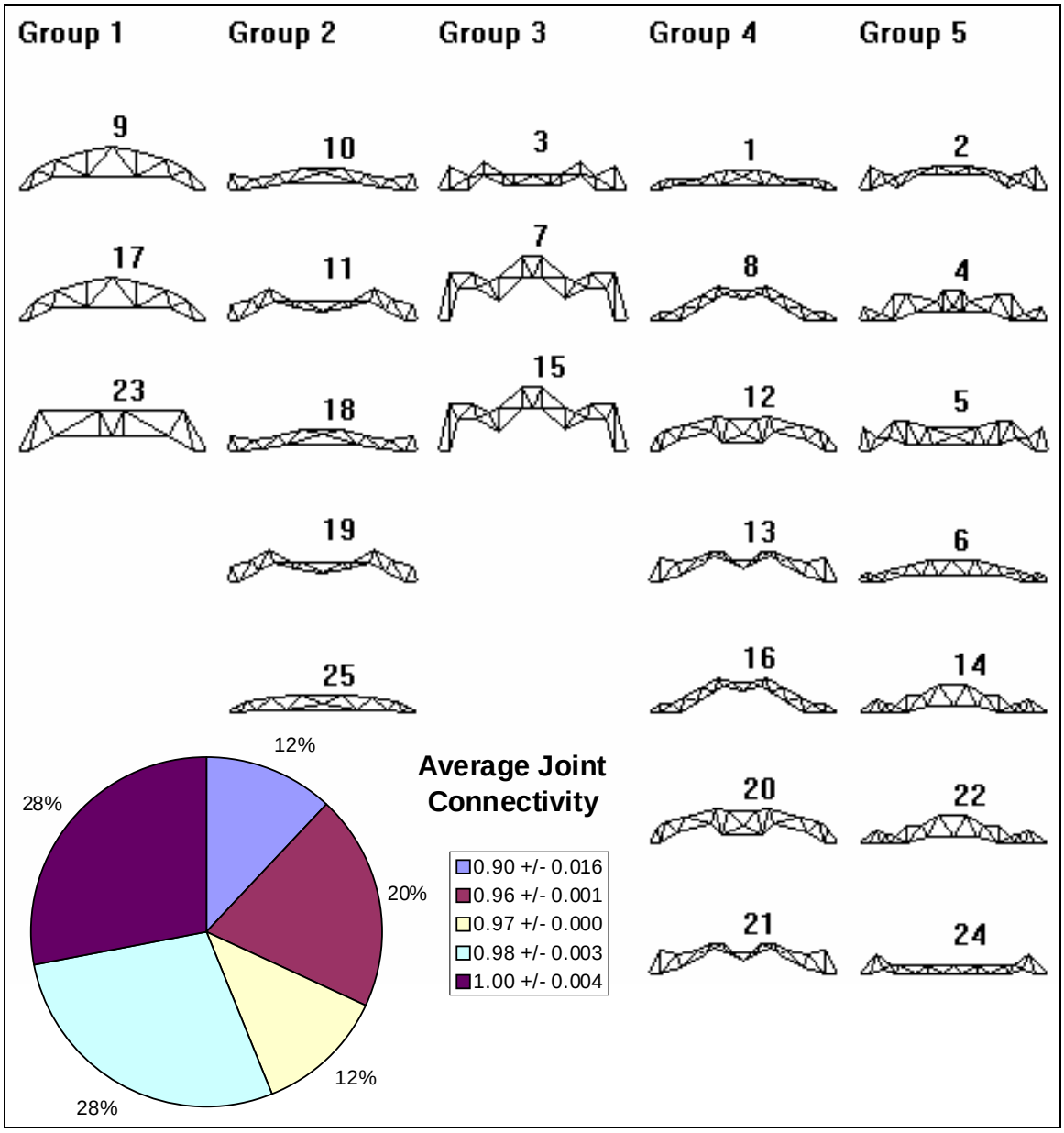


Figure A.9. Feature map created for S25M50 with average joint connectivity as input.

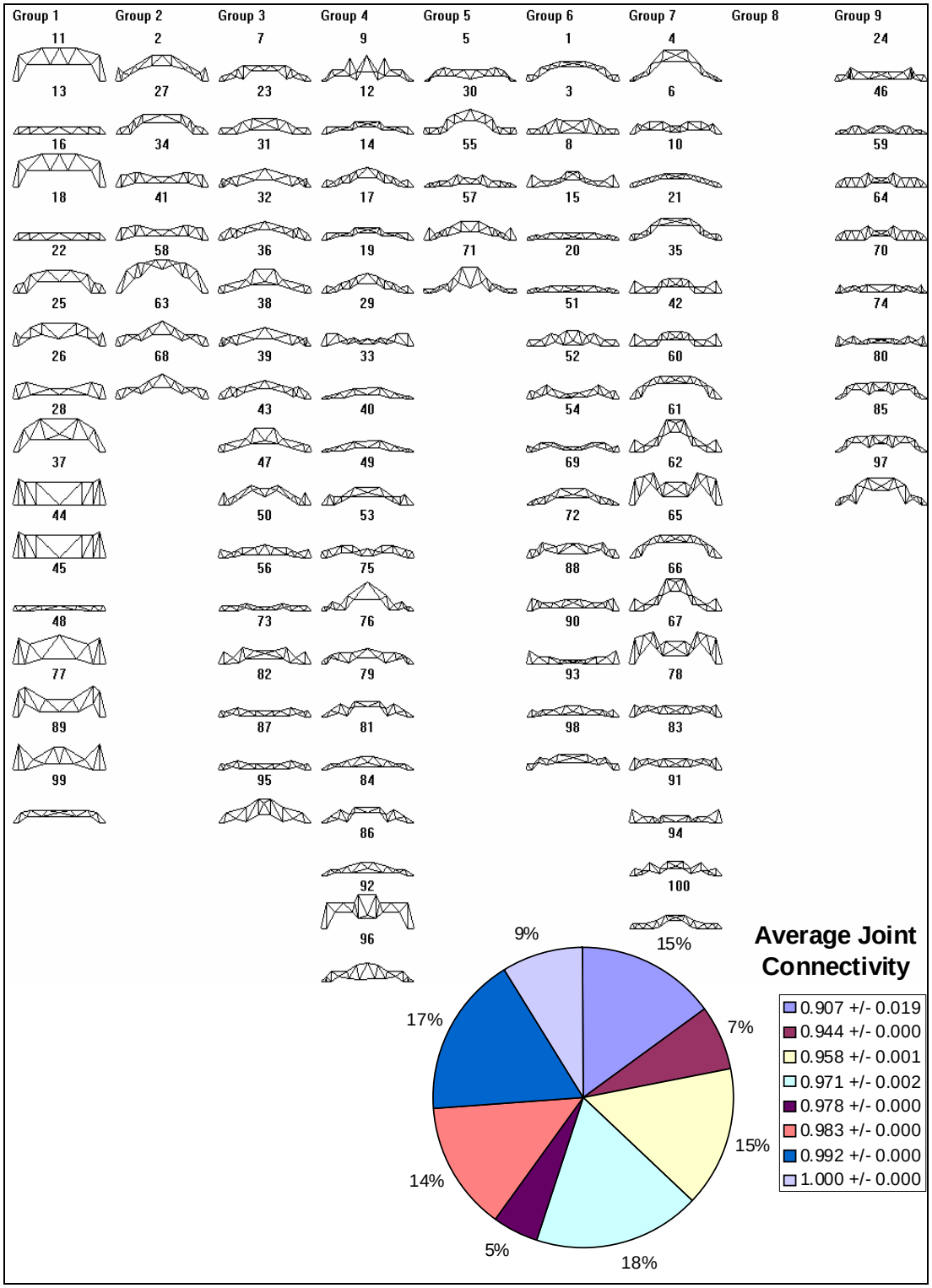


Figure A.10. Feature map created for S100M50 with average joint connectivity as input.

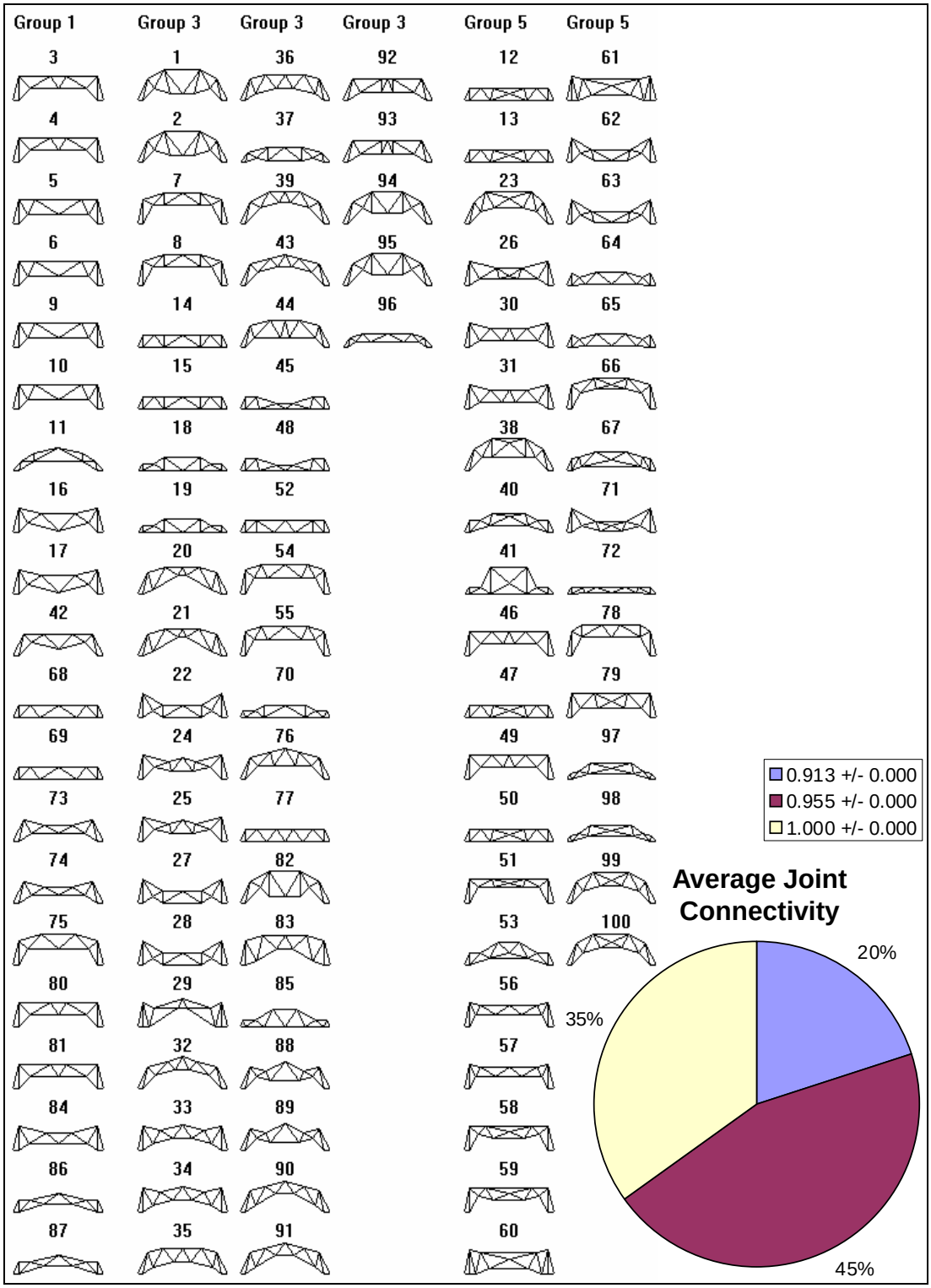


Figure A.11. Feature map created for S100M25 with average joint connectivity as input.

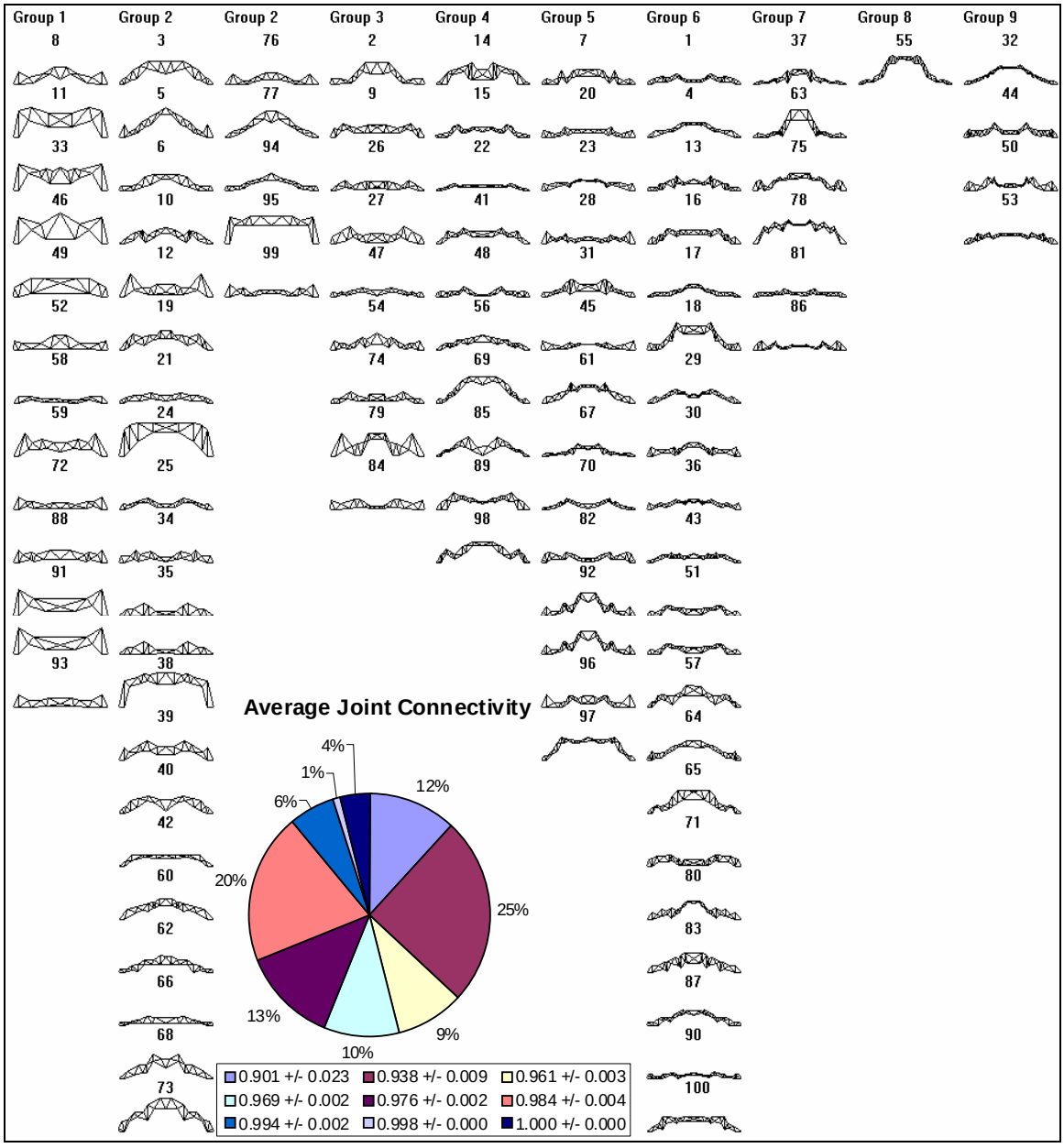


Figure A.12. Feature map created for S100M100 with average joint connectivity as input.

Top Chord Nodes

Figures A.13 through A.16 show feature maps for populations S25M50, S100M50, S100M25, and S100M100, respectively, for the number of top chord nodes.

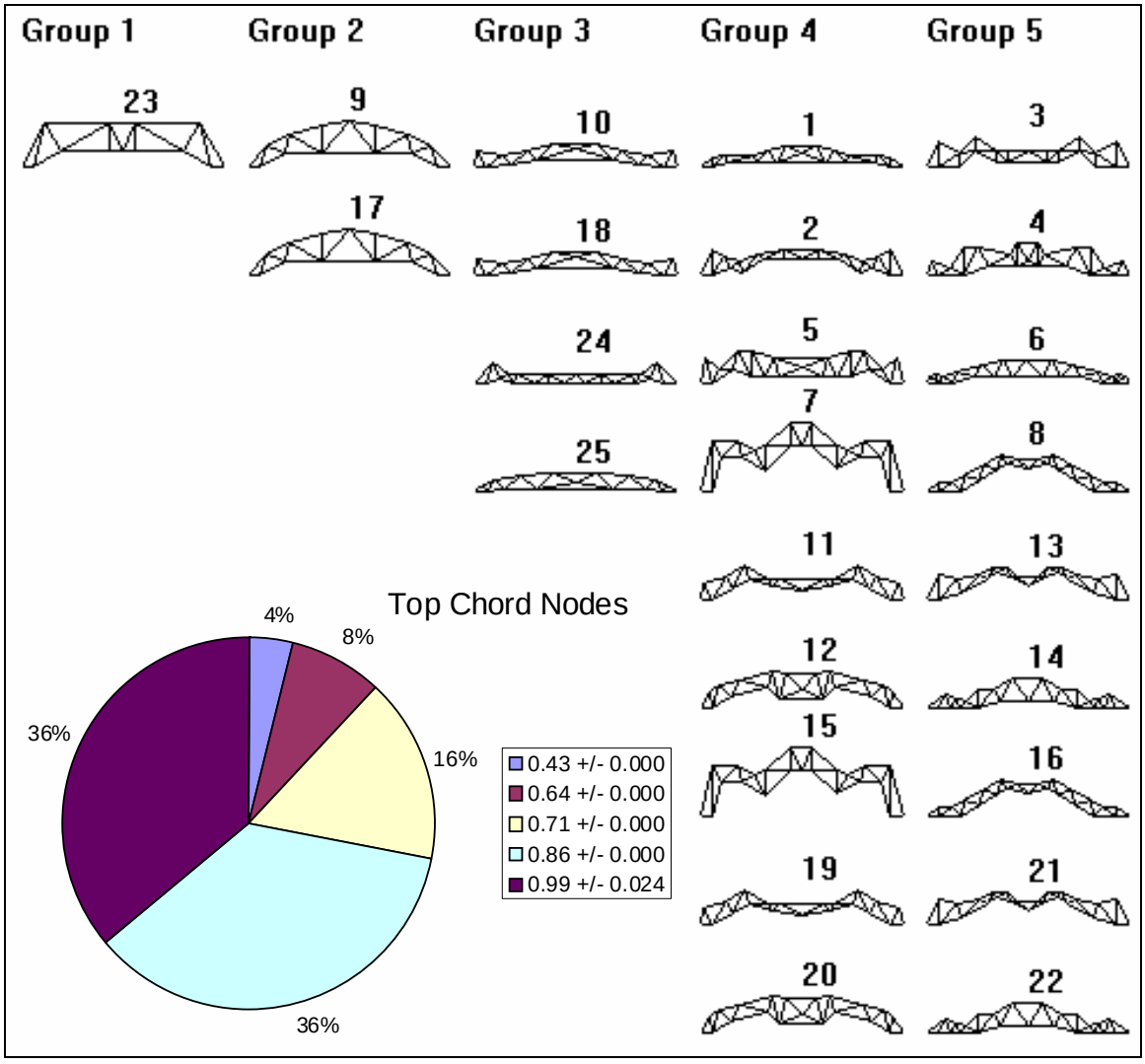


Figure A.13. Feature map created for S25M50 with the number of top chord nodes as input.

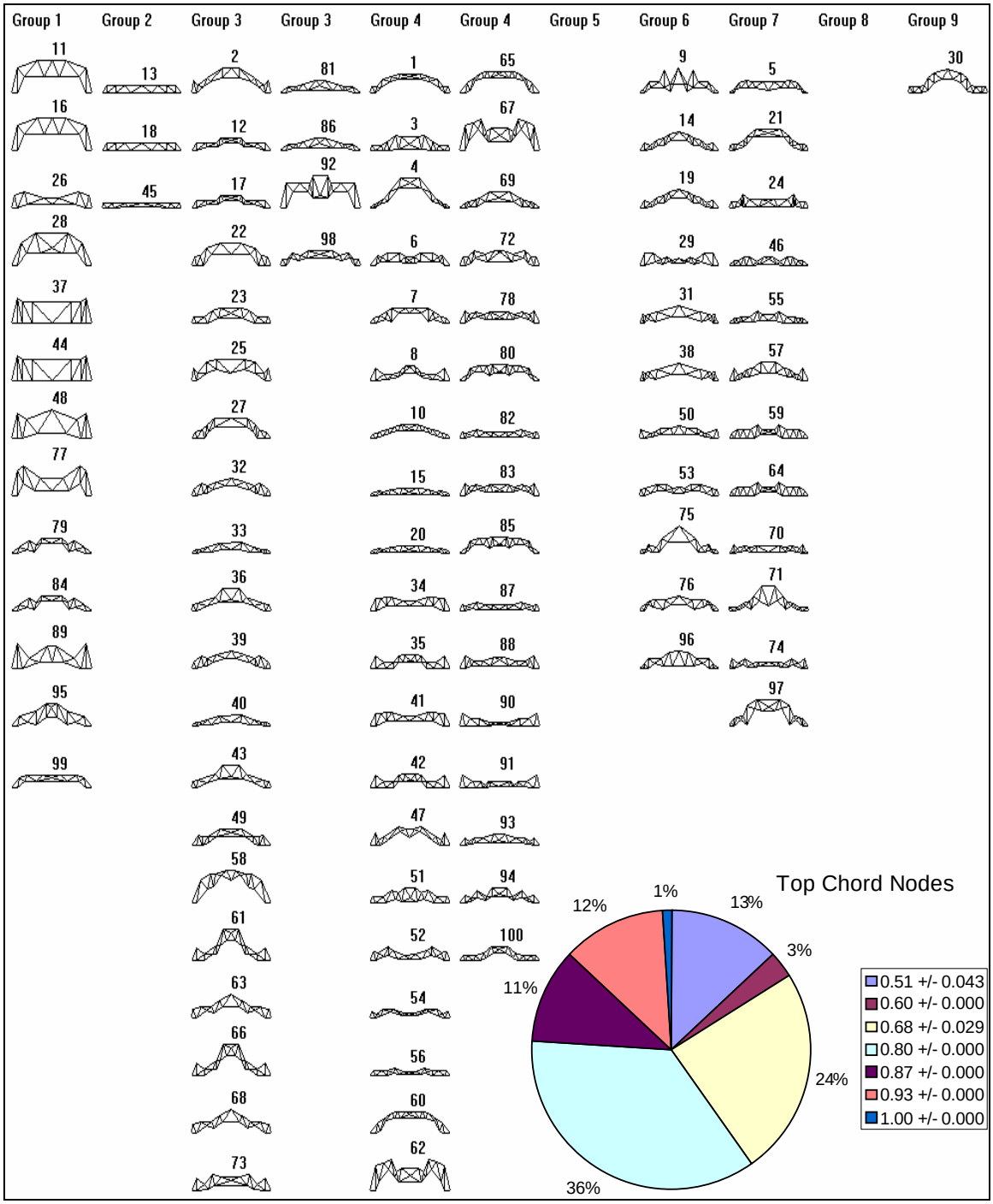


Figure A.14. Feature map created for S100M50 with the number of top chord nodes as input.

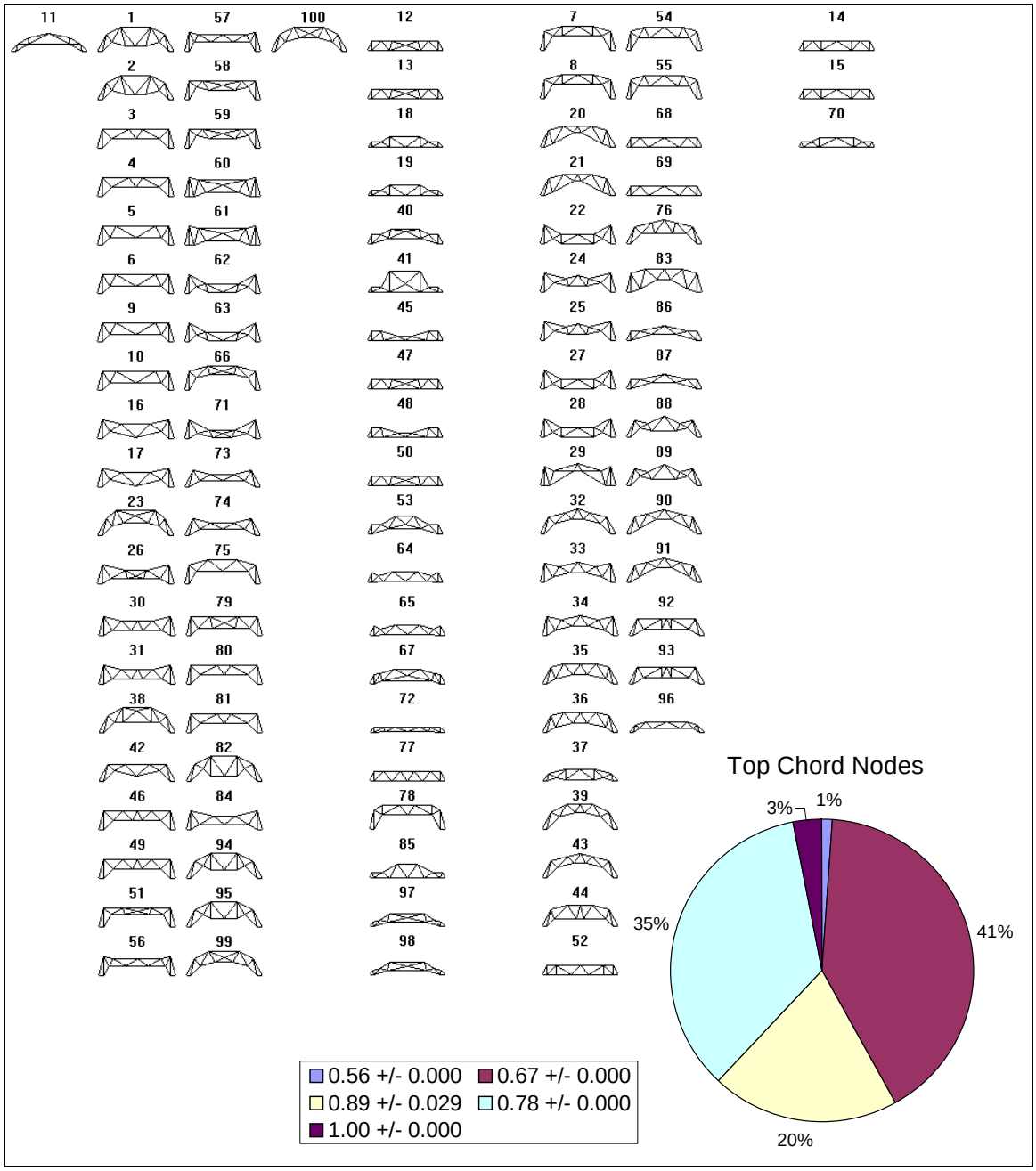


Figure A.15. Feature map created for S100M25 with the number of top chord nodes as input.

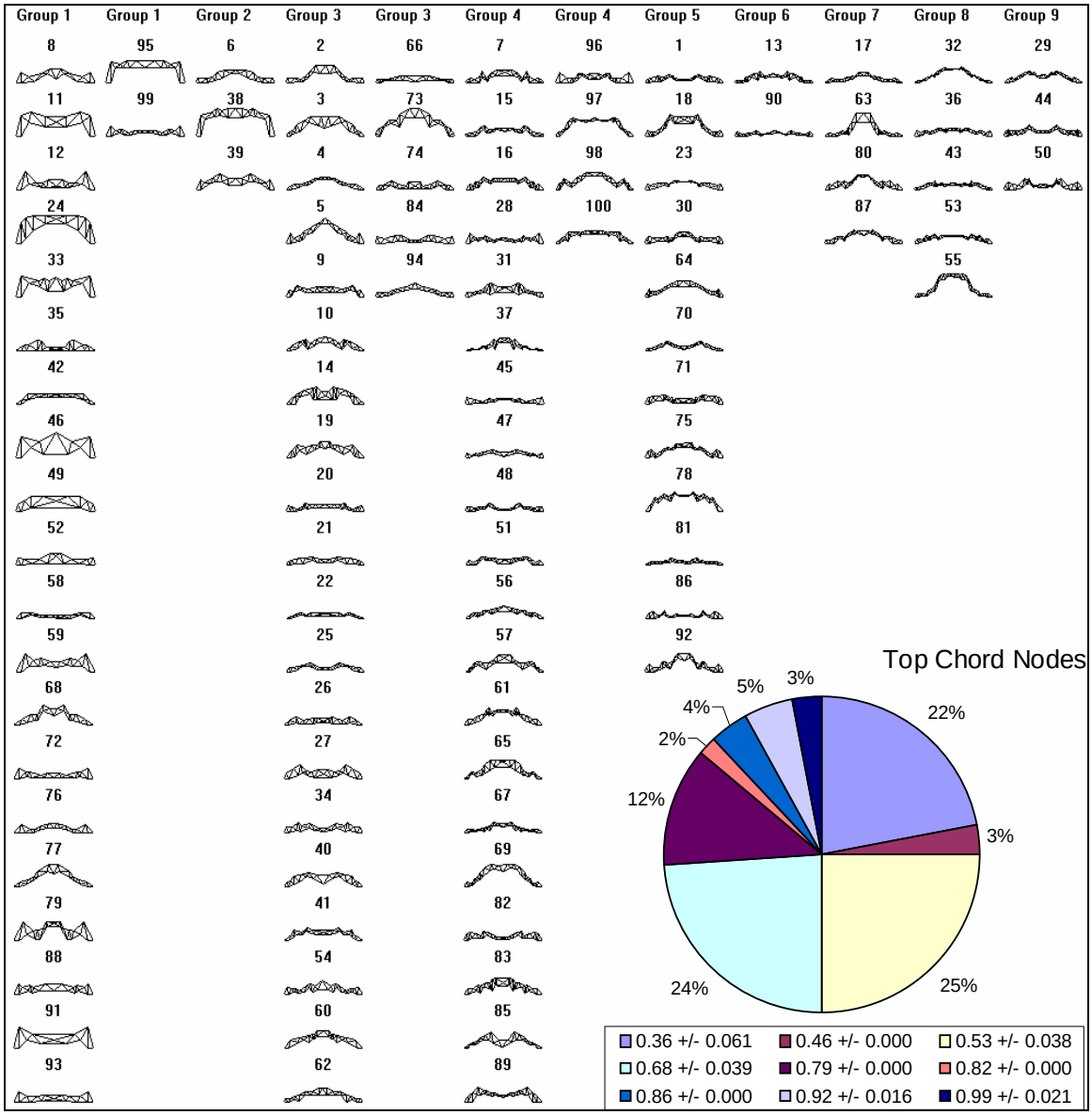


Figure A.16. Feature map created for S100M100 with the number of top chord nodes as input.

Total Member Length

Figures A.17 through A.20 show feature maps for populations S25M50, S100M50, S100M25, and S100M100, respectively, for total member length.

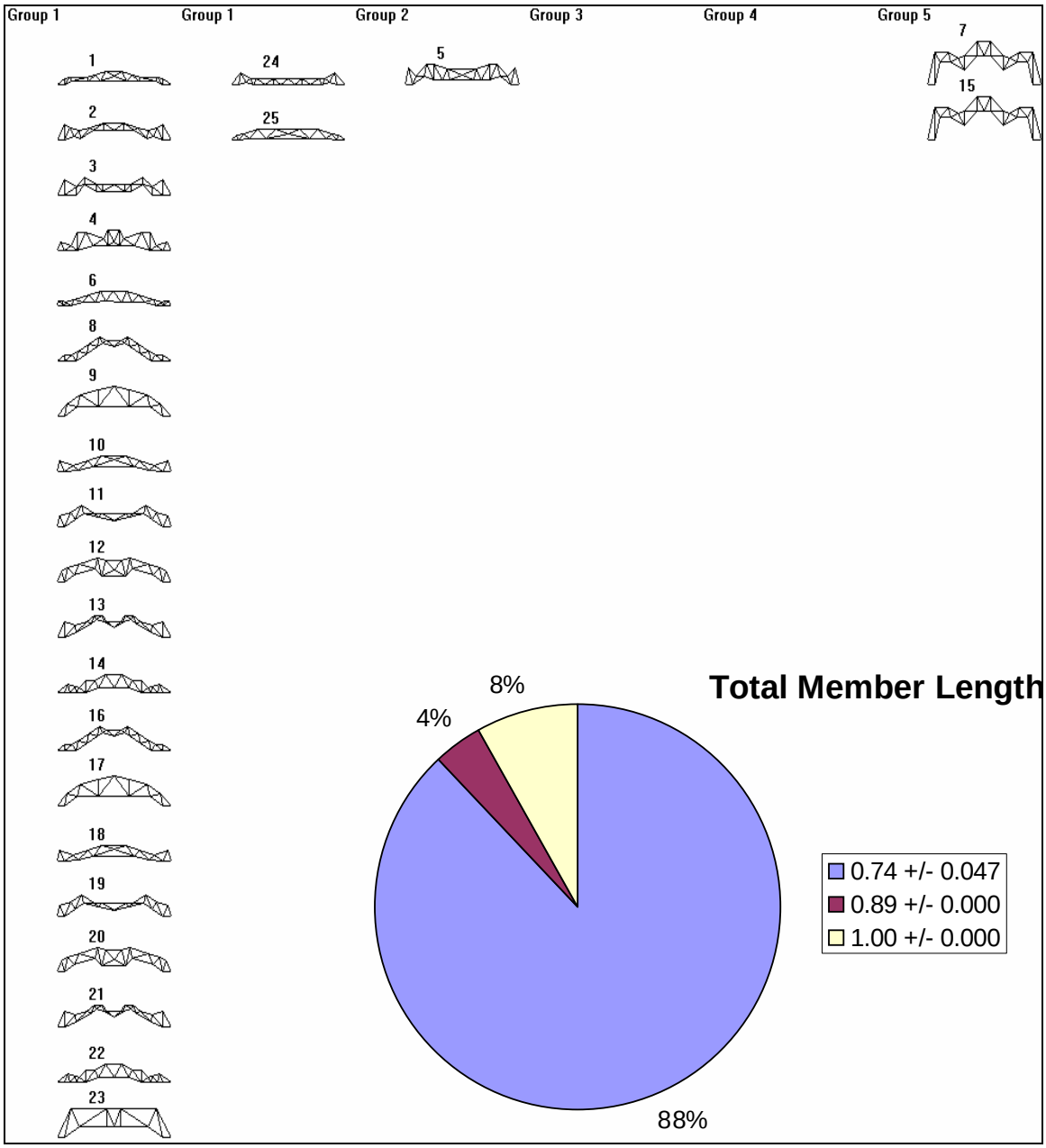


Figure A.17. Feature map created for S25M50 with total member length as input.

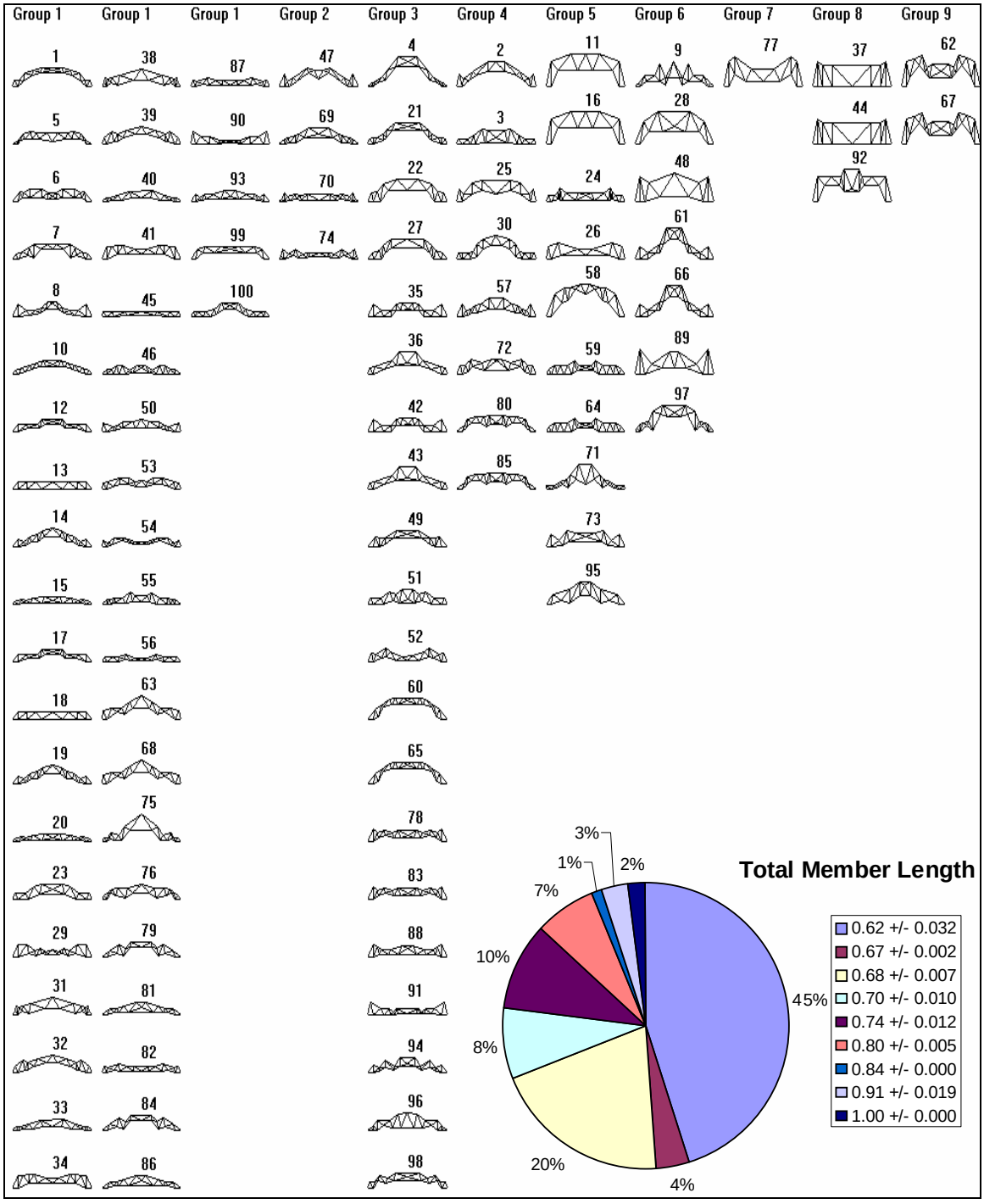


Figure A.18. Feature map created for S100M50 with total member length as input.

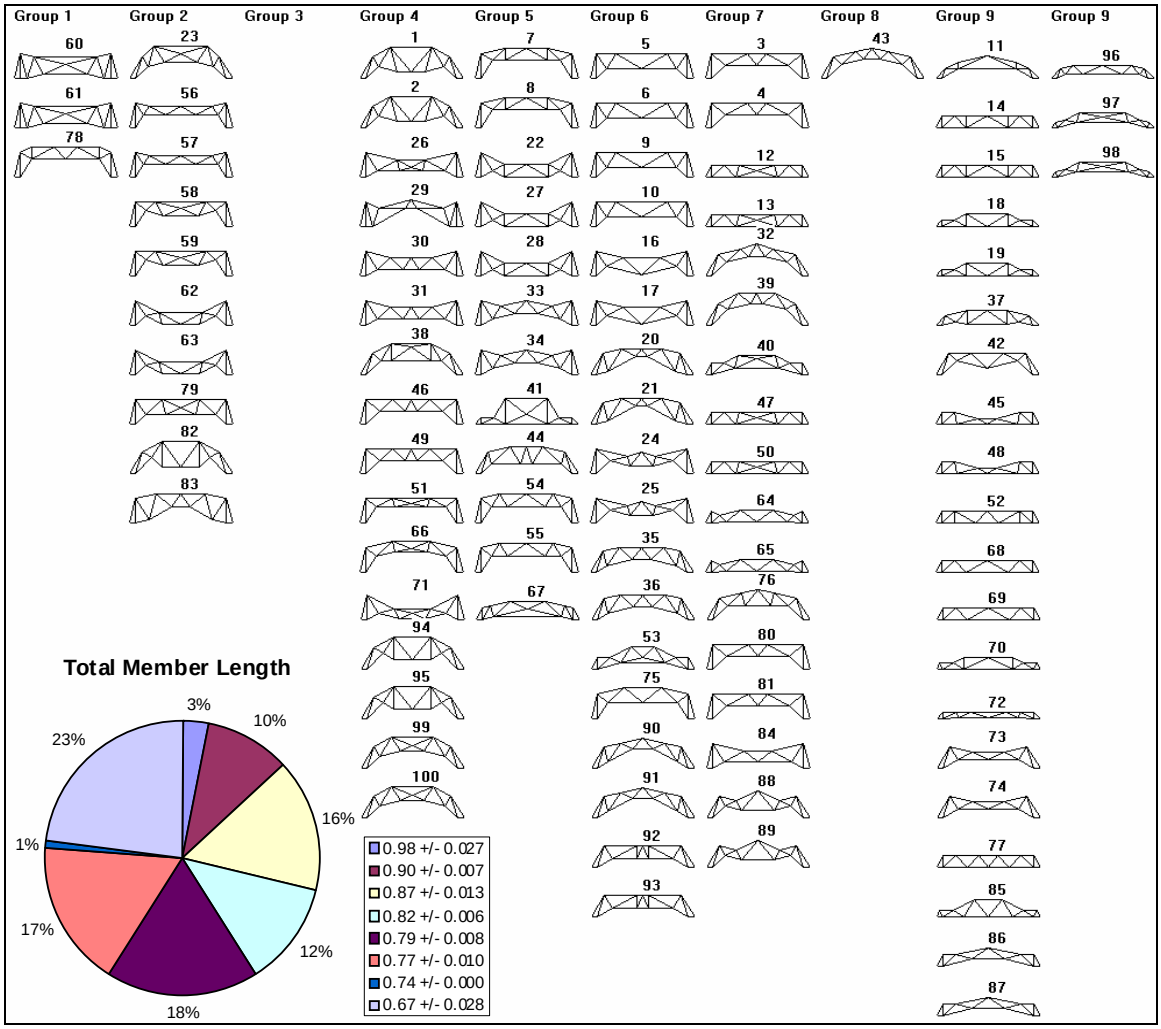


Figure A.19. Feature map created for S100M25 with total member length as input.

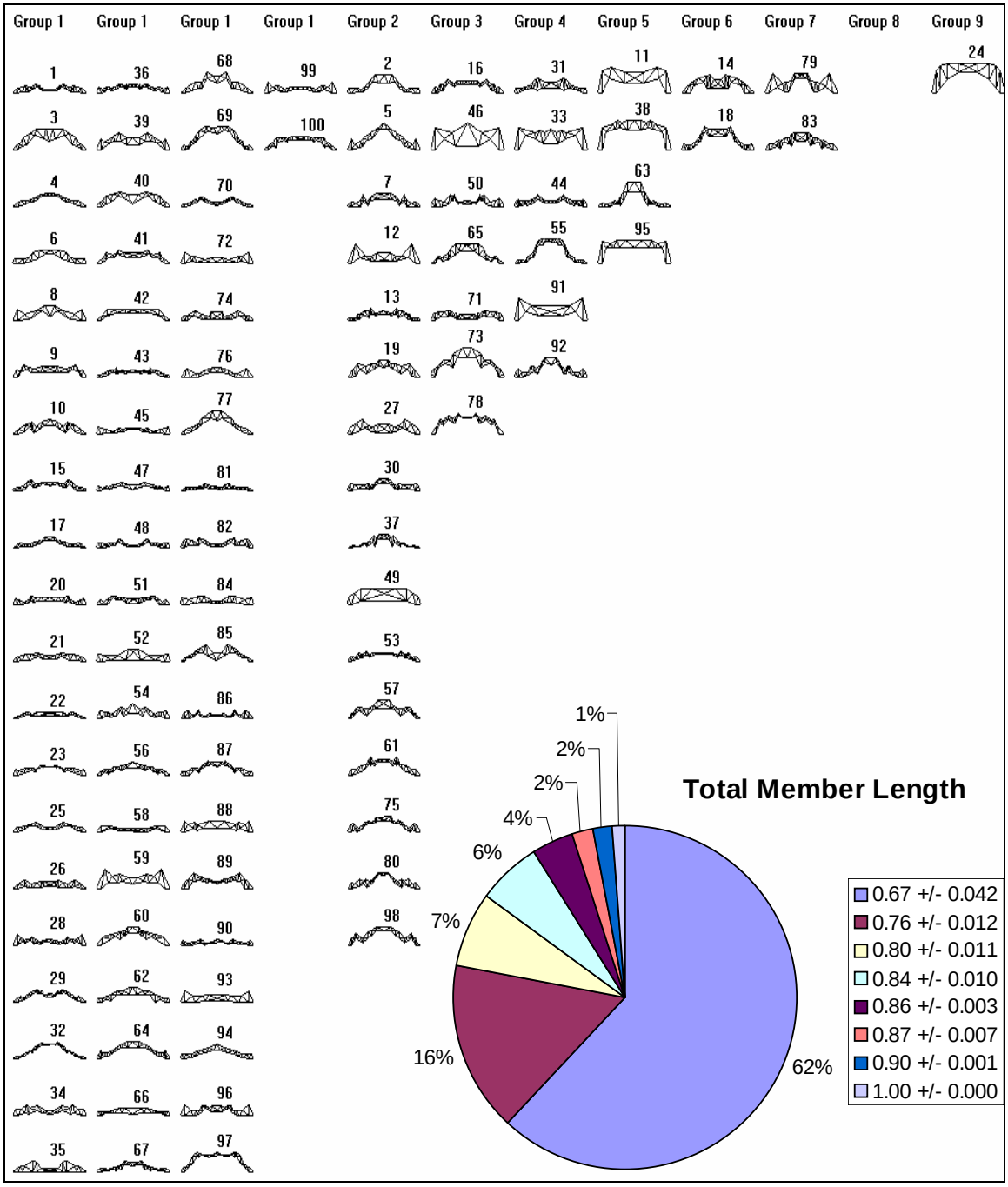


Figure A.20. Feature map created for S100M100 with total member length as input.

Ratio of Compression to Tension Members

Figures A.21 through A.24 show feature maps for populations S25M50,

S100M50, S100M25, and S100M100, respectively, for the ratio of compression to tension members.

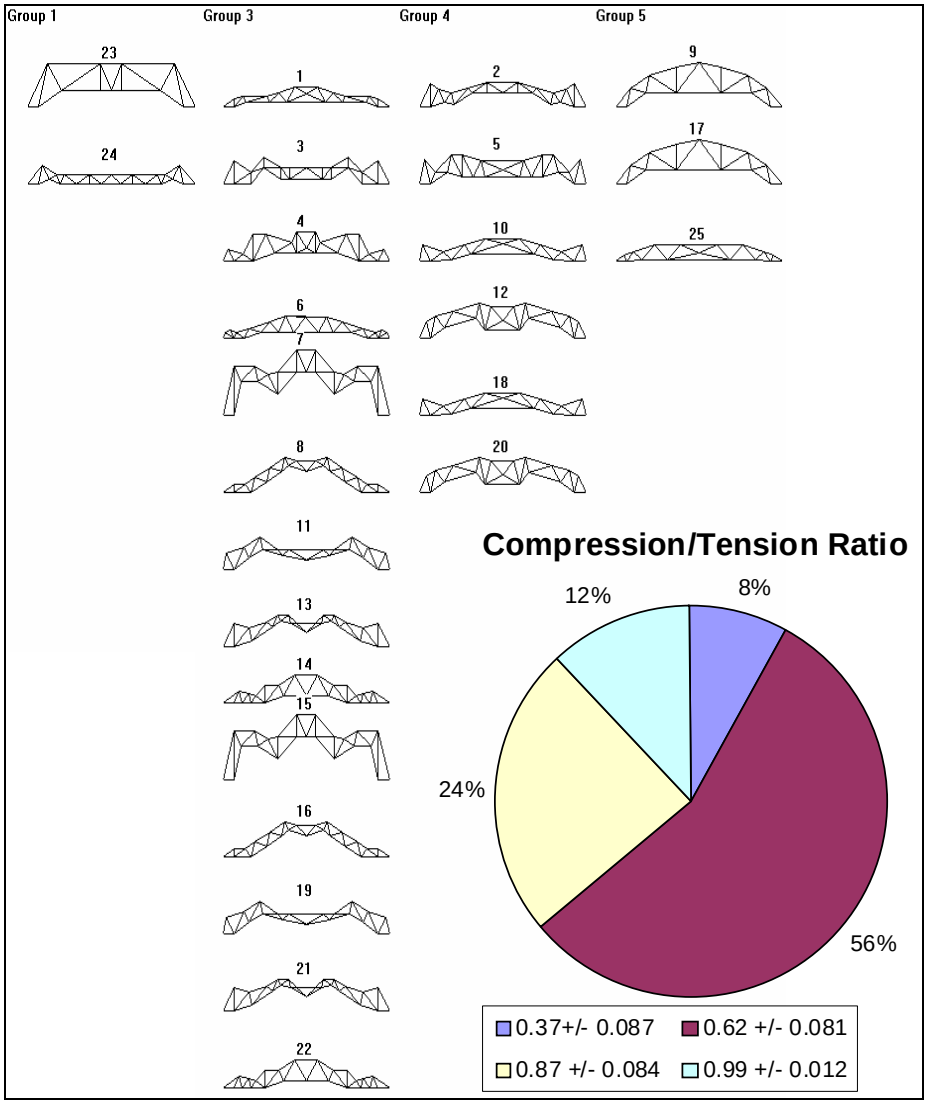


Figure A.21. Feature map created for S25M50 with compression-tension member ratio as input.

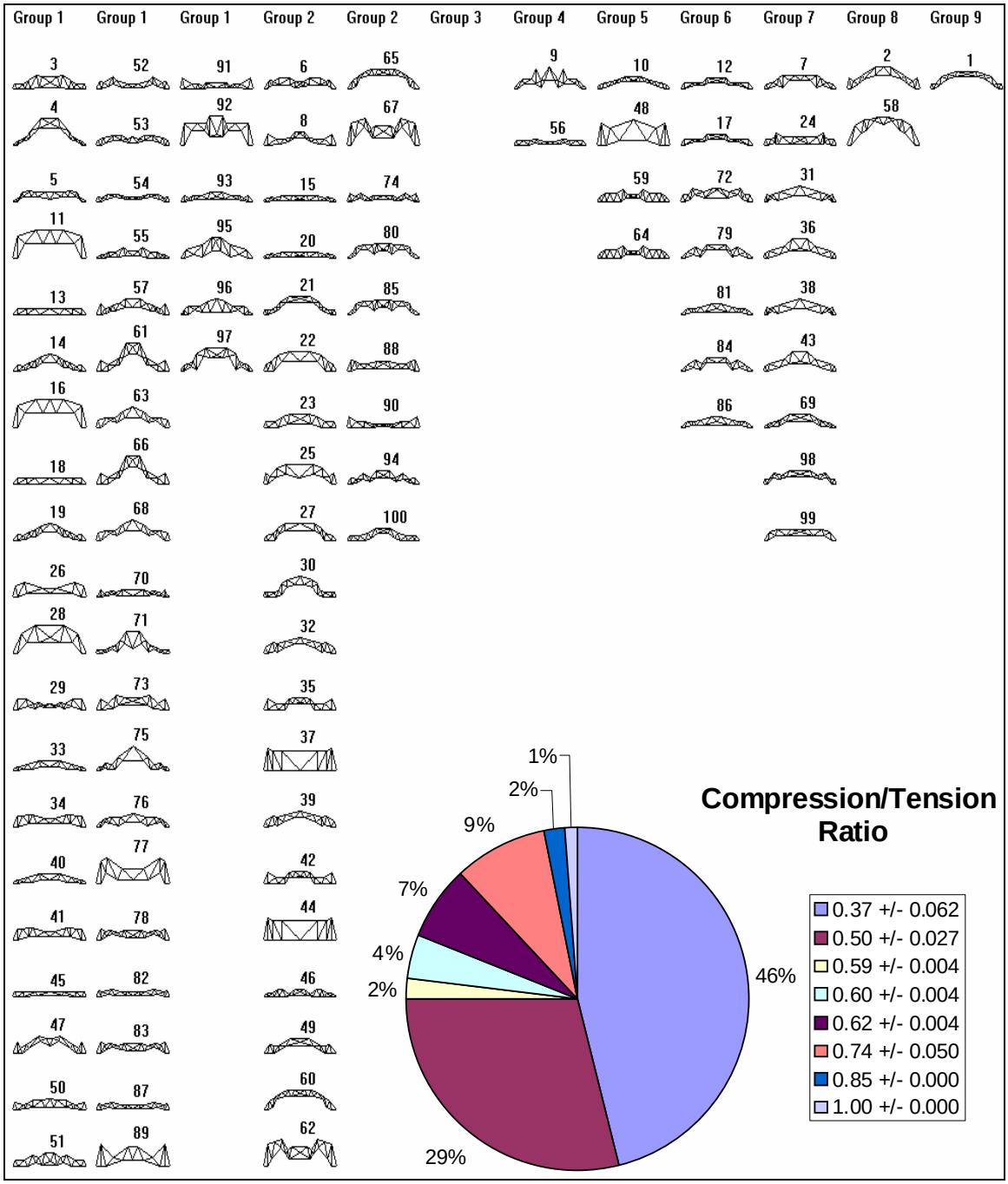


Figure A.22. Feature map created for S100M50 with compression-tension member ratio as input.

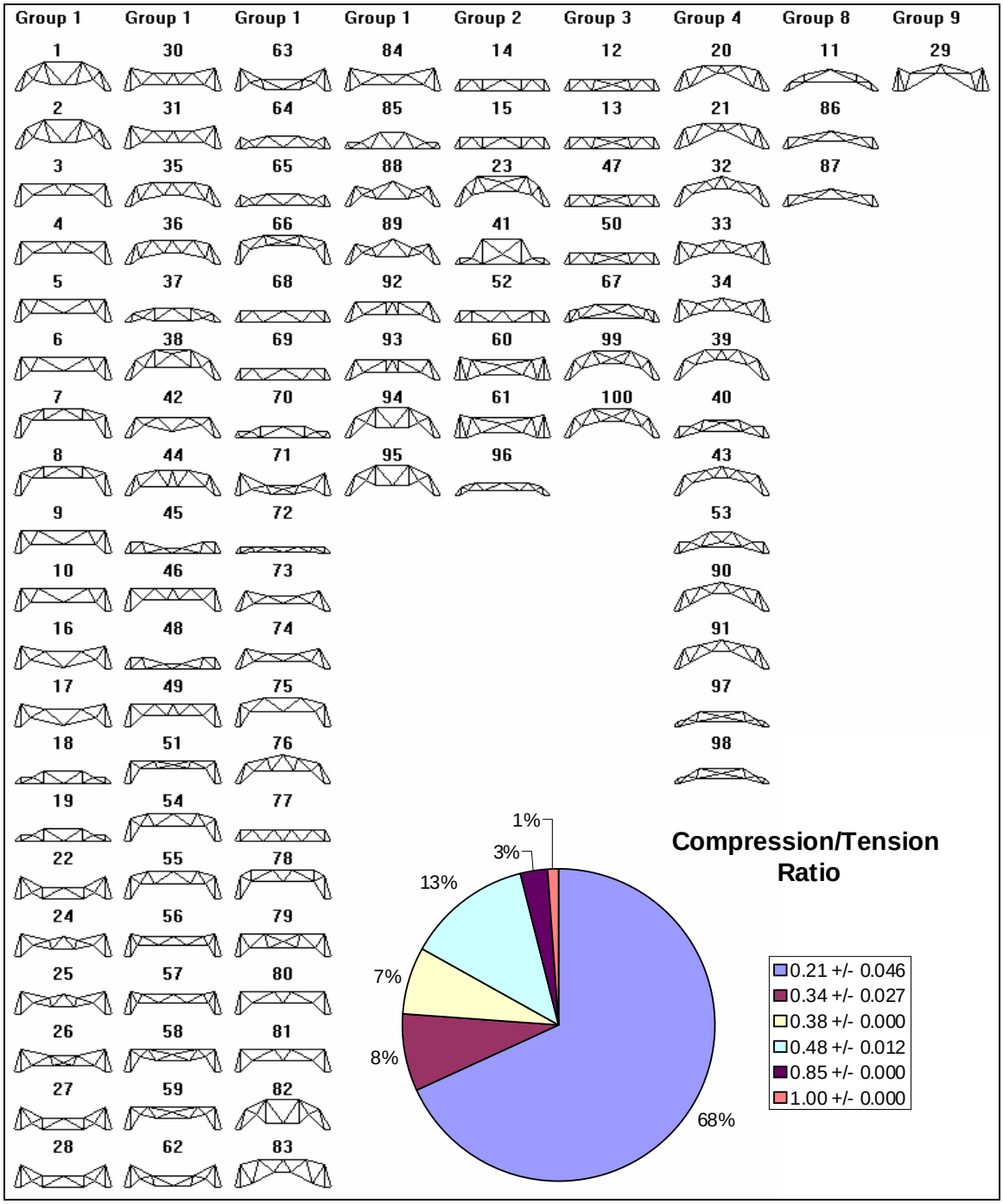


Figure A.23. Feature map created for S100M25 with compression-tension member ratio as input.

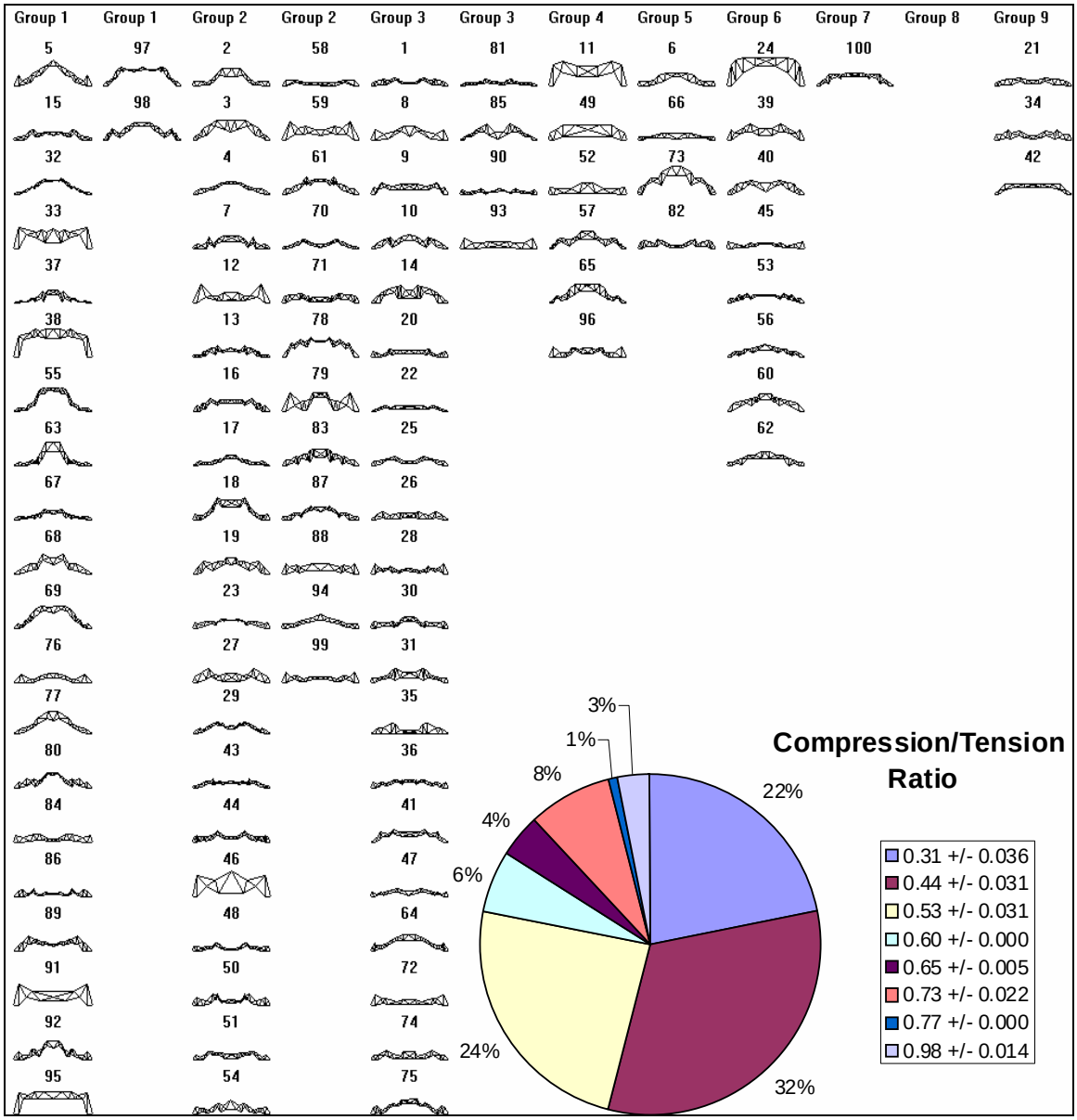


Figure A.24. Feature map created for S100M100 with compression-tension member ratio as input.

Mid-Span Clearance

Figures A.25 through A.28 show feature maps for populations S25M50, S100M50, S100M25, and S100M100, respectively, for mid-span clearance.

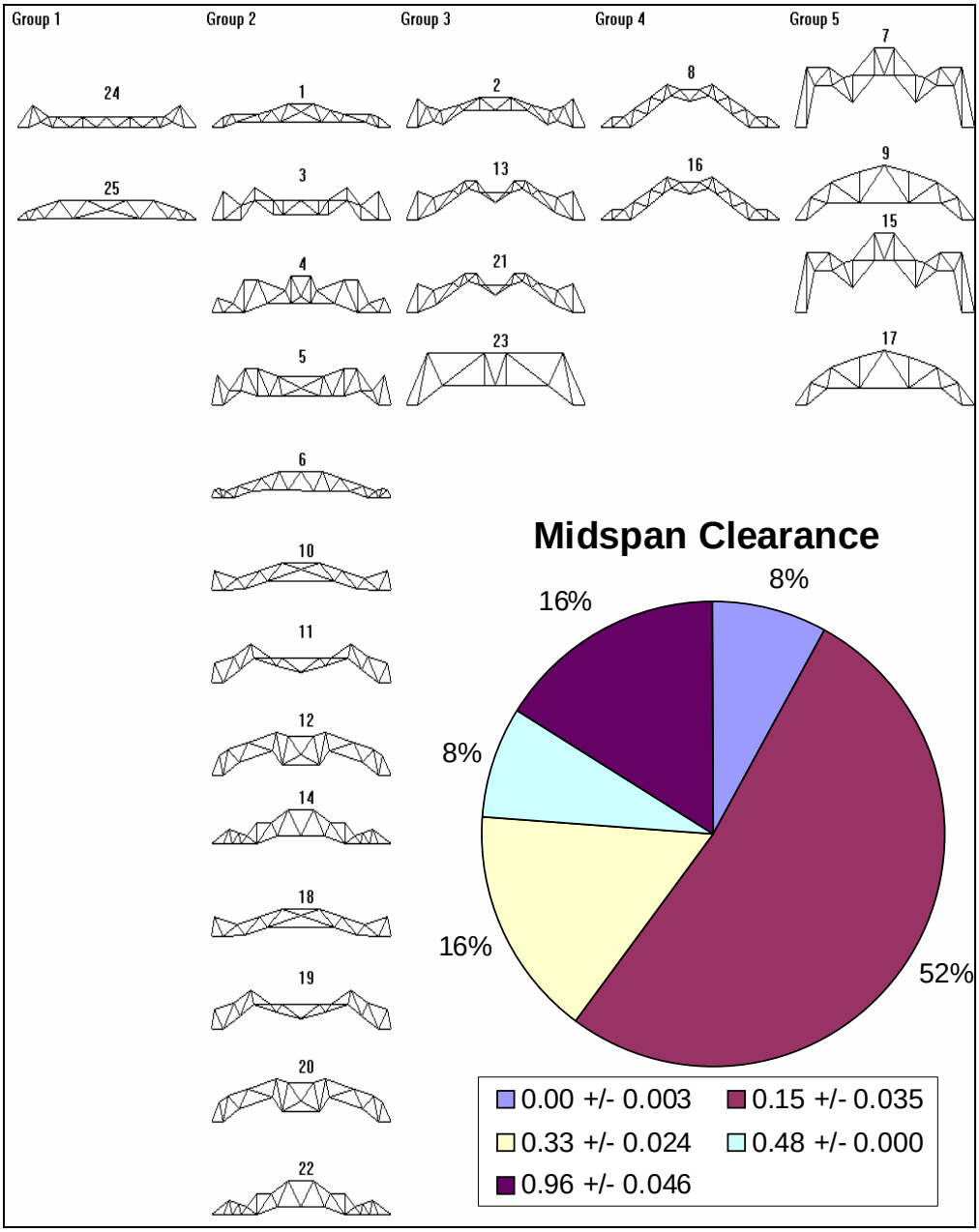


Figure A.25. Feature map created for S25M50 with mid-span clearance as input.

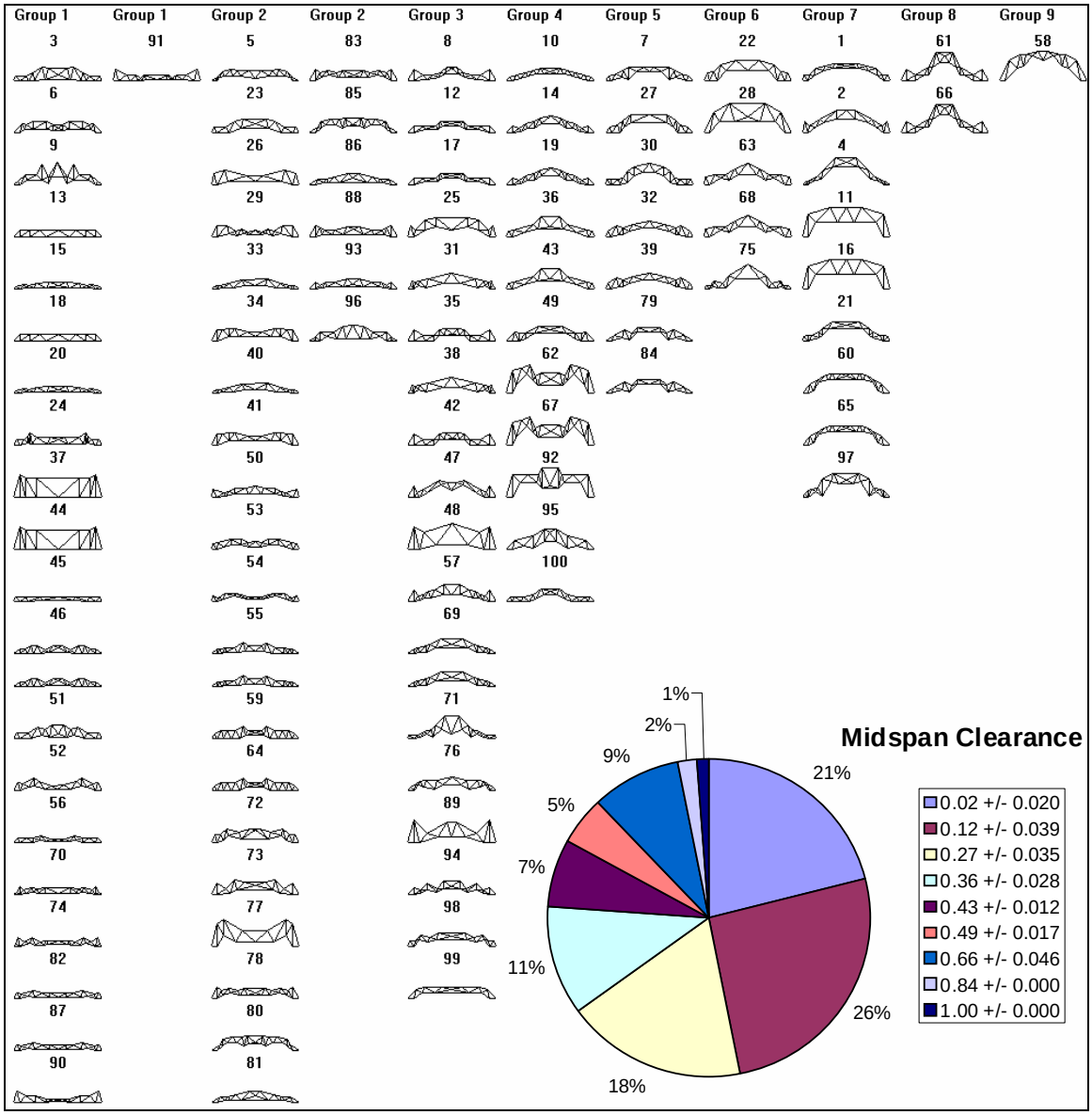


Figure A.26. Feature map created for S100M50 with mid-span clearance as input.

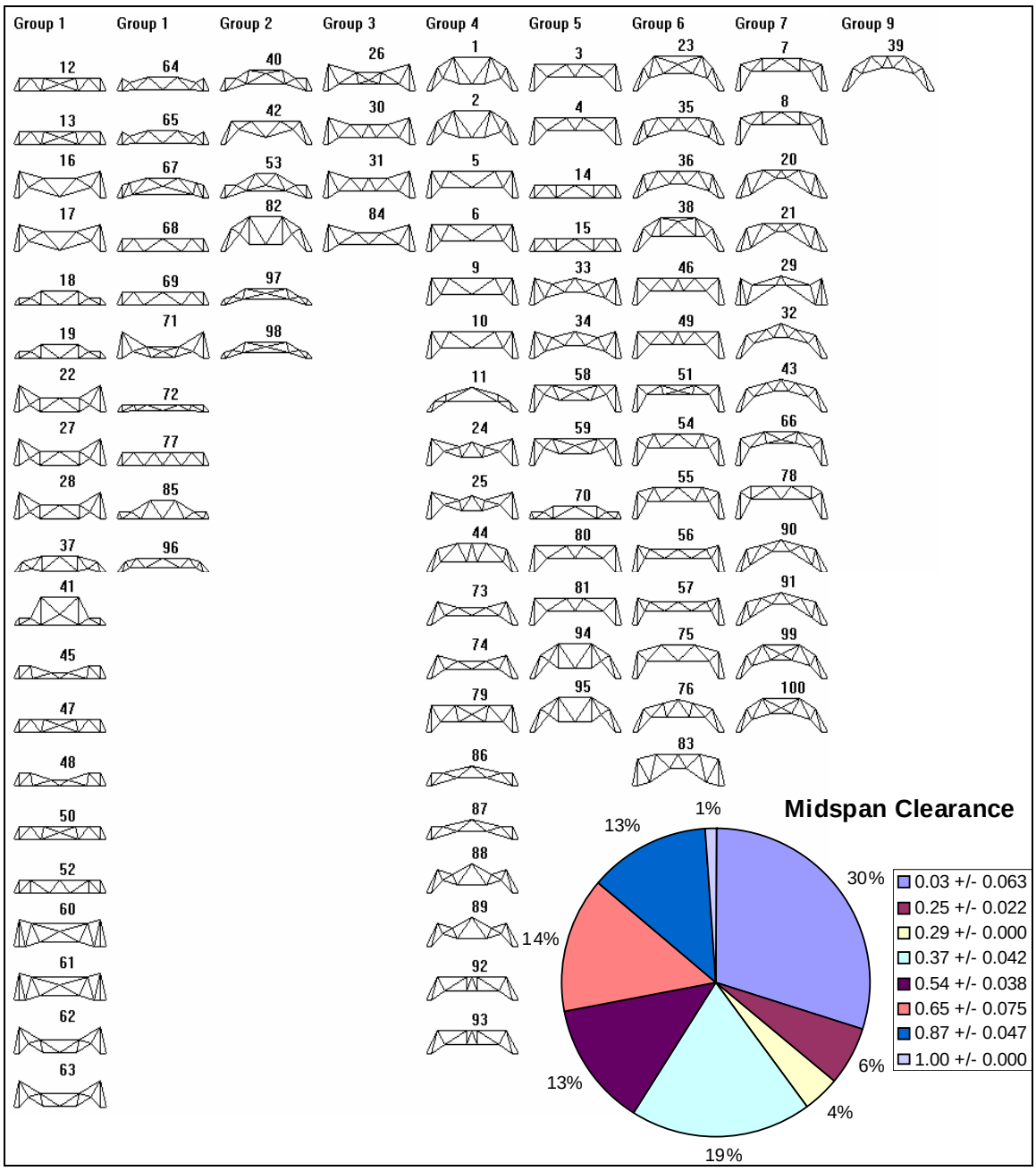


Figure A.27. Feature map created for S100M25 with mid-span clearance as input.

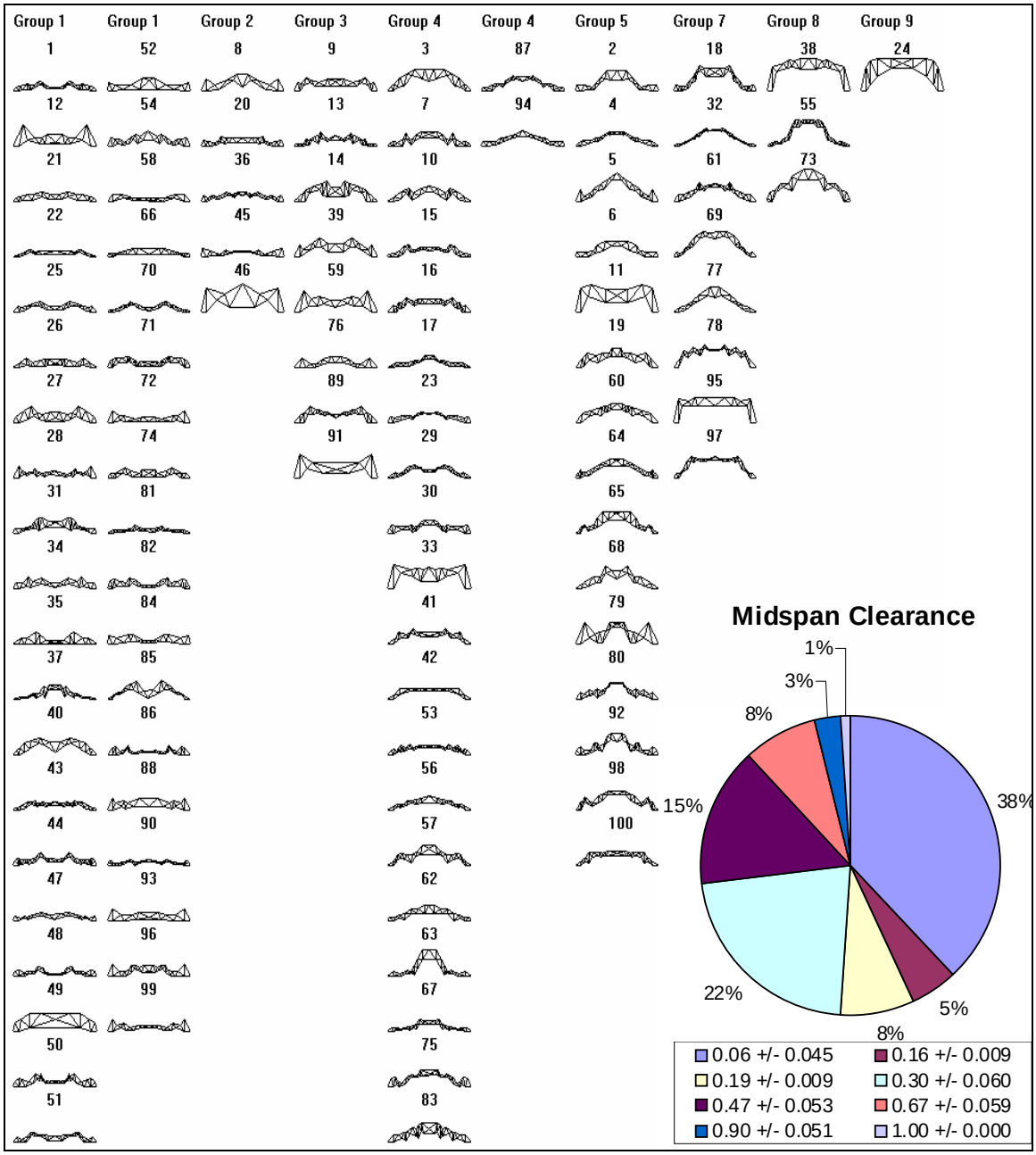


Figure A.28. Feature map created for S100M100 with mid-span clearance as input.

Number of Nodes

Figures A.29 through A.32 show feature maps for populations S25M50,

S100M50, S100M25, and S100M100, respectively, for the number of nodes.

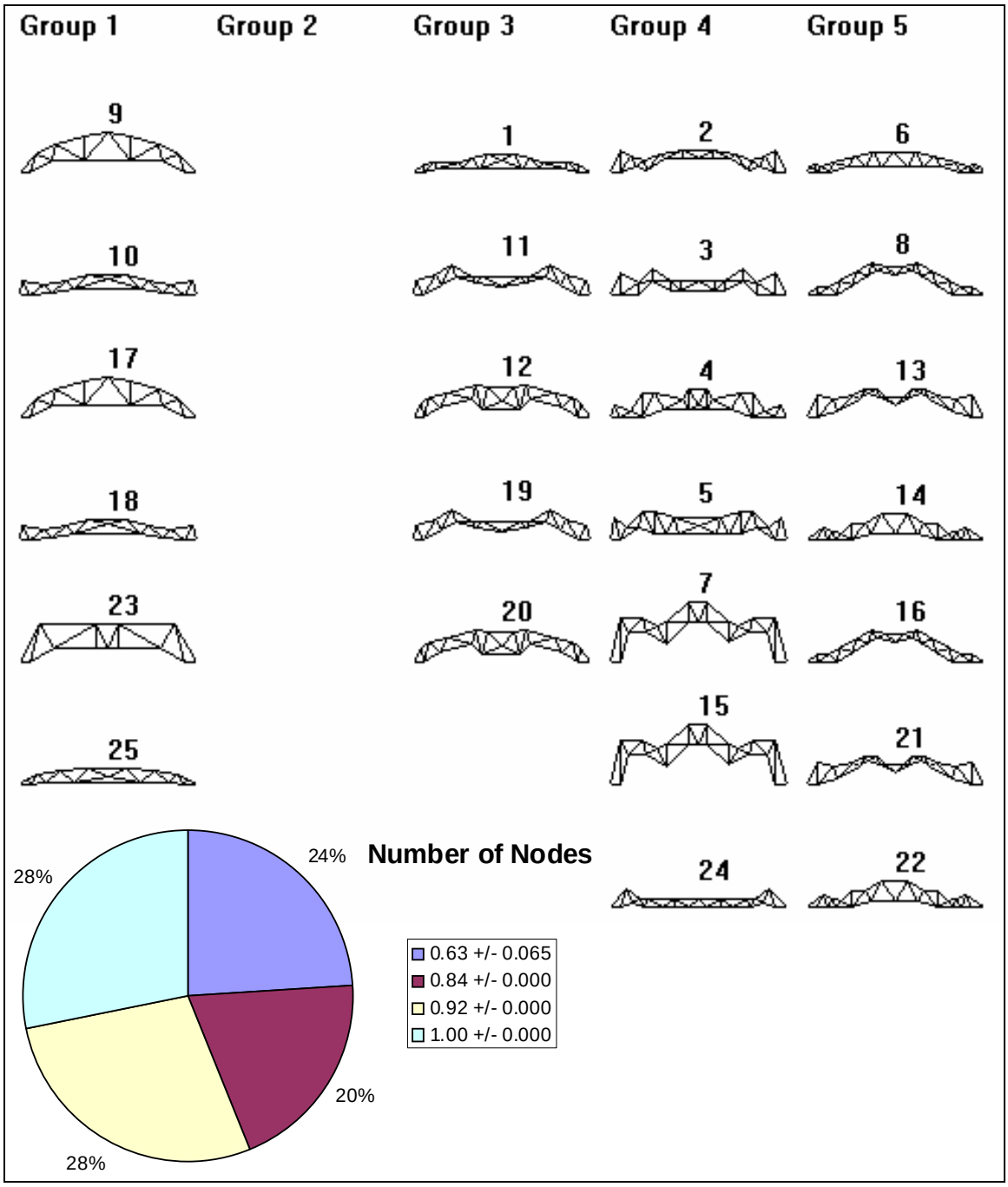


Figure A.29. Feature map created for S25M50 with the number of nodes as input.

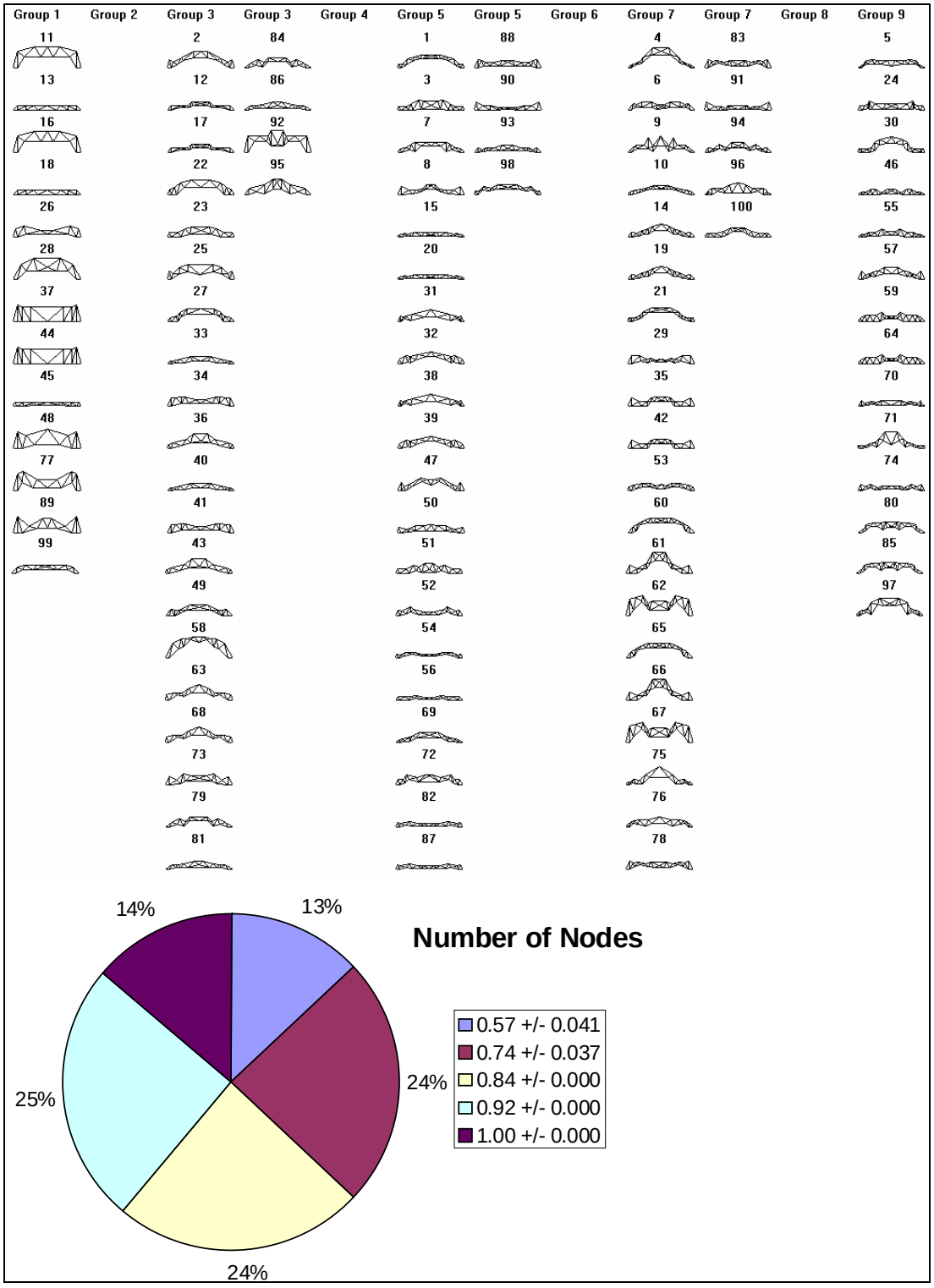


Figure A.30. Feature map created for S100M50 with the number of nodes as input.

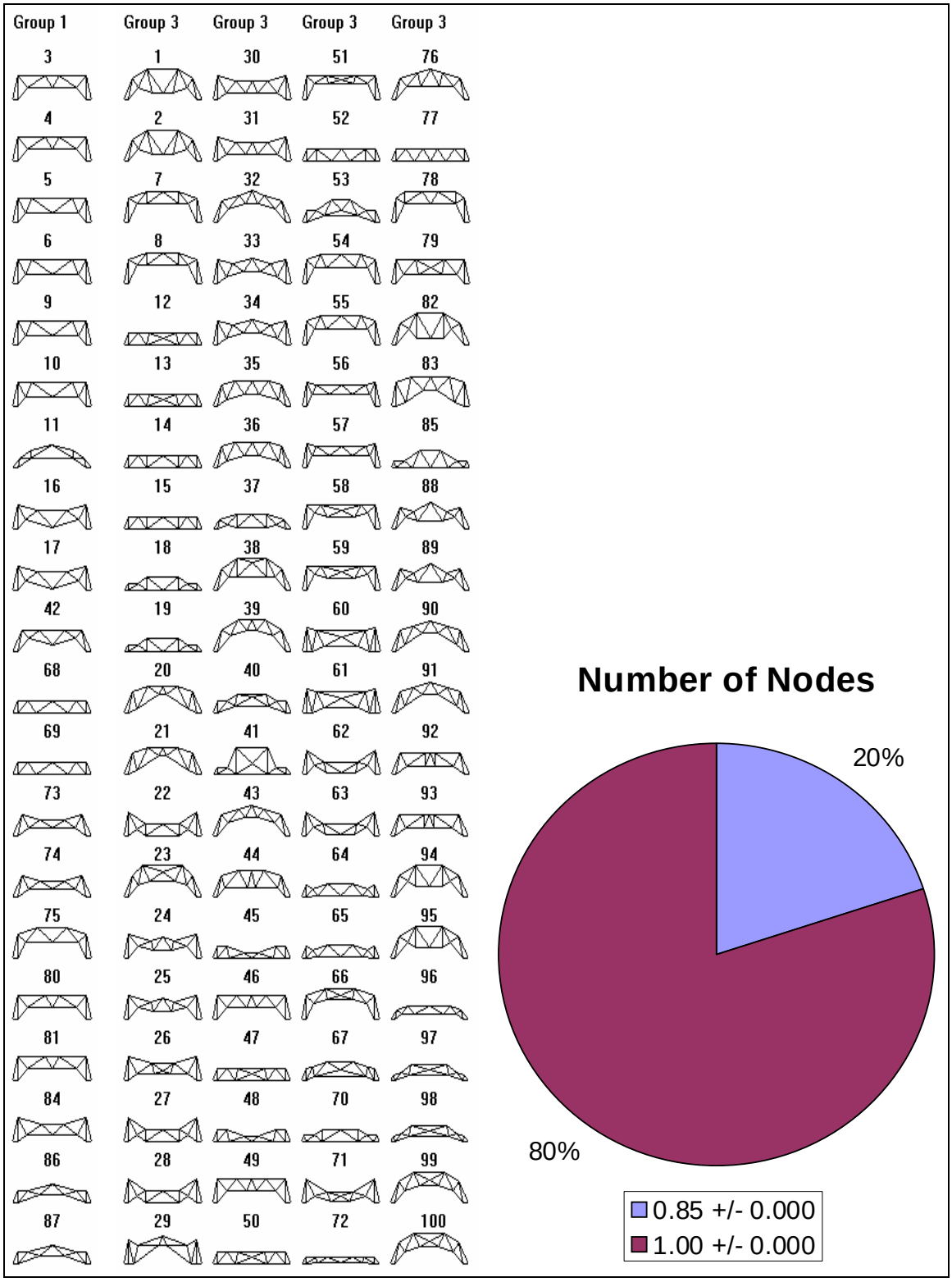


Figure A.31. Feature map created for S100M25 with the number of nodes as input.

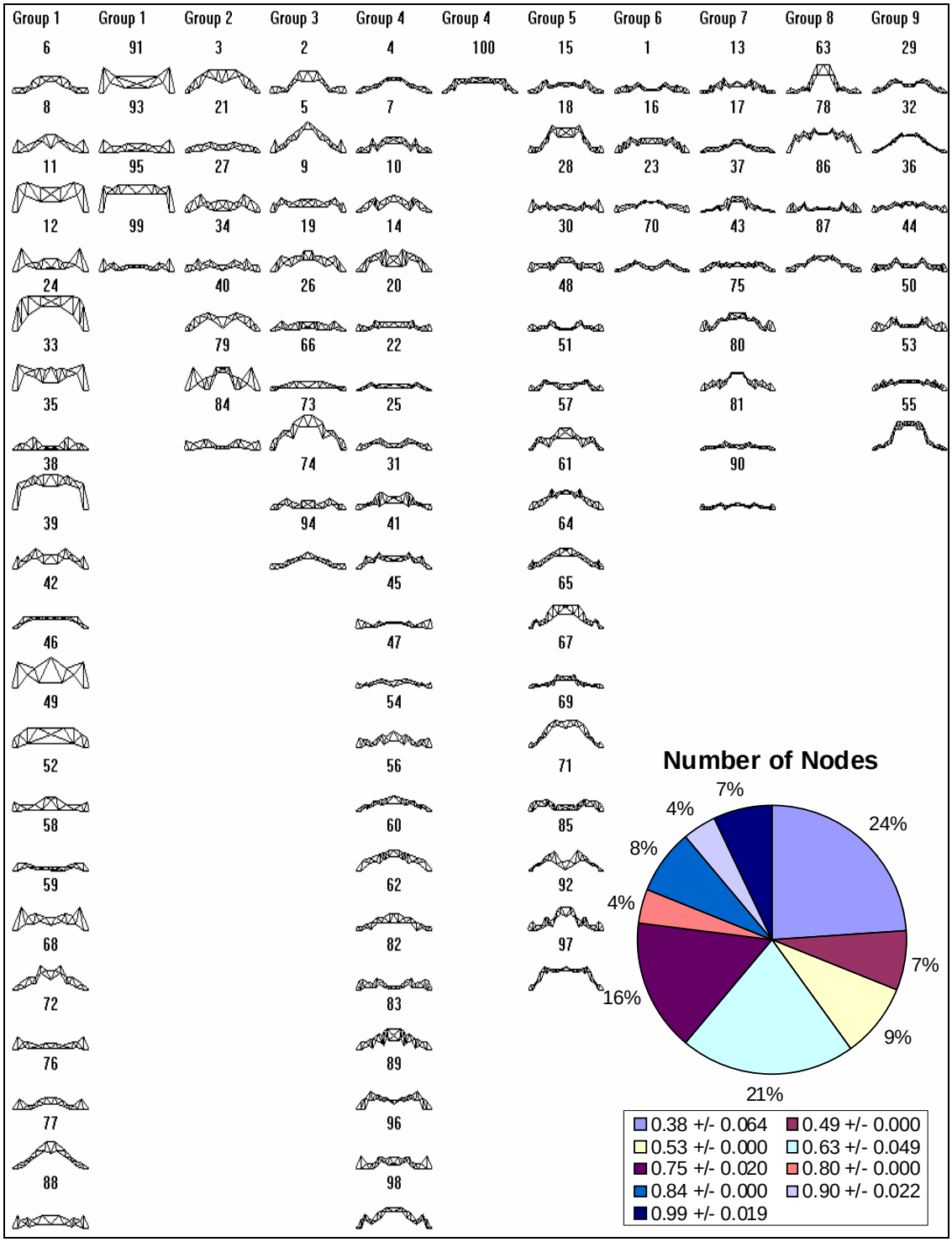


Figure A.32. Feature map created for S100M100 with the number of nodes as input.

Number of Members

Figures A.33 through A.36 show feature maps for populations S25M50, S100M50, S100M25, and S100M100, respectively, for the number of members.

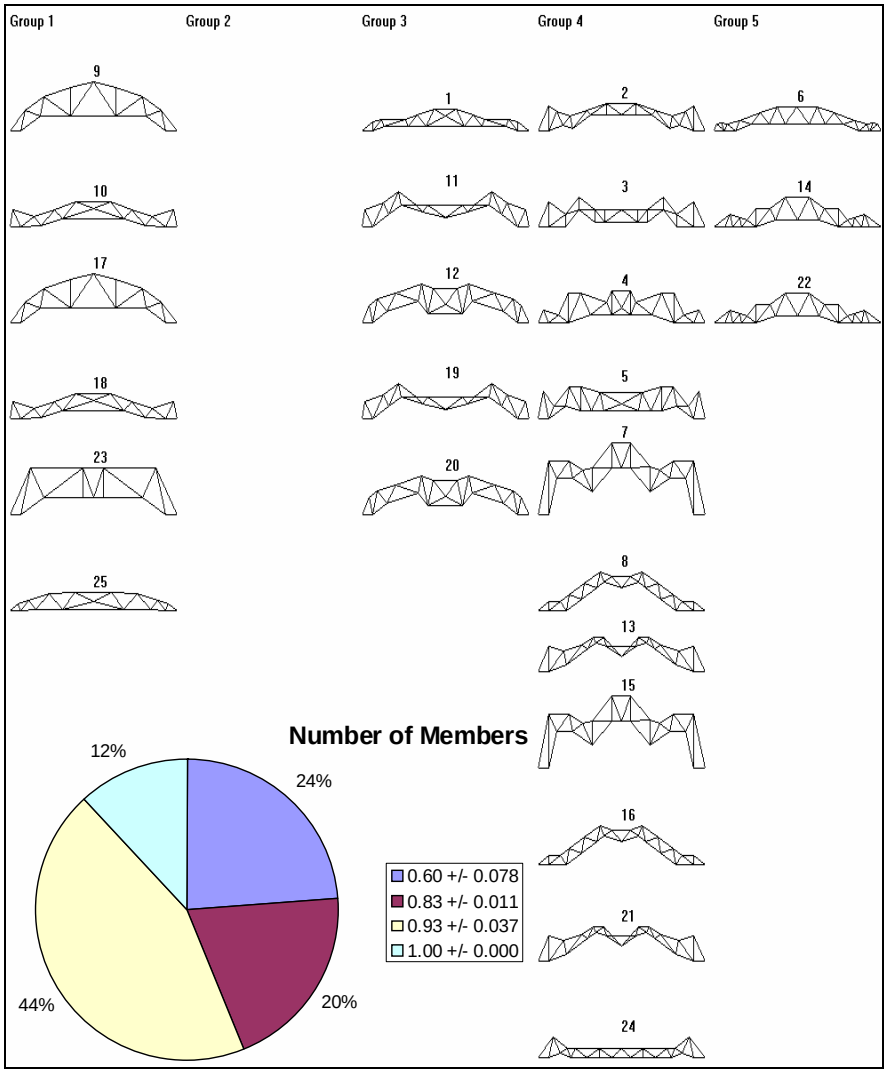


Figure A.33. Feature map created for S25M50 with the number of members as input.

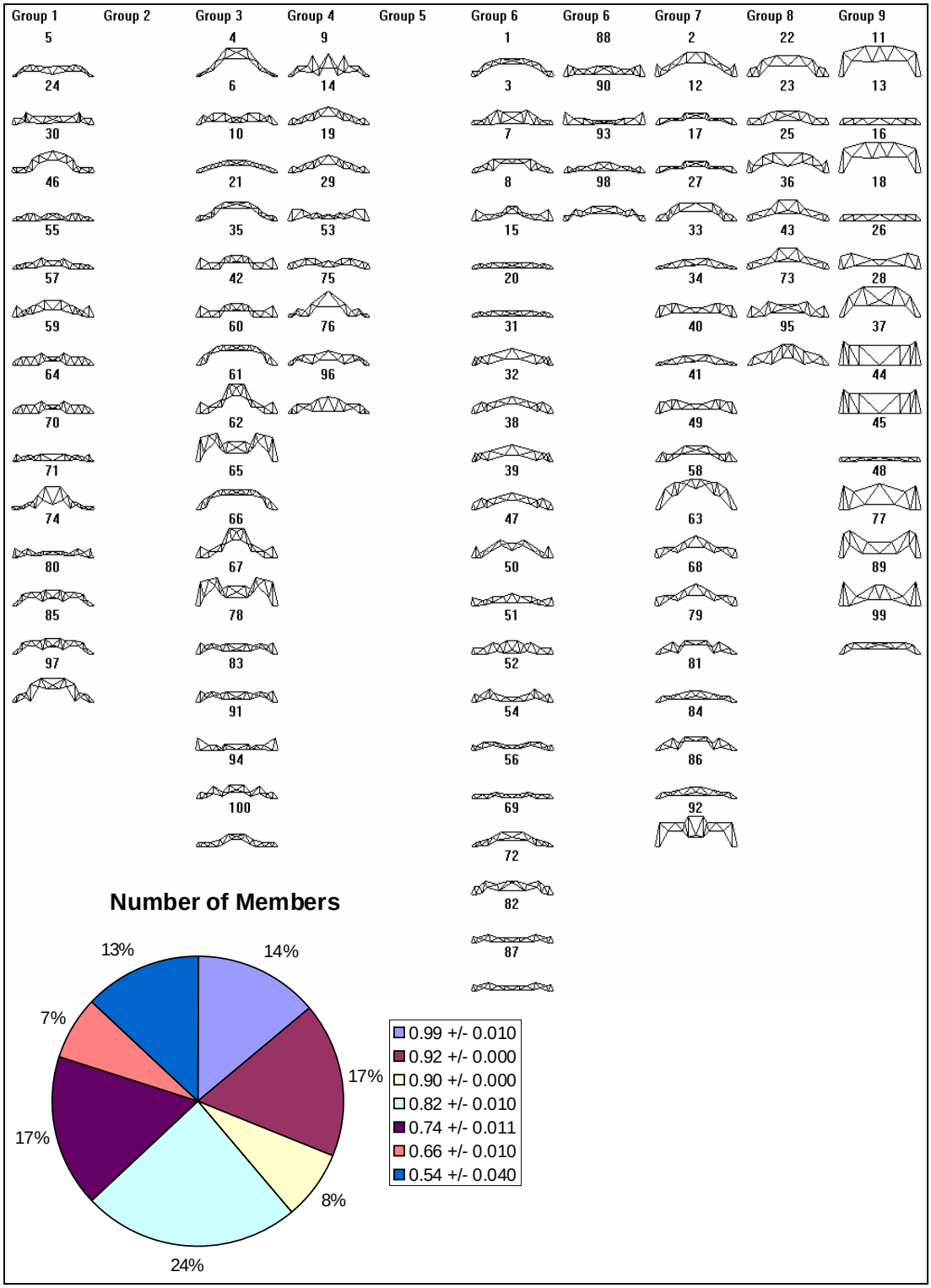


Figure A.34. Feature map created for S100M50 with the number of members as input.

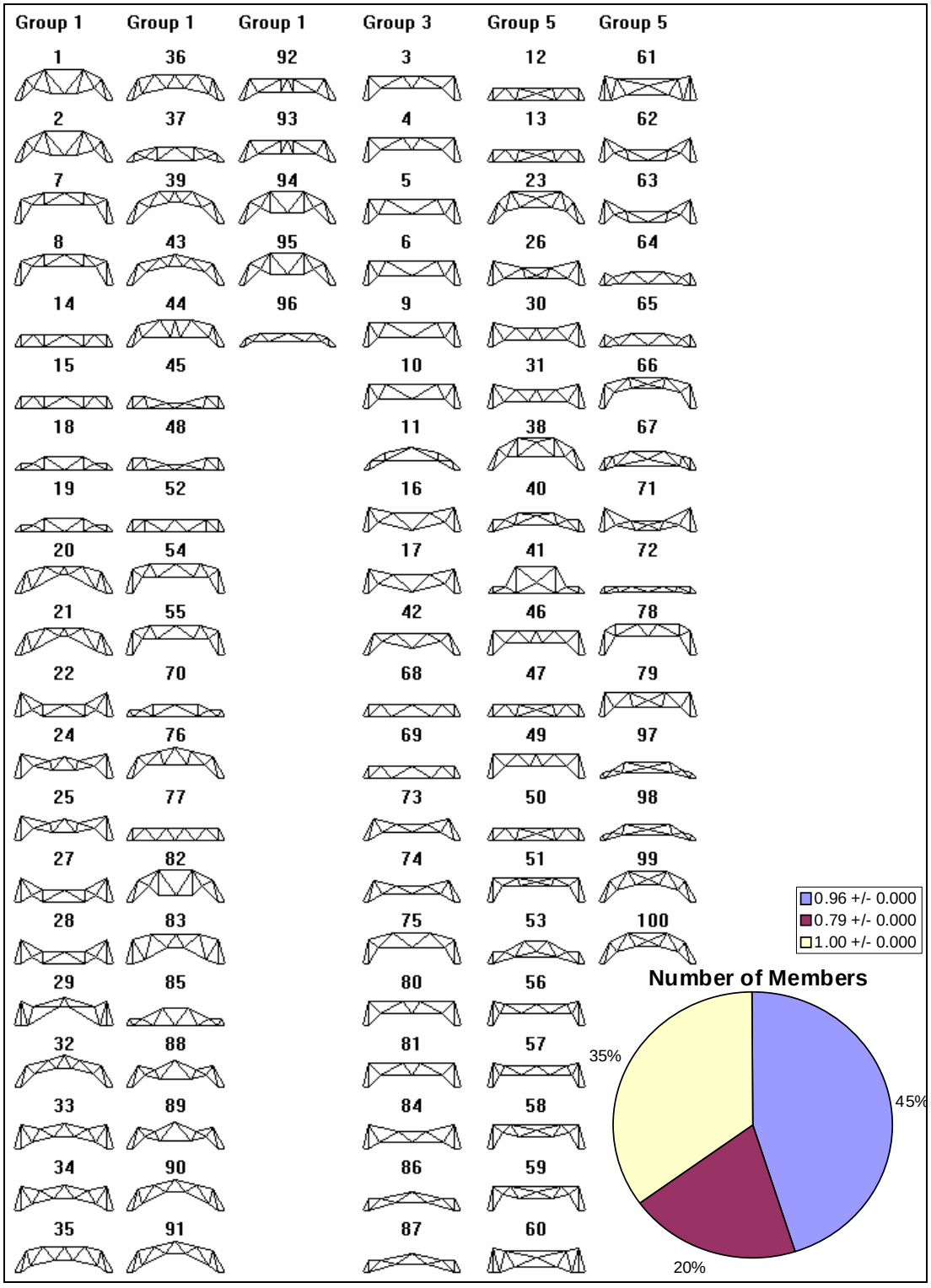


Figure A.35. Feature map created for S100M25 with the number of members as input.

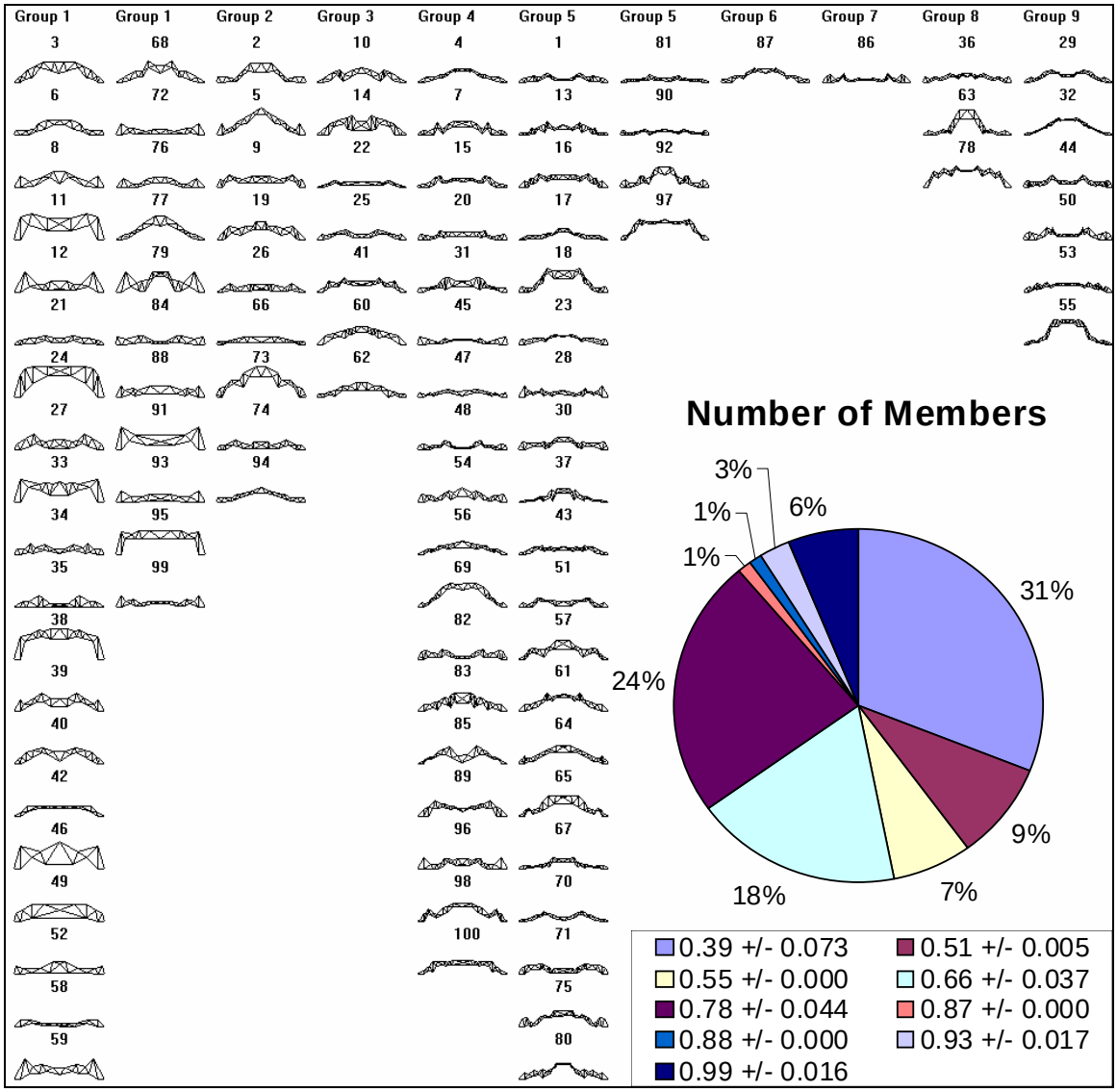


Figure A.36. Feature map created for S100M100 with the number of members as input.

Top Chord Concavity

Figures A.37 through A.40 show feature maps for populations S25M50, S100M50, S100M25, and S100M100, respectively, for top chord concavity.

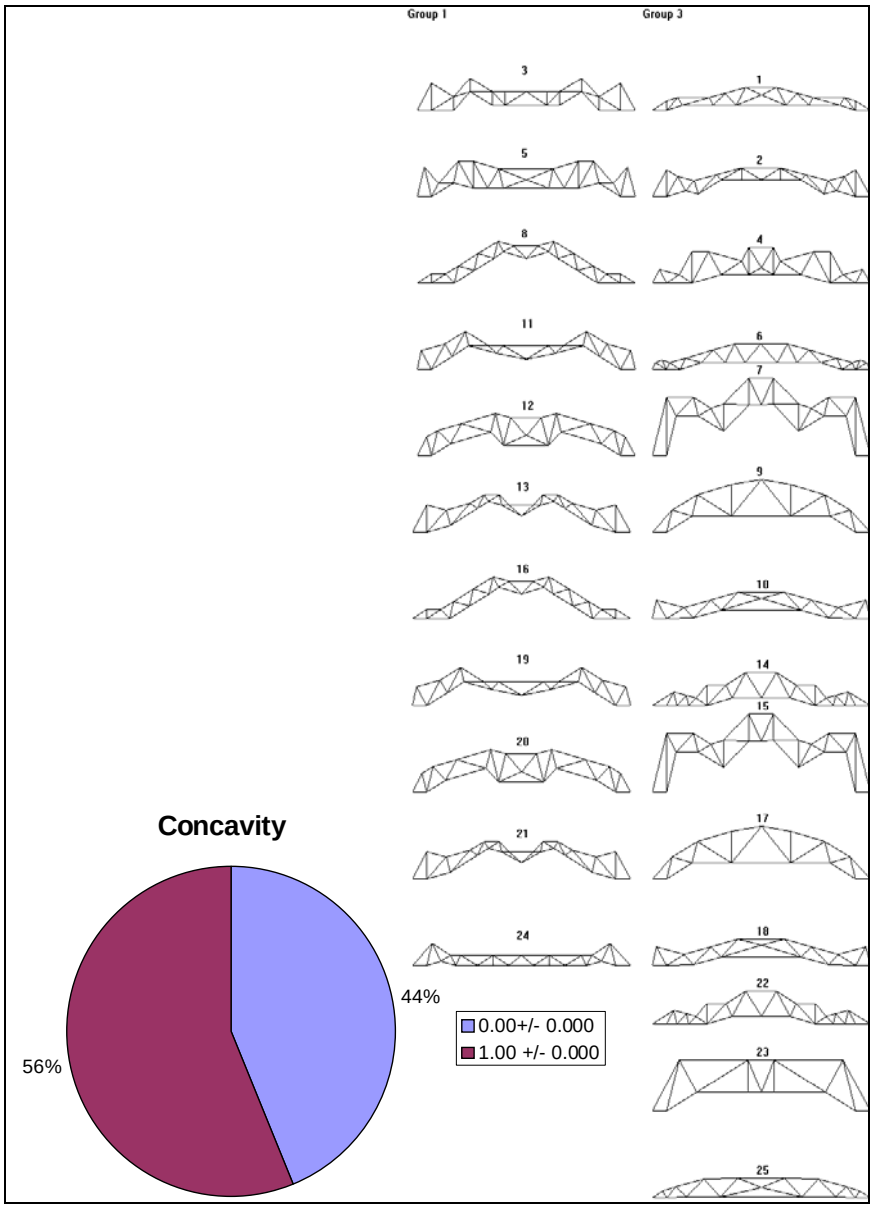


Figure A.37. Feature map created for S25M50 with top chord concavity as input.

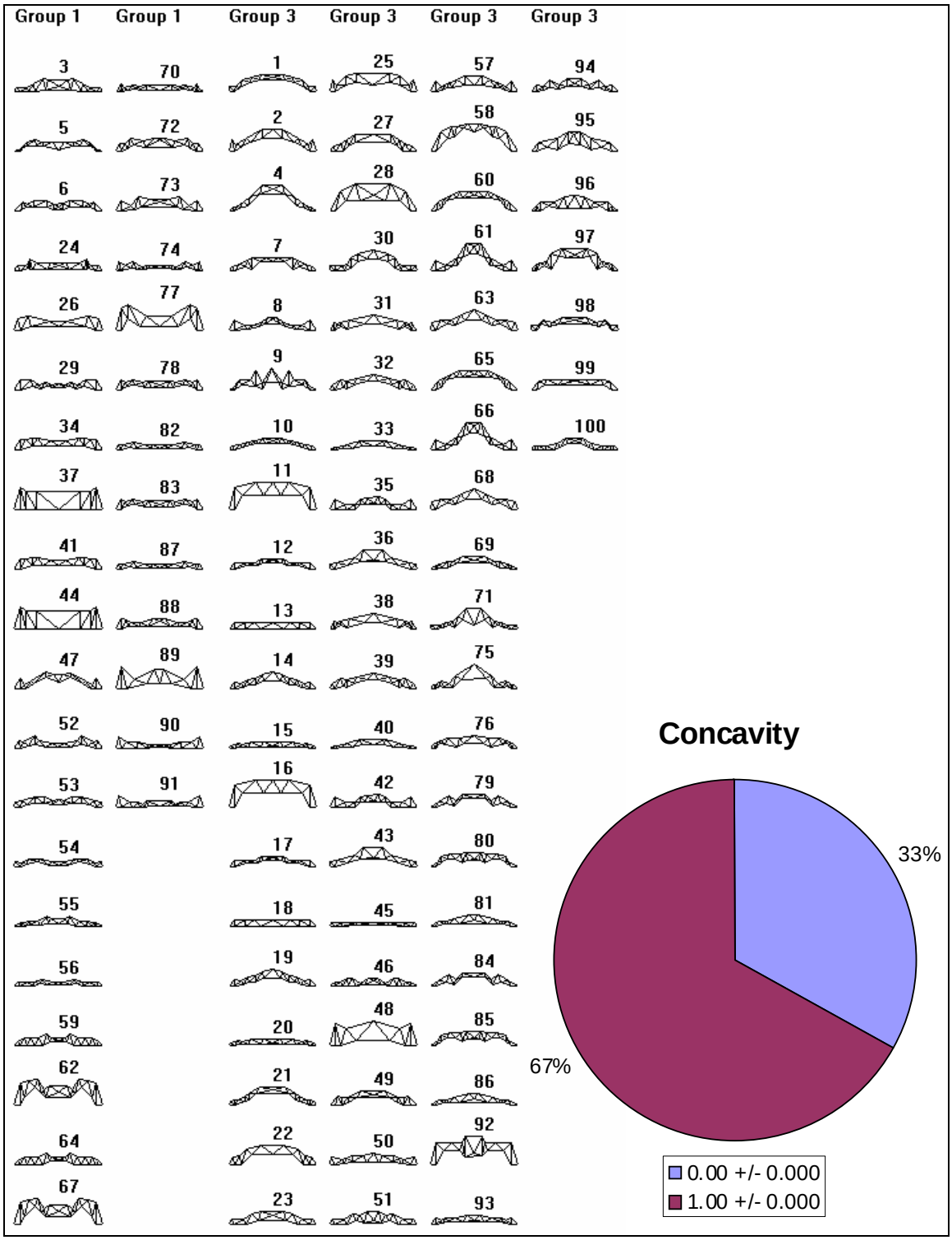


Figure A.38. Feature map created for S100M50 with top chord concavity as input.

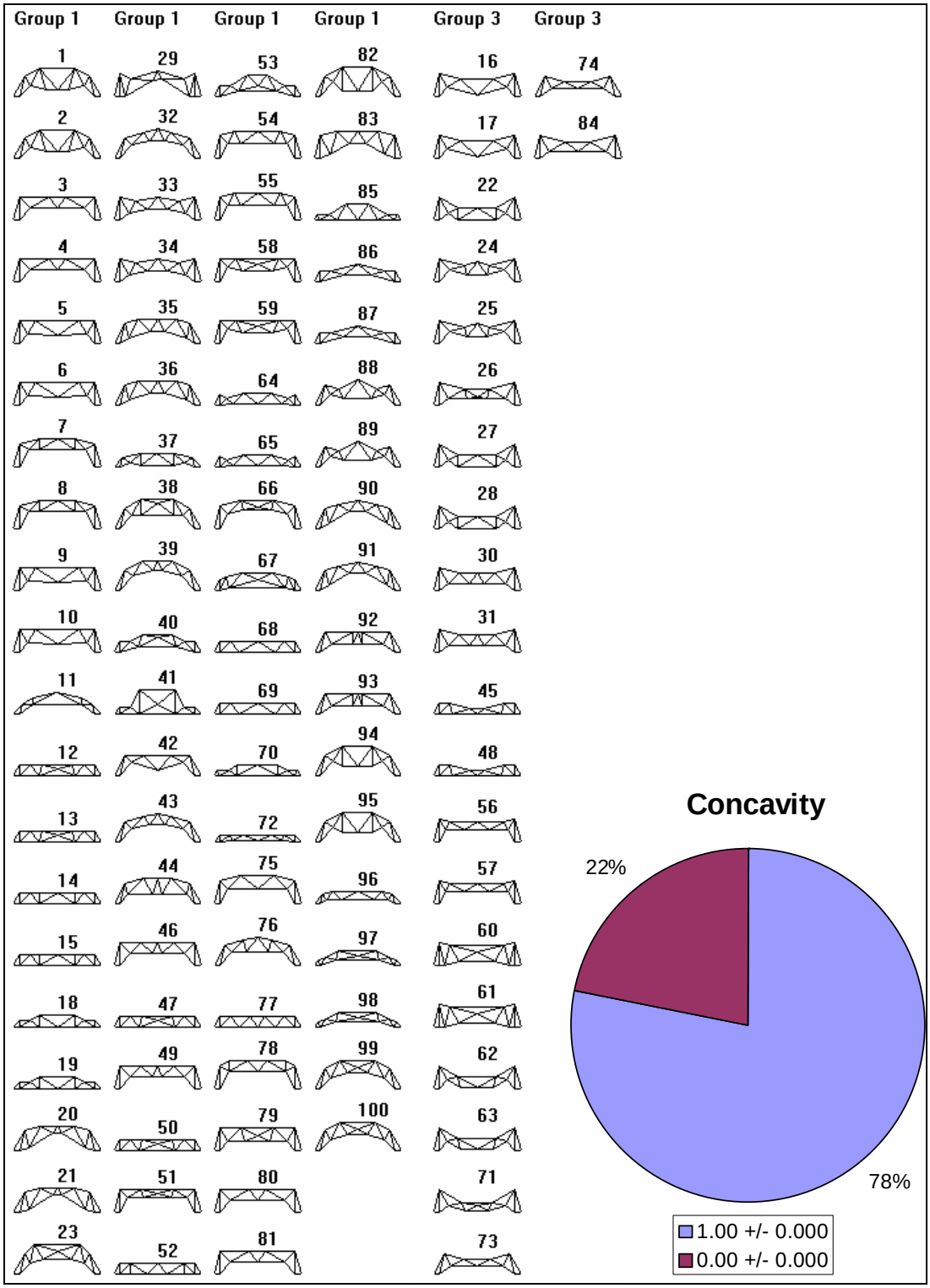


Figure A.39. Feature map created for S100M25 with top chord concavity as input.

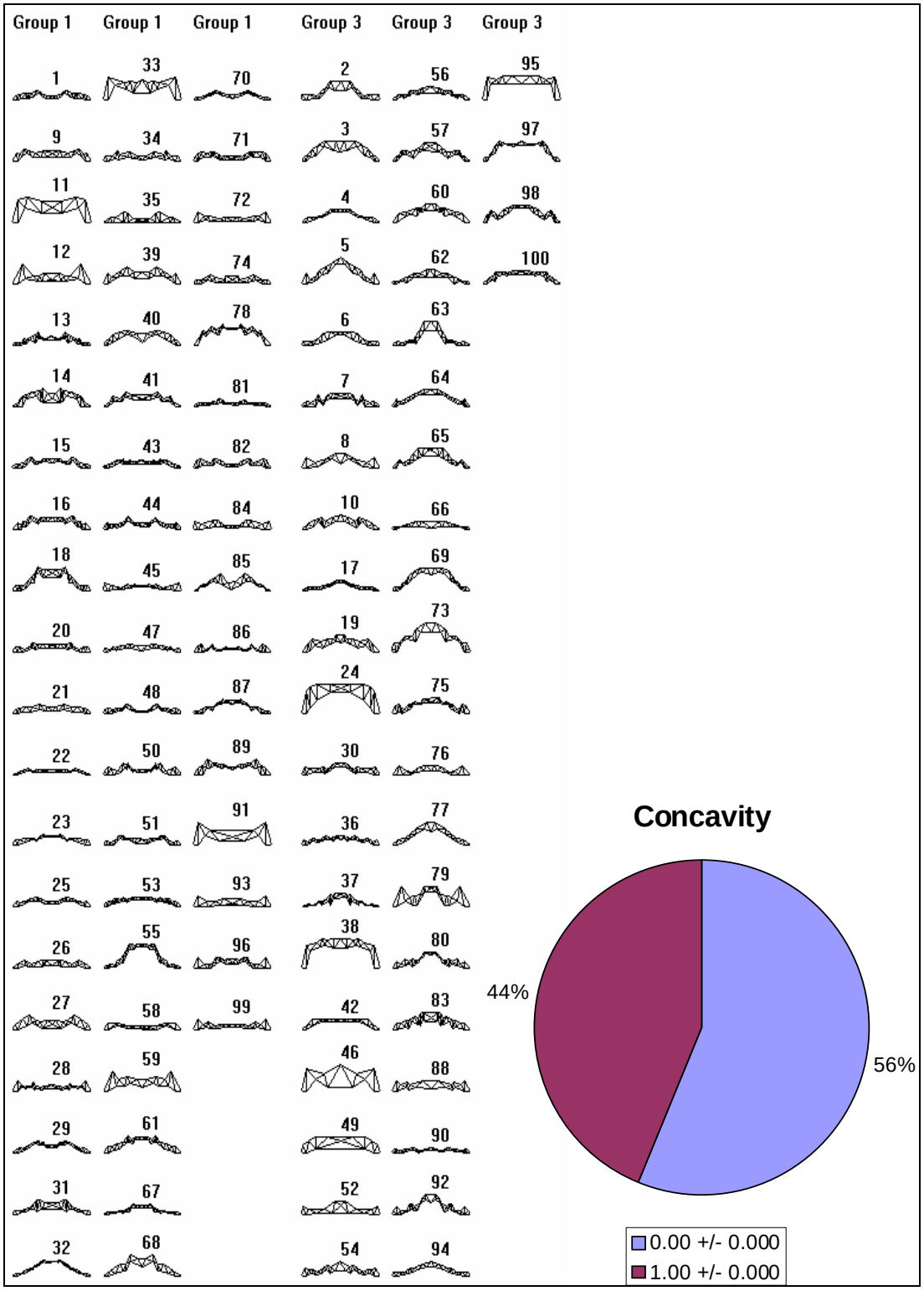


Figure A.40. Feature map created for S100M100 with top chord concavity as input.

Top Chord Dip

Figures A.41 through A.44 show feature maps for populations S25M50, S100M50, S100M25, and S100M100, respectively, for the presence of top chord dips.

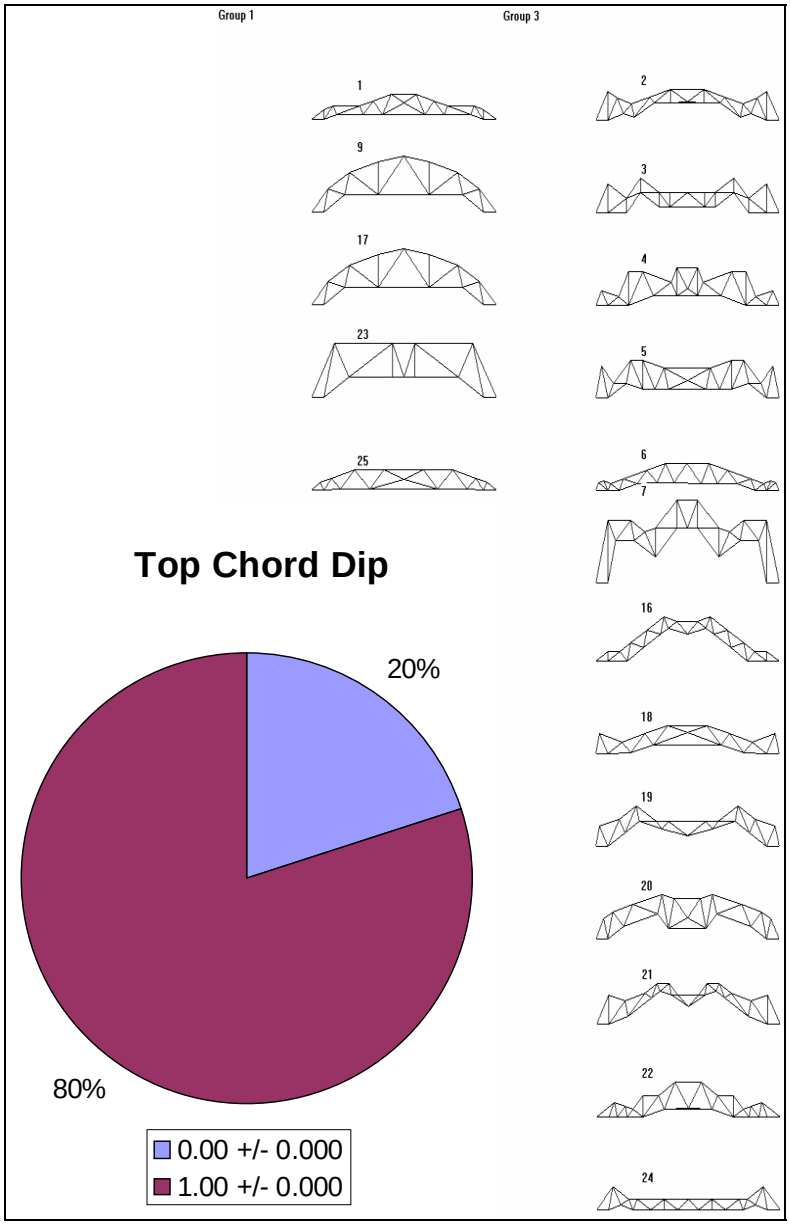


Figure A.41. Feature map created for S25M50 with the presence of a top chord dip as input.

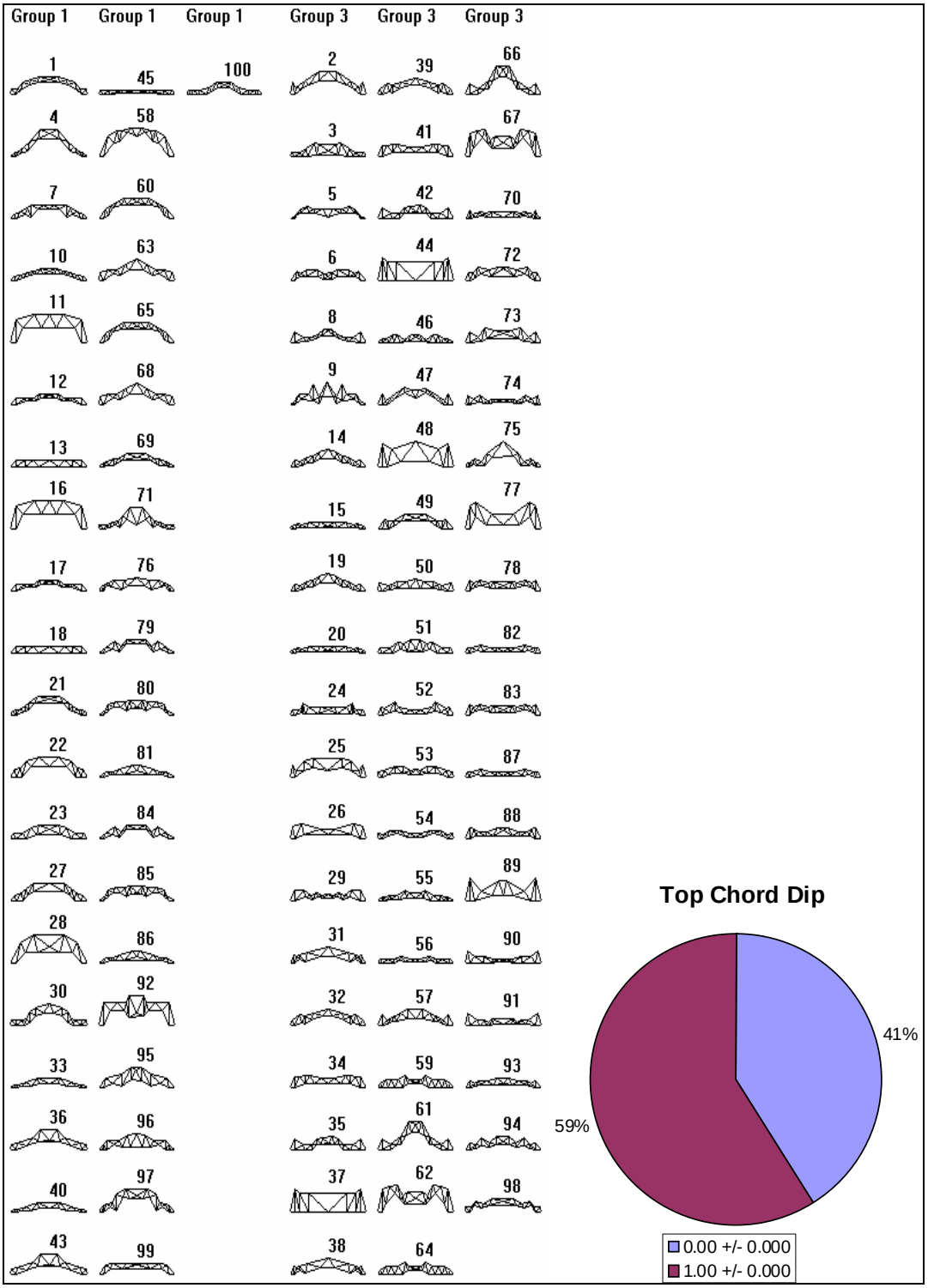


Figure A.42. Feature map created for S100M50 with the presence of a top chord dip as input.

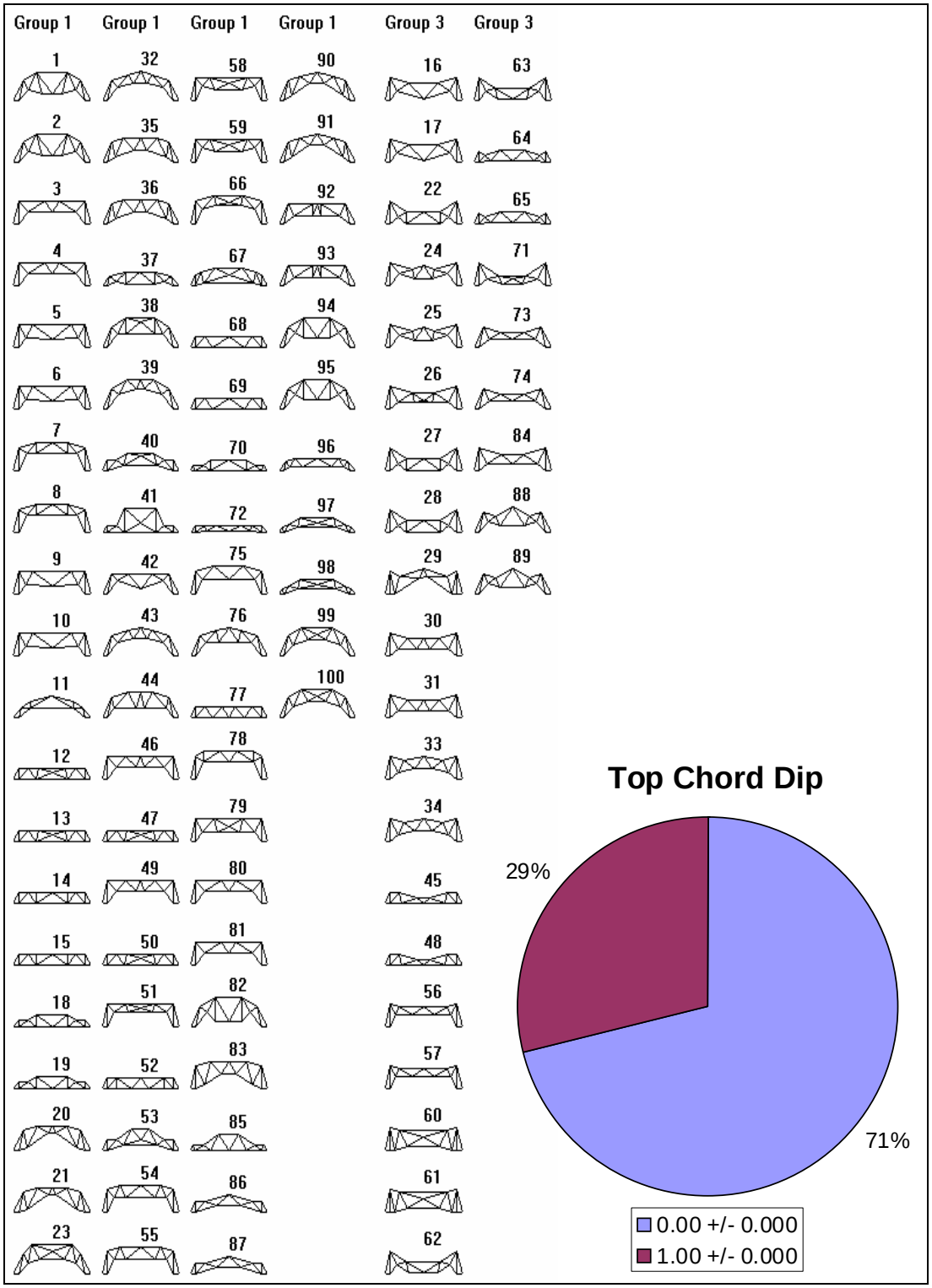


Figure A.43. Feature map created for S100M25 with the presence of a top chord dip as input.

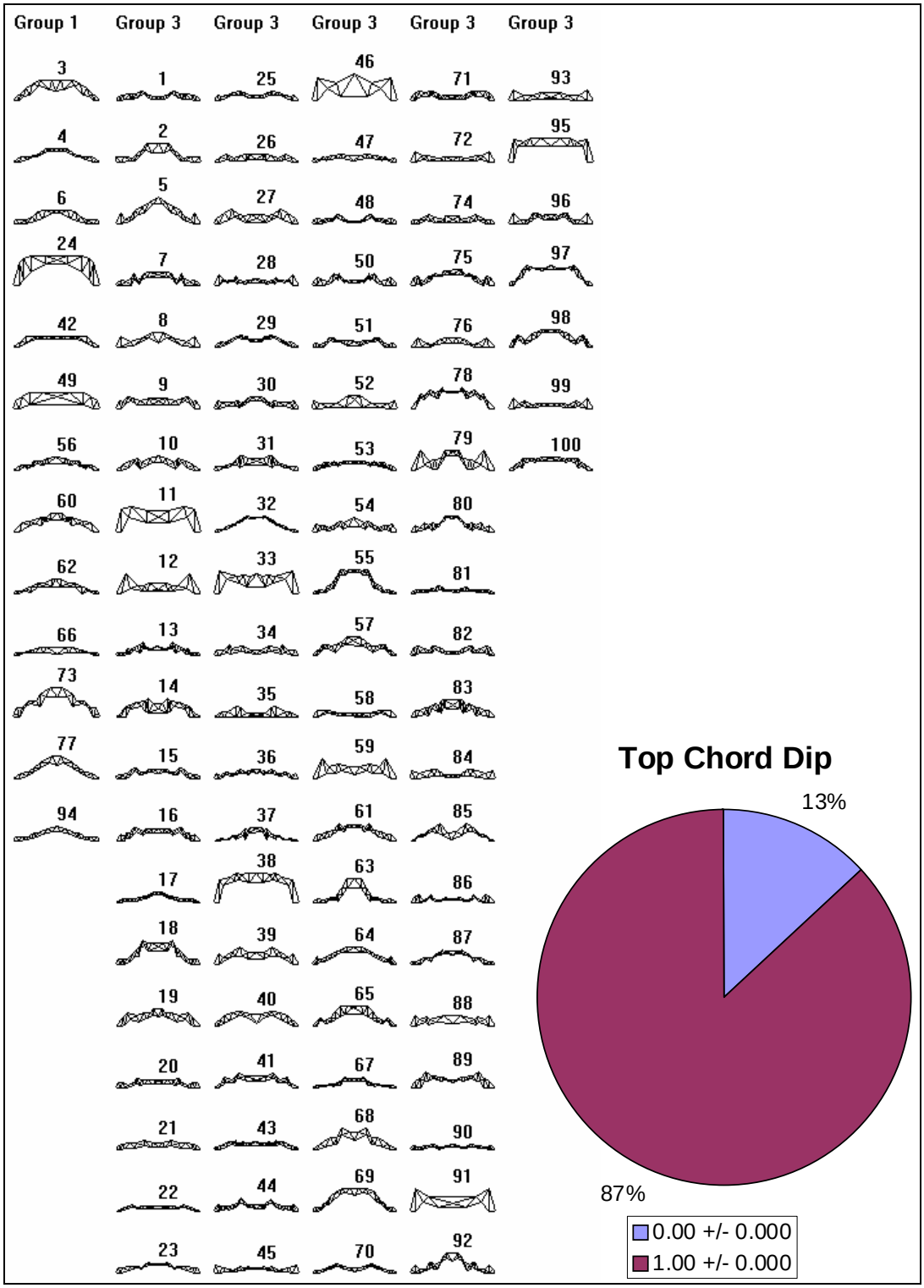


Figure A.44. Feature map created for S100M100 with the presence of a top chord dip as input.

Top Chord Flatness

Figures A.45 through A.48 show feature maps for populations S25M35, S25M50, S50M25, and S50M35, respectively, for the second measure of top chord flatness.

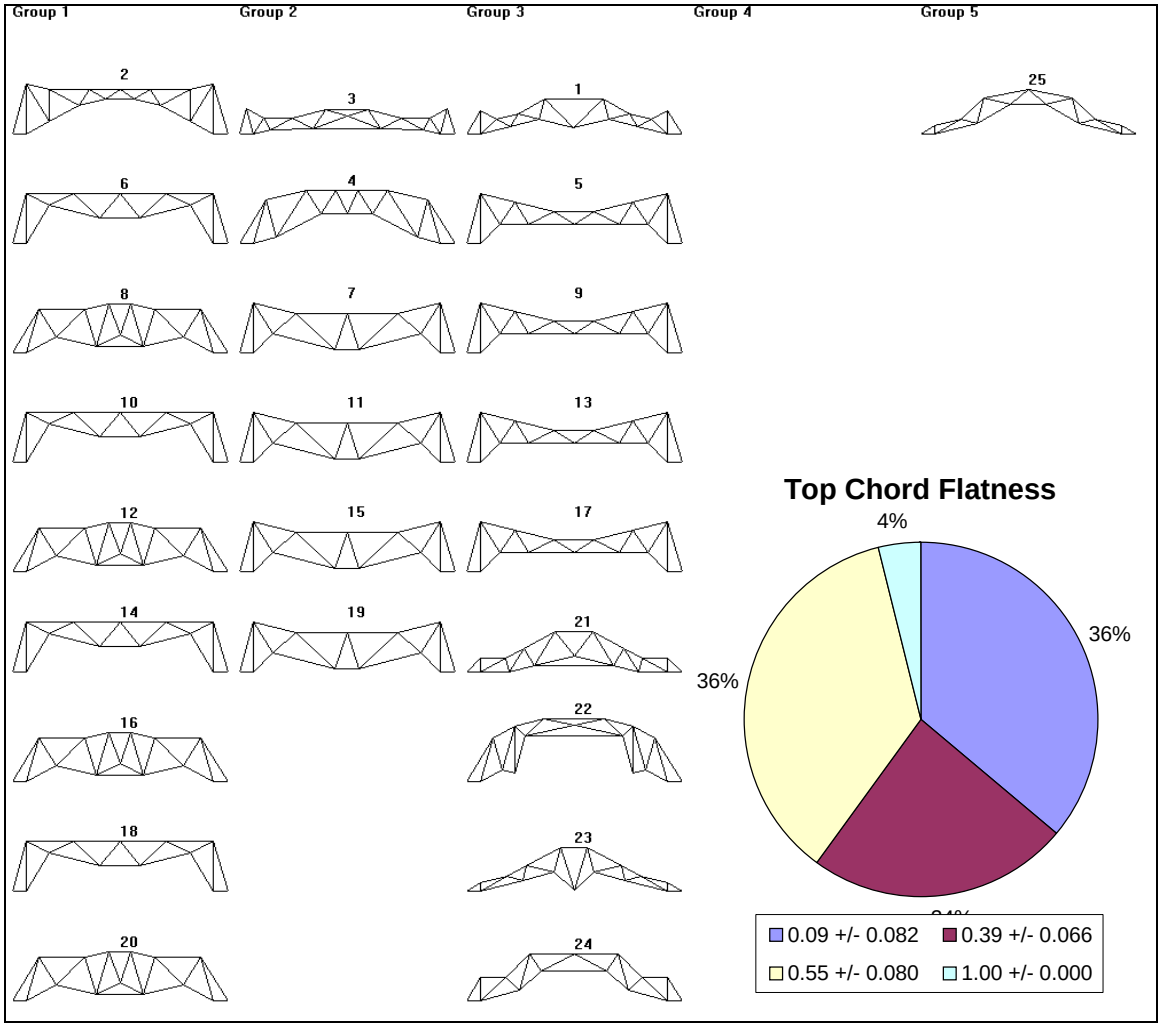


Figure A.45. Feature map created for S25M35 with top chord flatness as input.

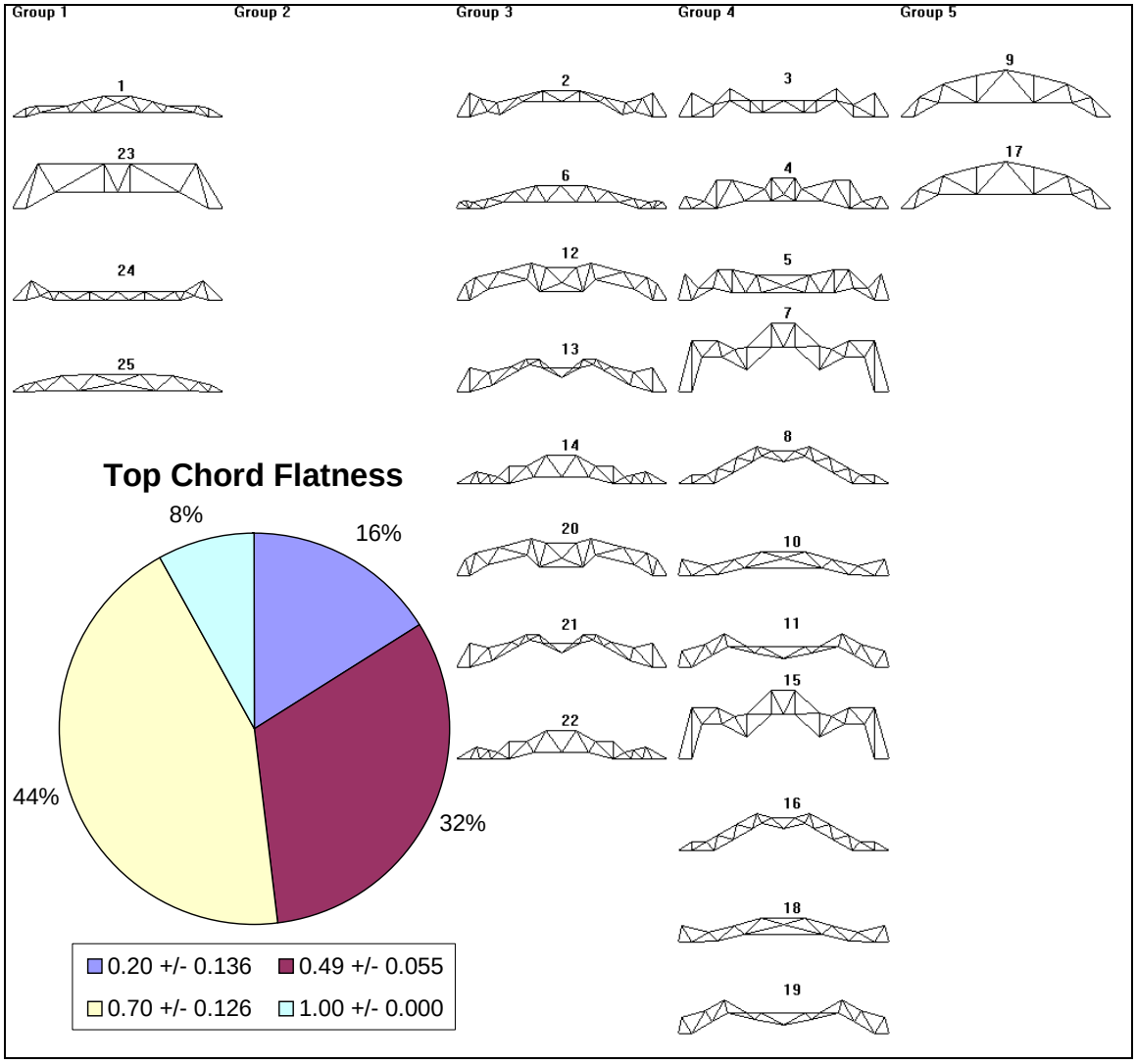


Figure A.46. Feature map created for S25M50 with top chord flatness as input.

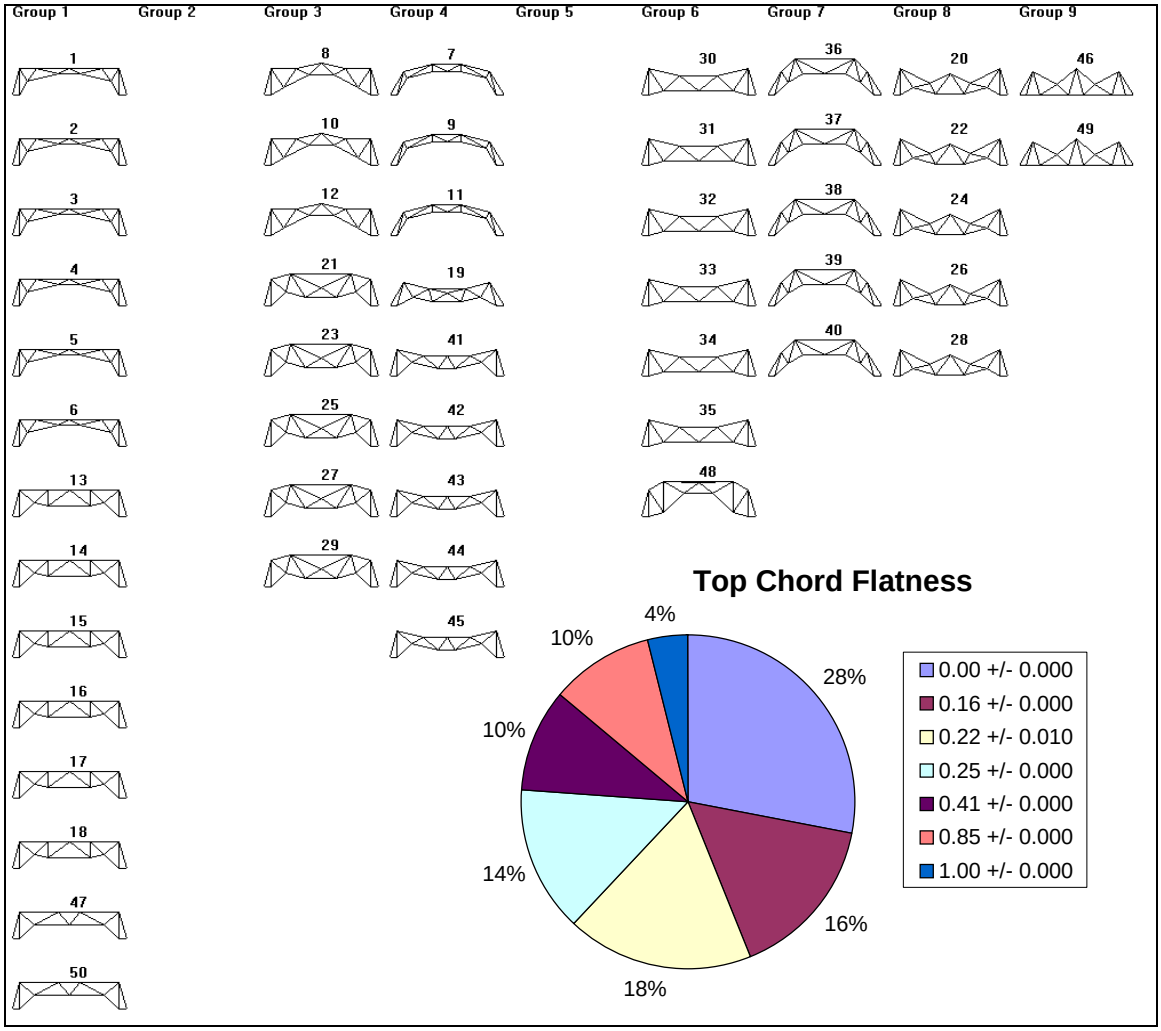


Figure A.47. Feature map created for S50M25 with top chord flatness as input.

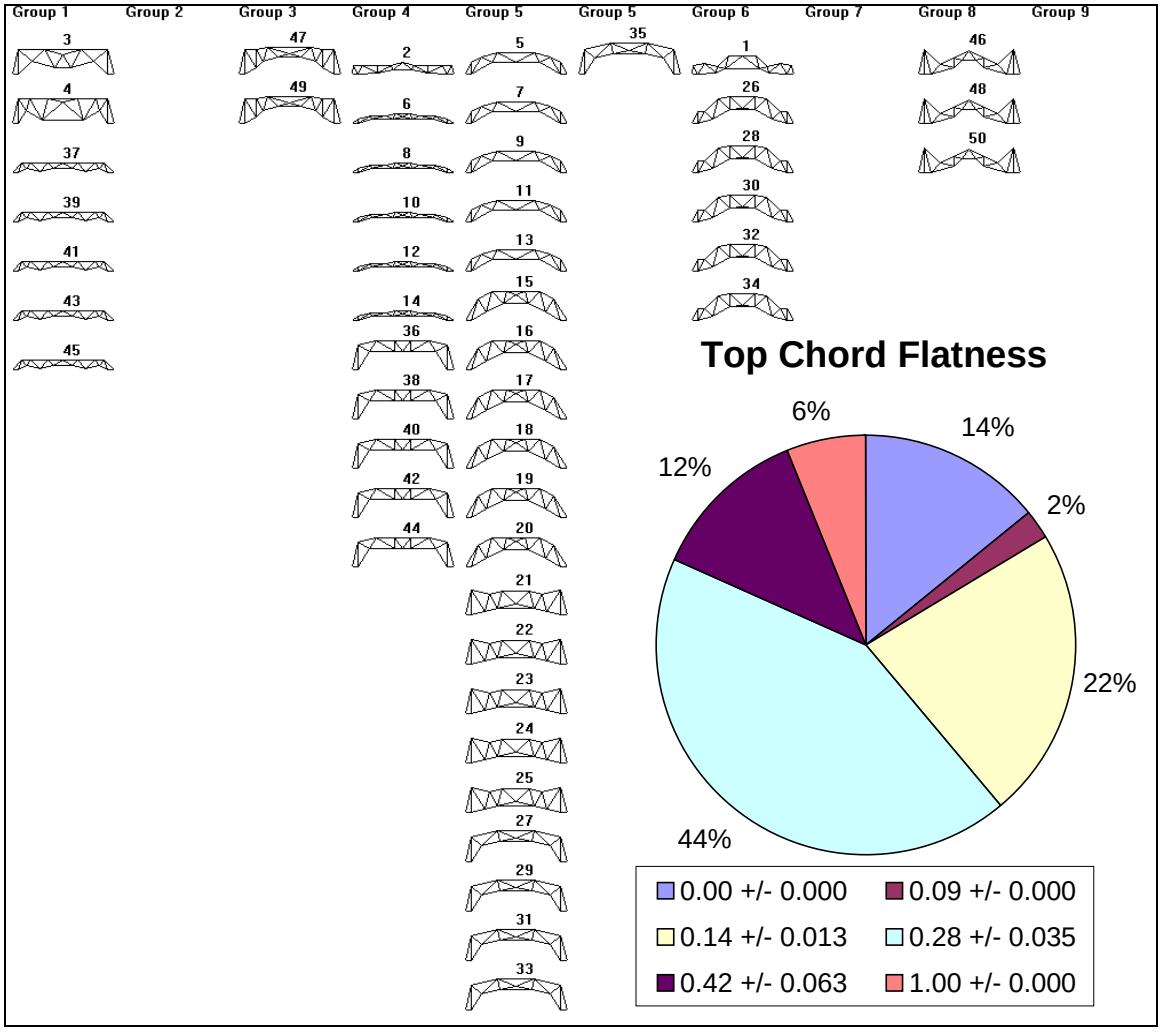


Figure A.48. Feature map created for S50M35 with top chord flatness as input.

Bottom Chord Nodes

Figures A.49 through A.52 show feature maps for populations S25M35, S25M50, S50M25, and S50M35, respectively, for the number of bottom chord nodes.

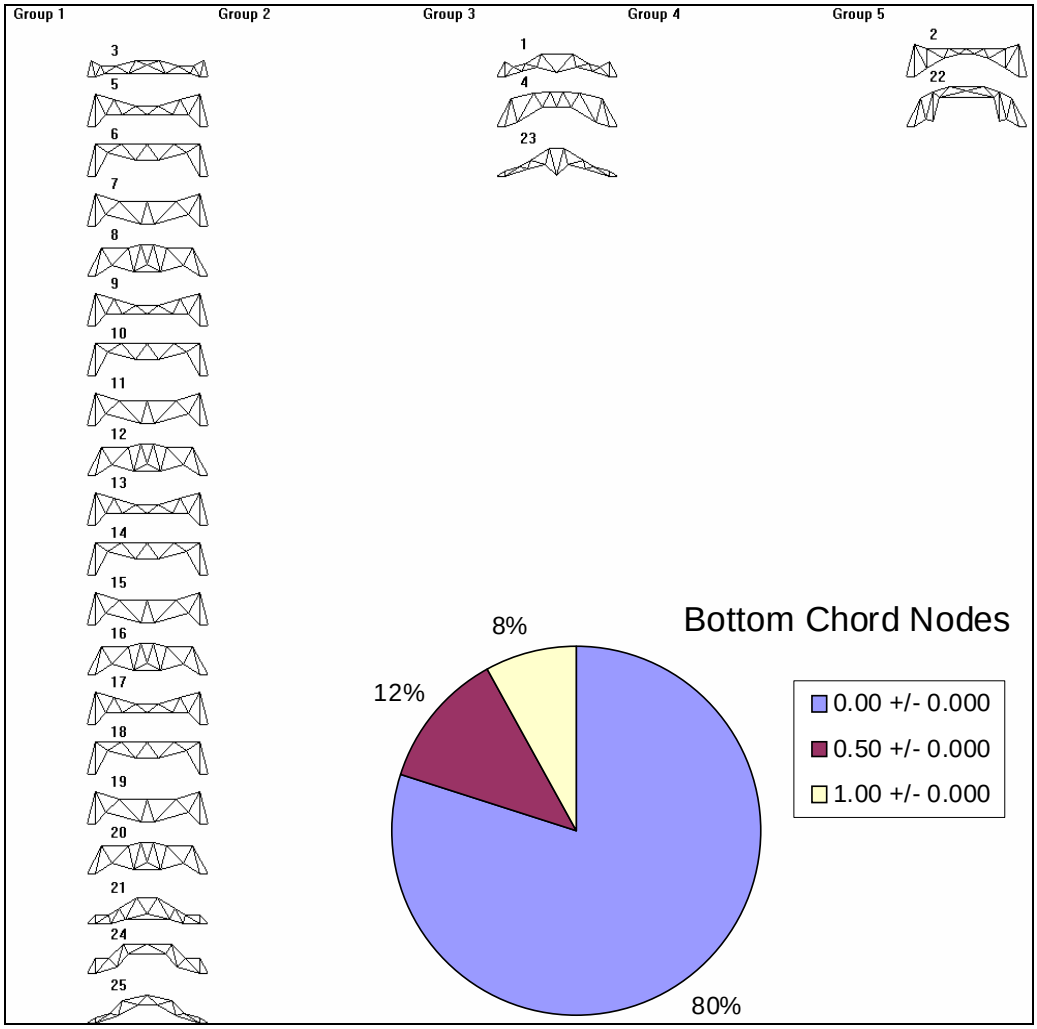


Figure A.49. Feature map created for S25M35 with number of bottom chord nodes as input.

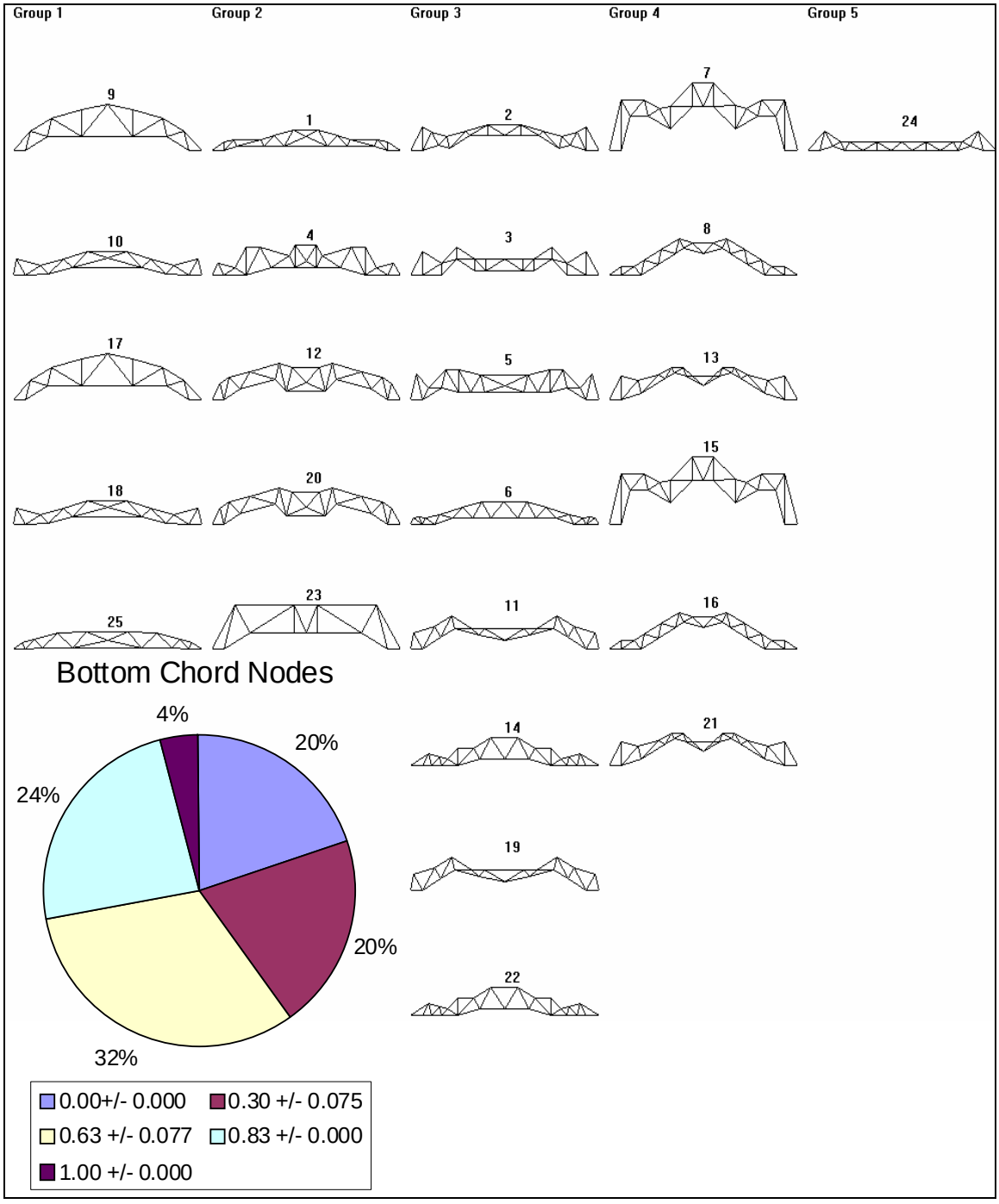


Figure A.50. Feature map created for S25M50 with number of bottom chord nodes as input.

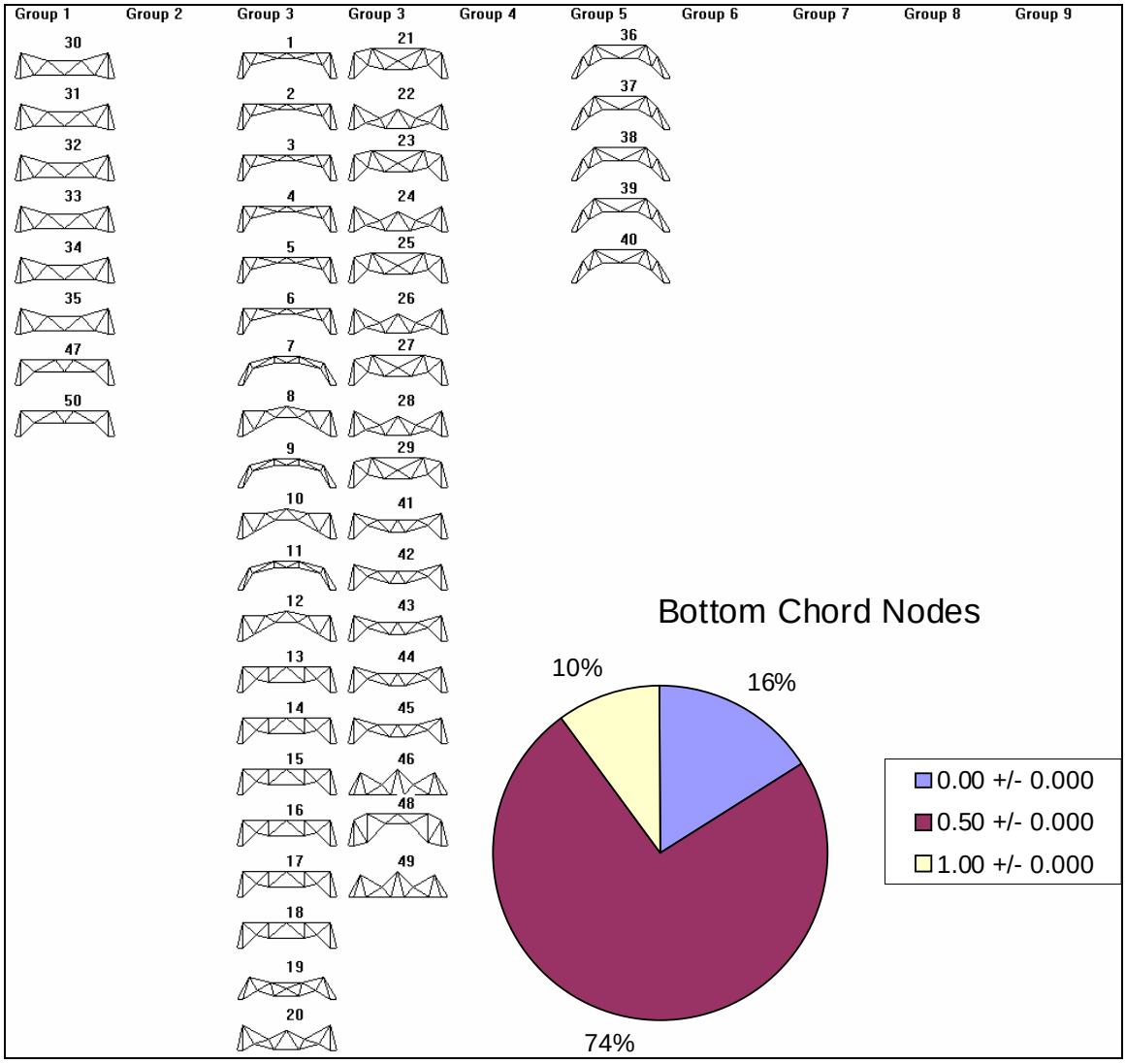


Figure A.51. Feature map created for S50M25 with number of bottom chord nodes as input.

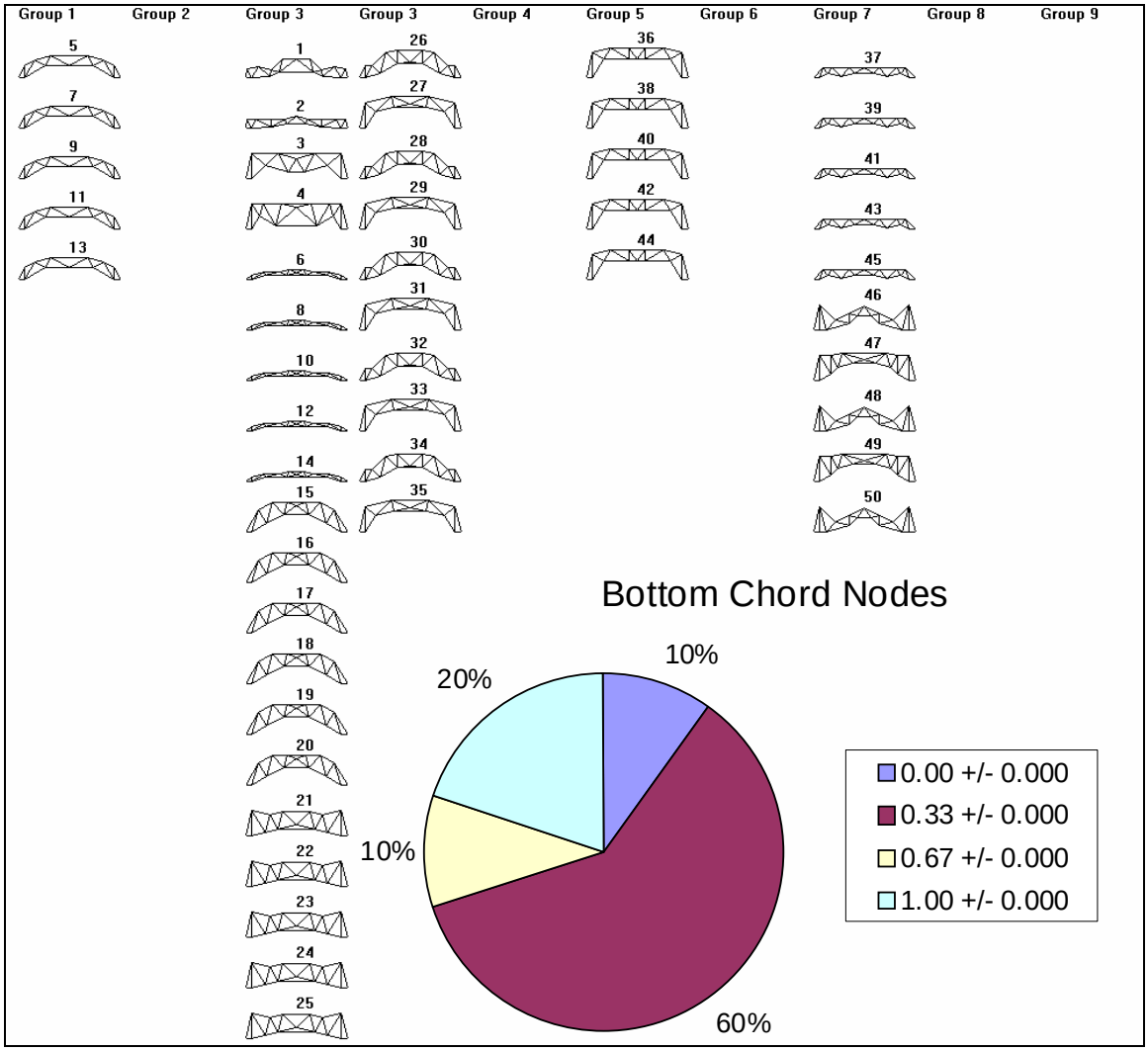


Figure A.52. Feature map created for S50M35 with number of bottom chord nodes as input.

Ratio of Top to Bottom Chord Nodes

Figures A.53 through A.56 show feature maps for populations S25M35, S25M50, S50M25, and S50M35, respectively, for the ratio of top to bottom chord nodes.

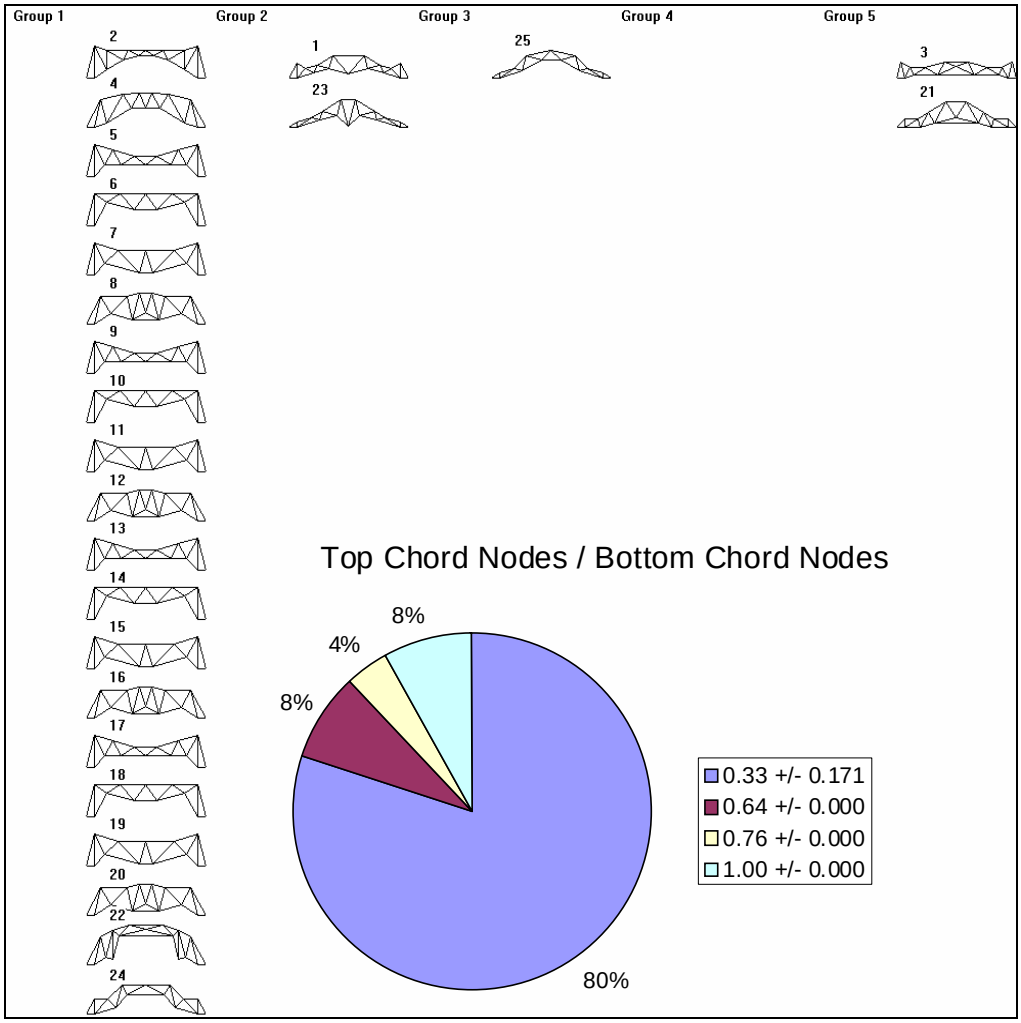


Figure A.53. Feature map created for S25M35 with ratio of top to bottom chord nodes as input.

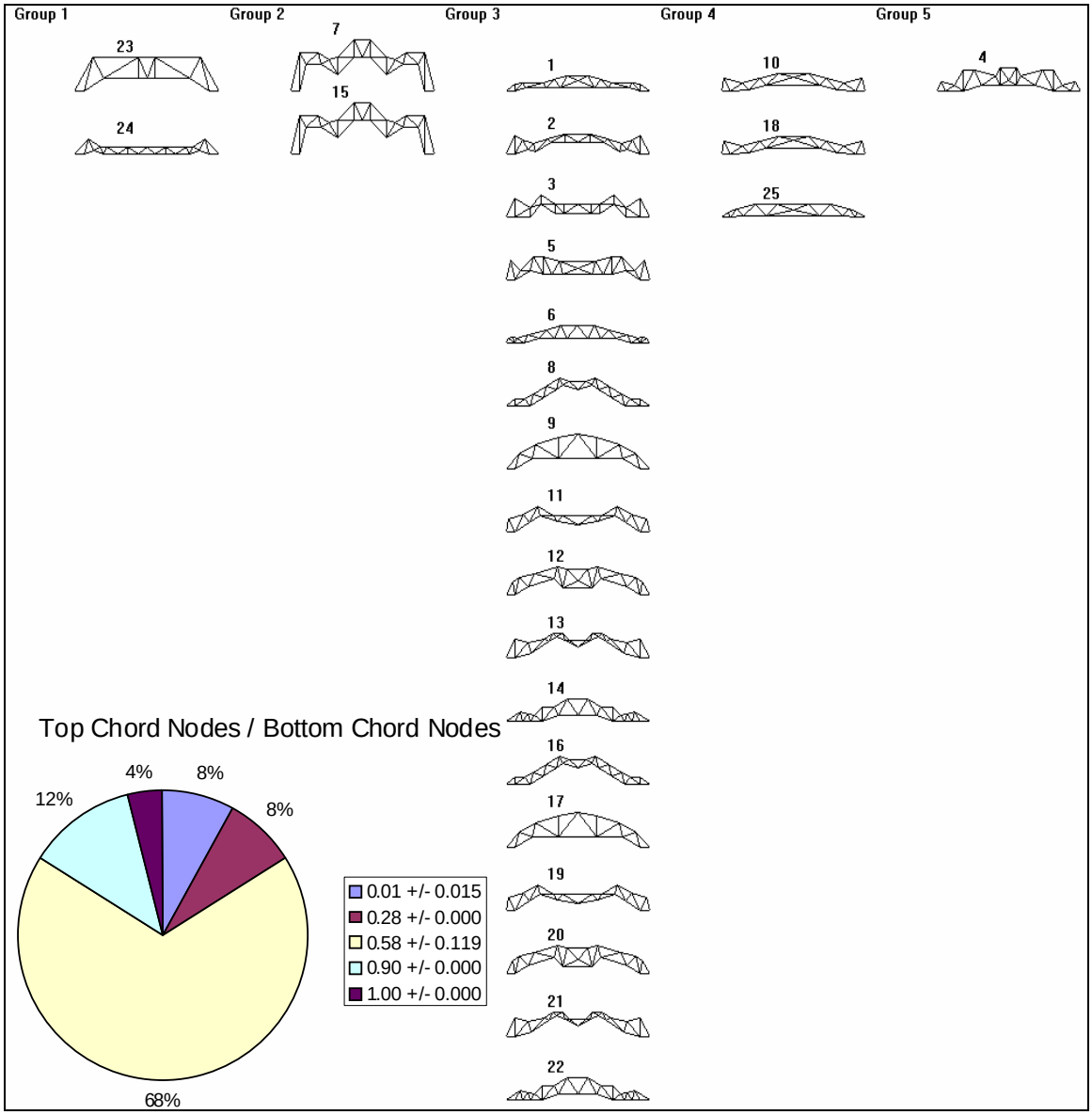


Figure A.54. Feature map created for S25M50 with ratio of top to bottom chord nodes as input.

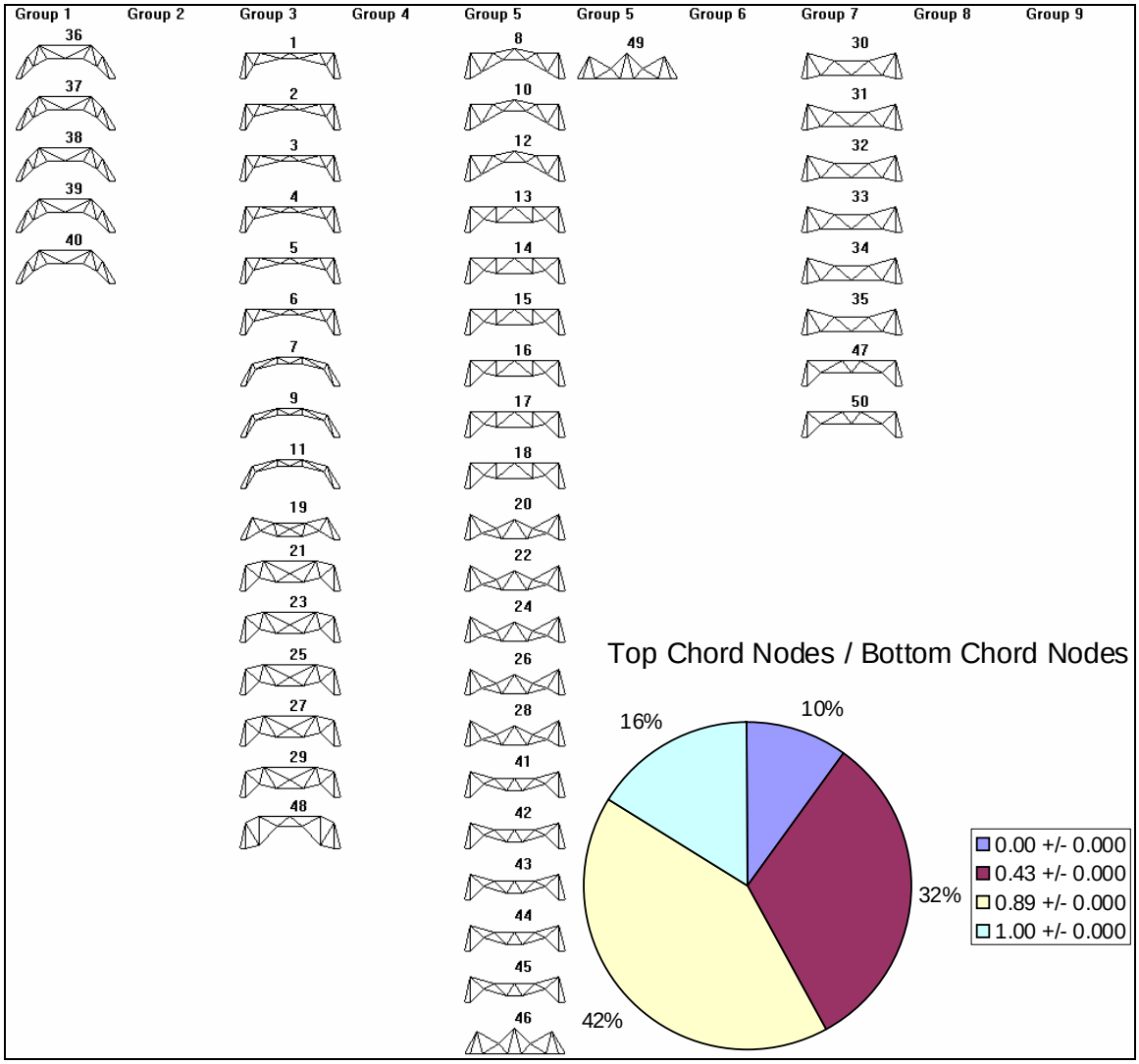


Figure A.55. Feature map created for S50M25 with ratio of top to bottom chord nodes as input.

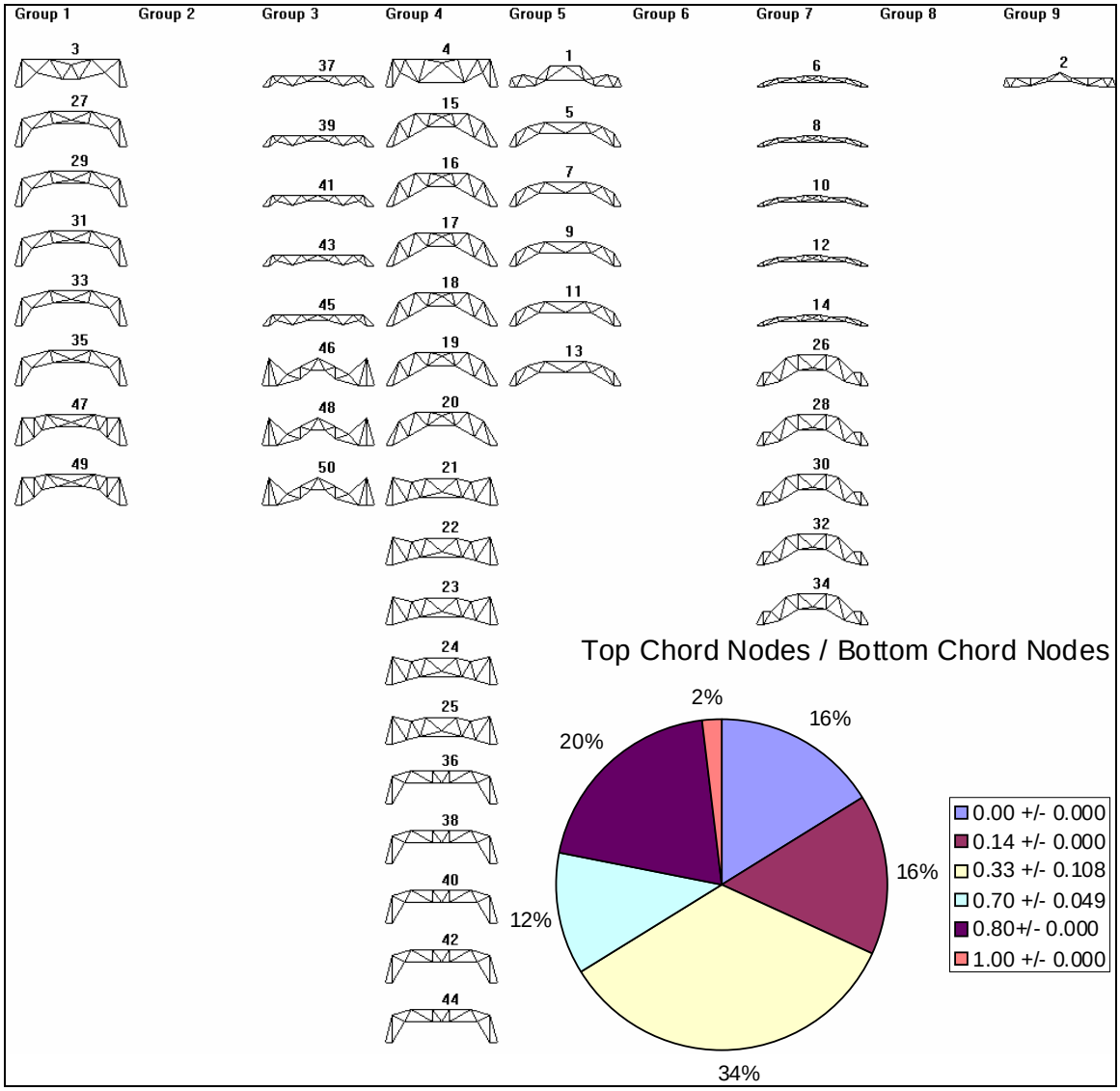


Figure A.56. Feature map created for S50M35 with ratio of top to bottom chord nodes as input.

Bottom Chord Flatness

Figures A.57 through A.60 show feature maps for populations S25M35, S25M50, S50M25, and S50M35, respectively, for the measure of bottom chord flatness.

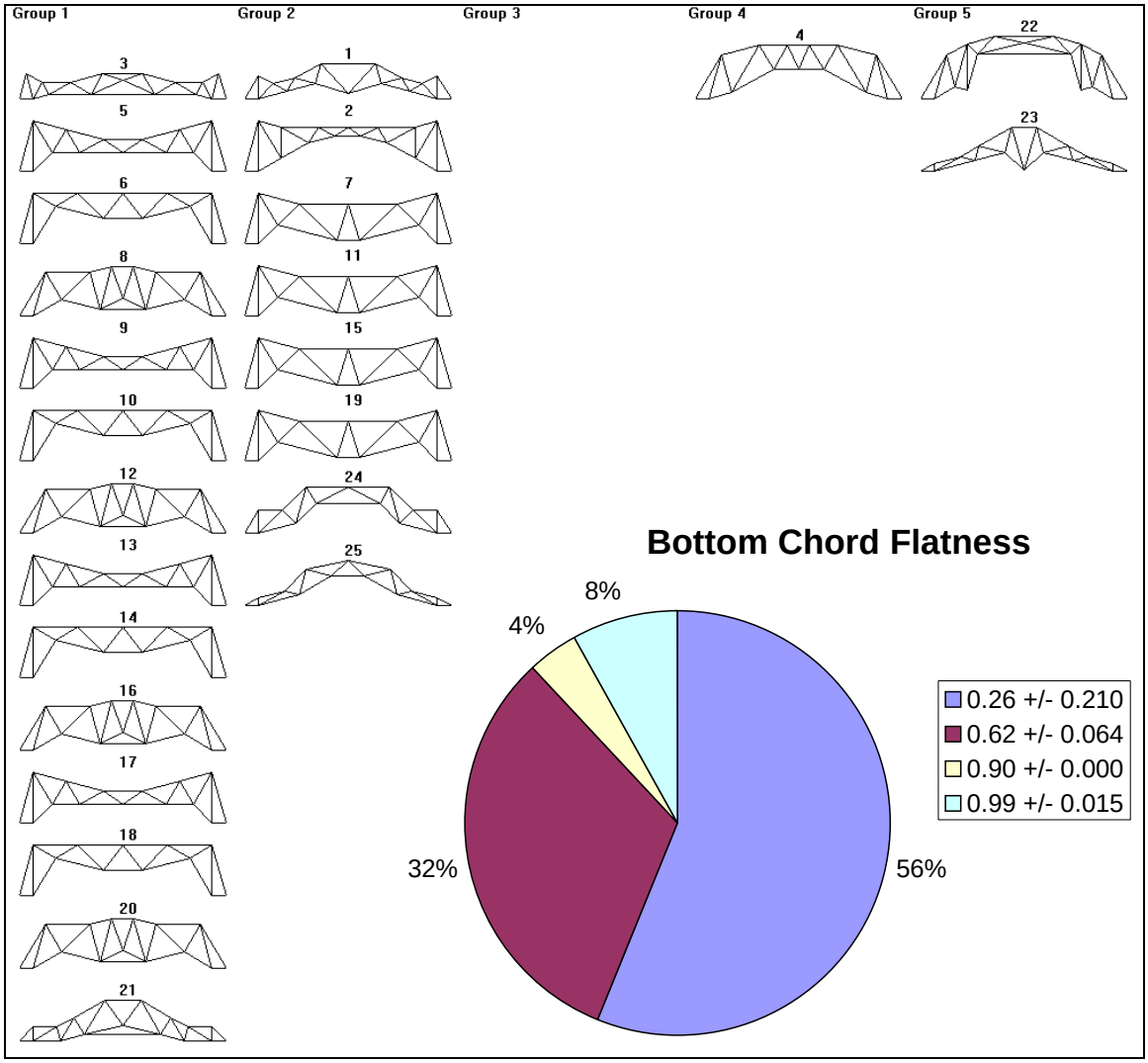


Figure A.57. Feature map created for S25M35 with bottom chord flatness as input.

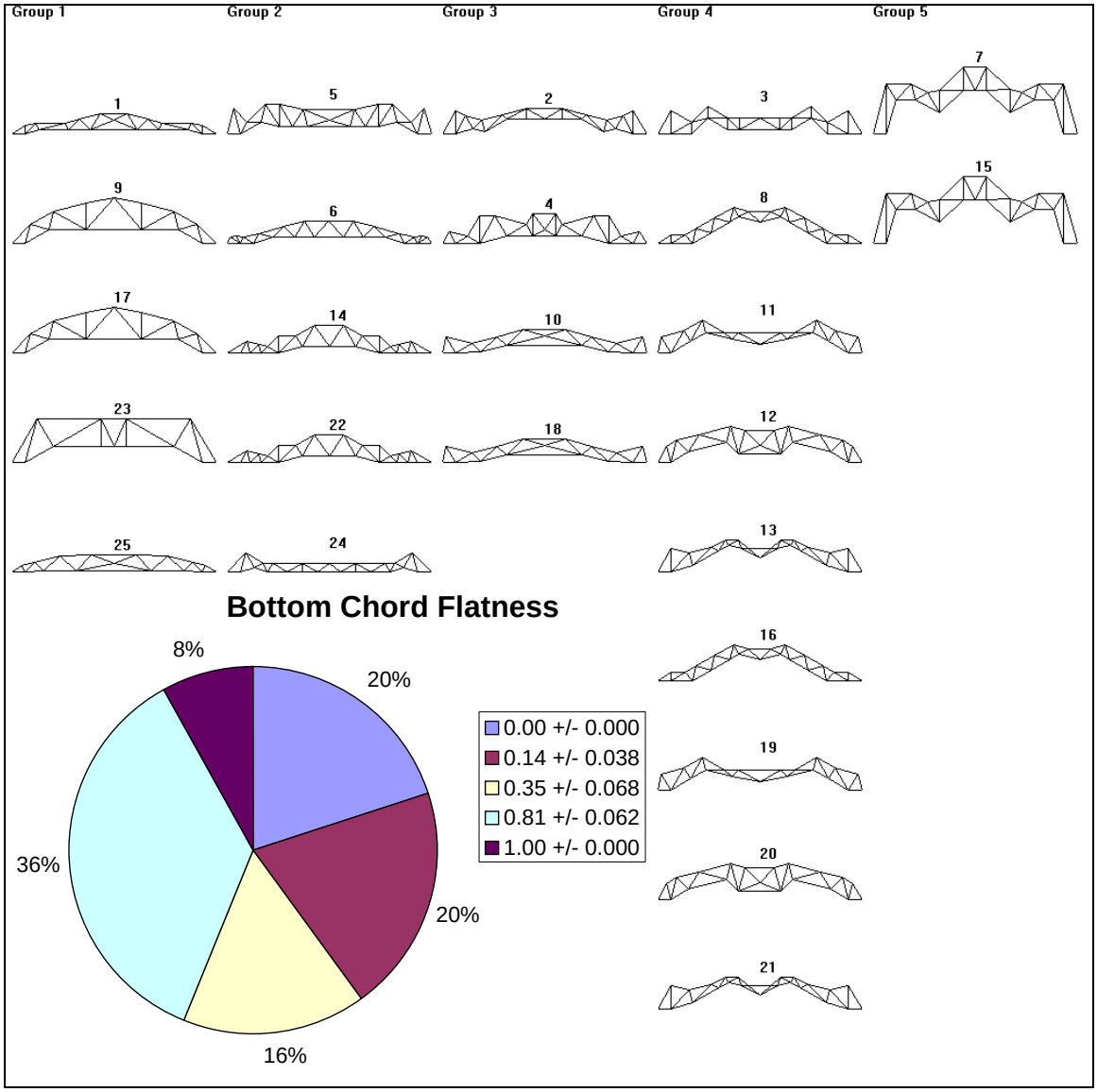


Figure A.58. Feature map created for S25M50 with bottom chord flatness as input.

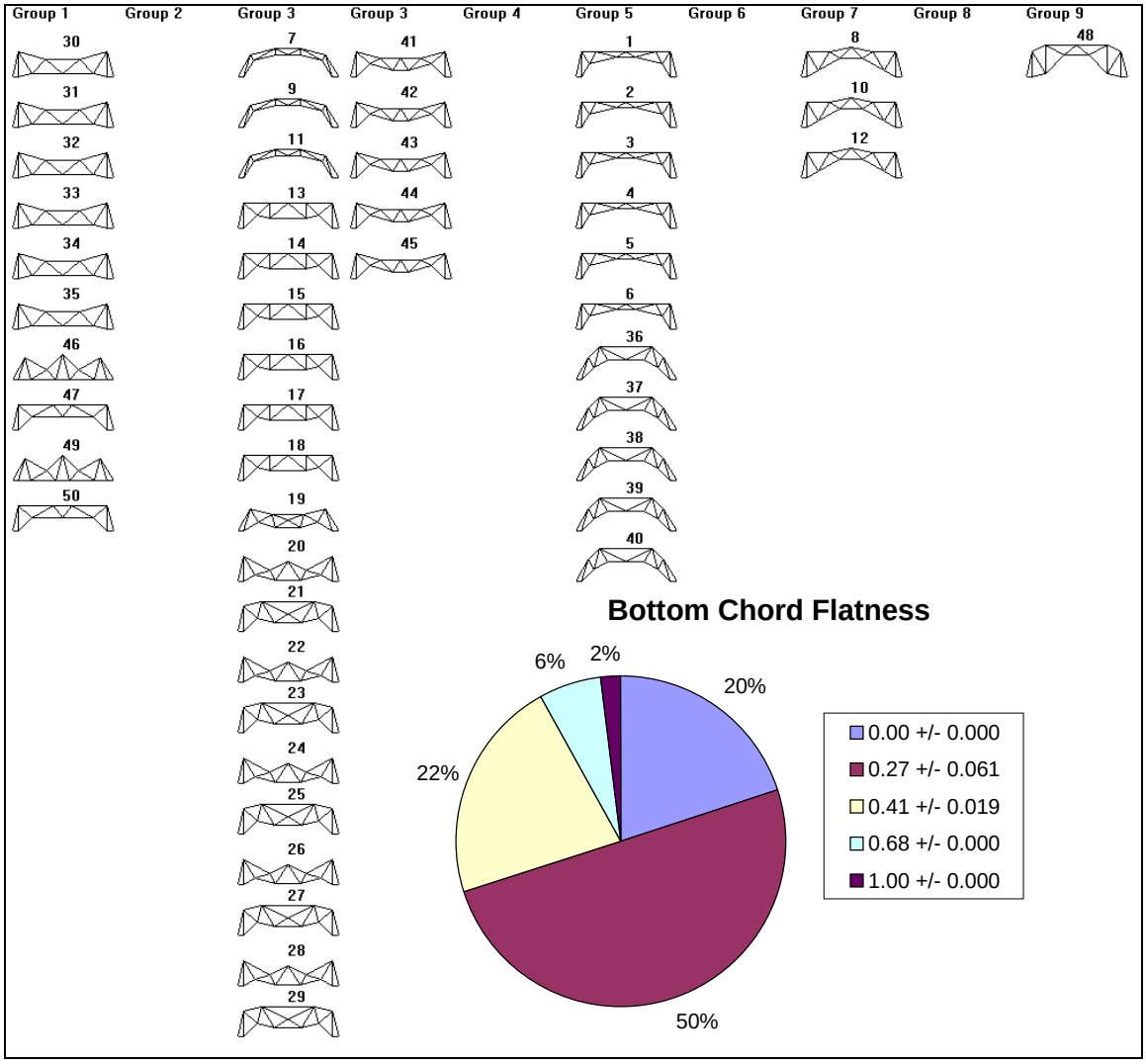


Figure A.59. Feature map created for S50M25 with bottom chord flatness as input.

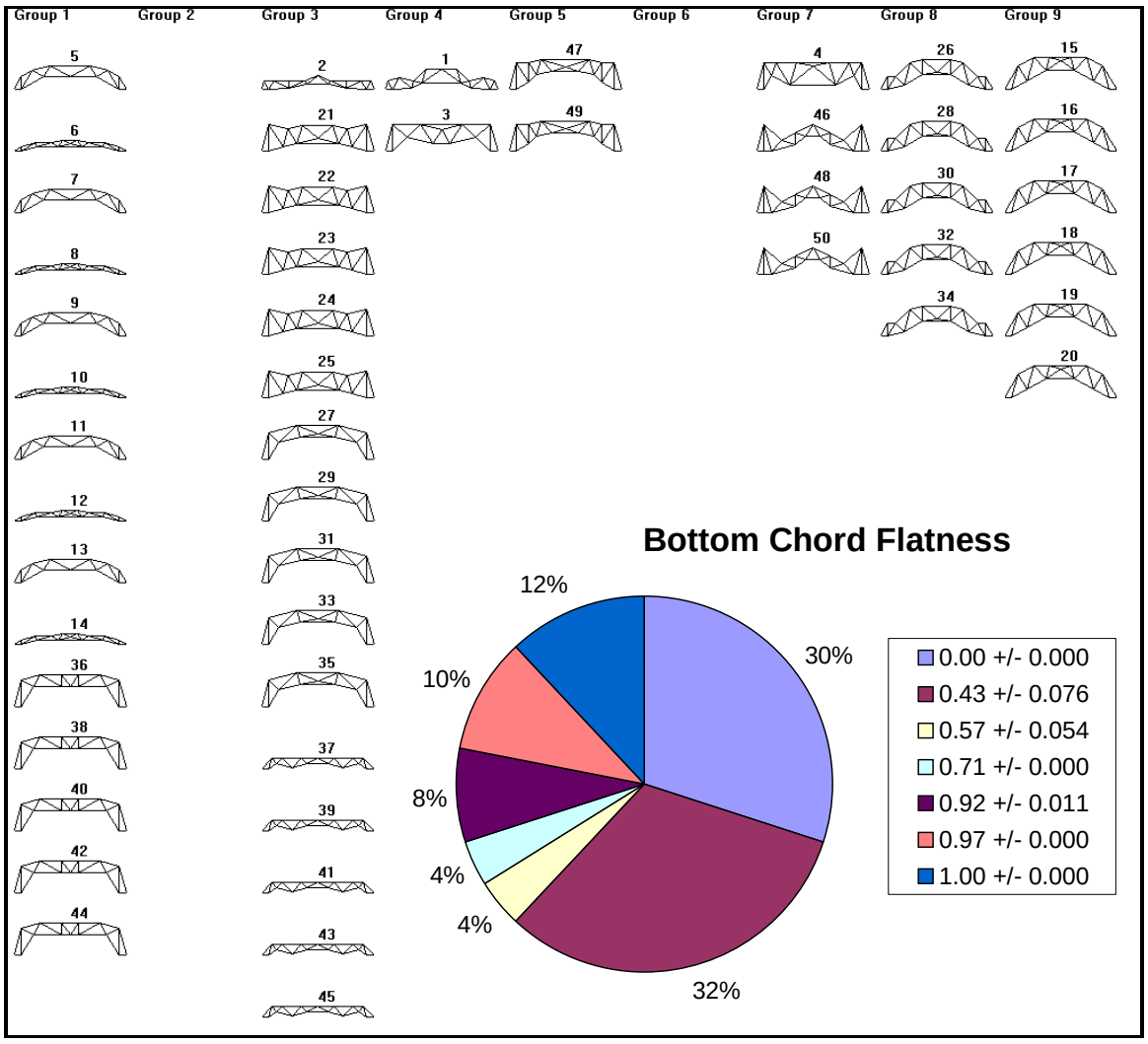


Figure A.60. Feature map created for S50M35 with bottom chord flatness as input.

Top Chord Angularity

Figures A.61 through A.64 show feature maps for populations S25M35, S25M50, S50M25, and S50M35, respectively, for top chord angularity.

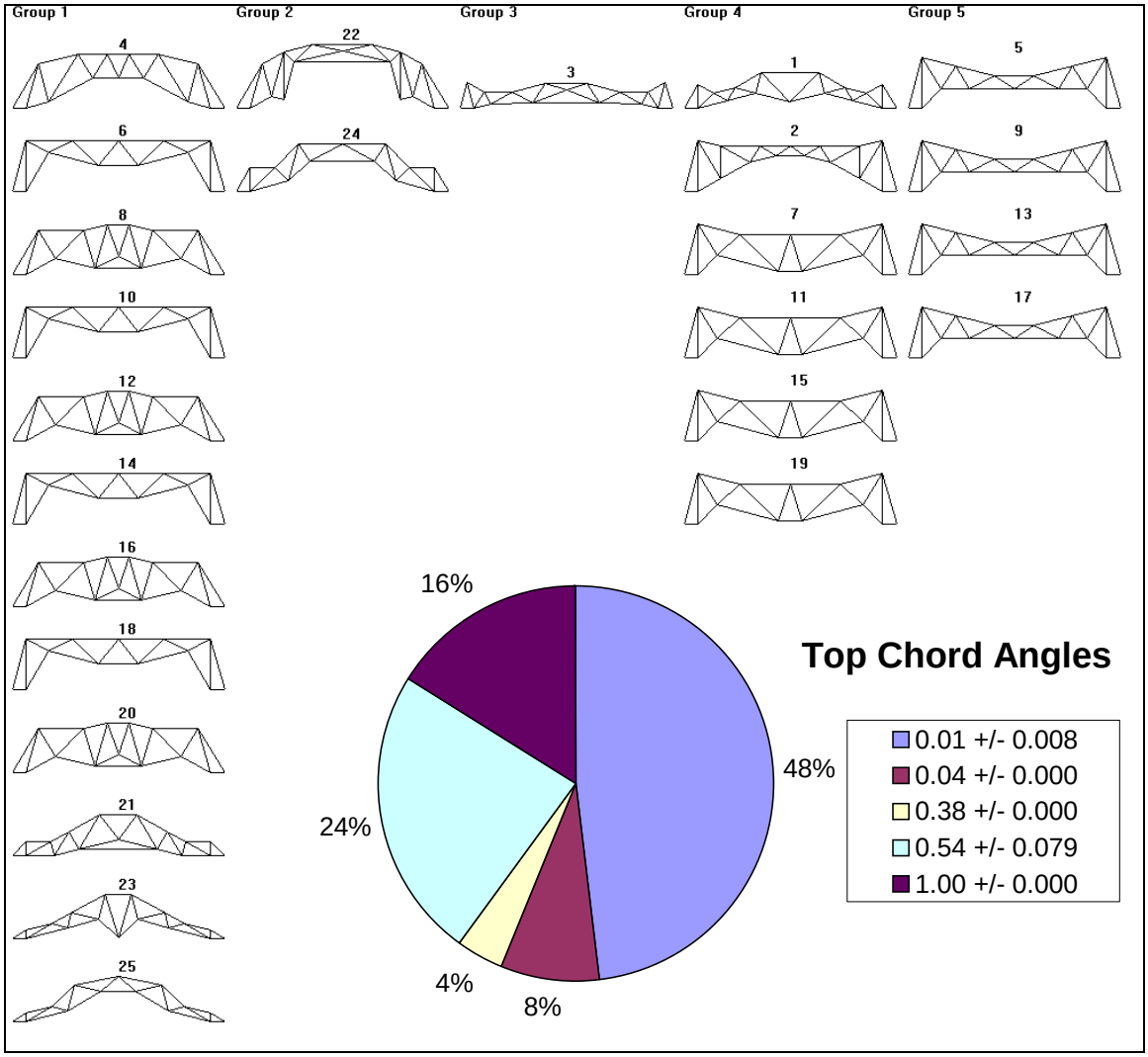


Figure A.61. Feature map created for S25M35 with top chord angles as input.

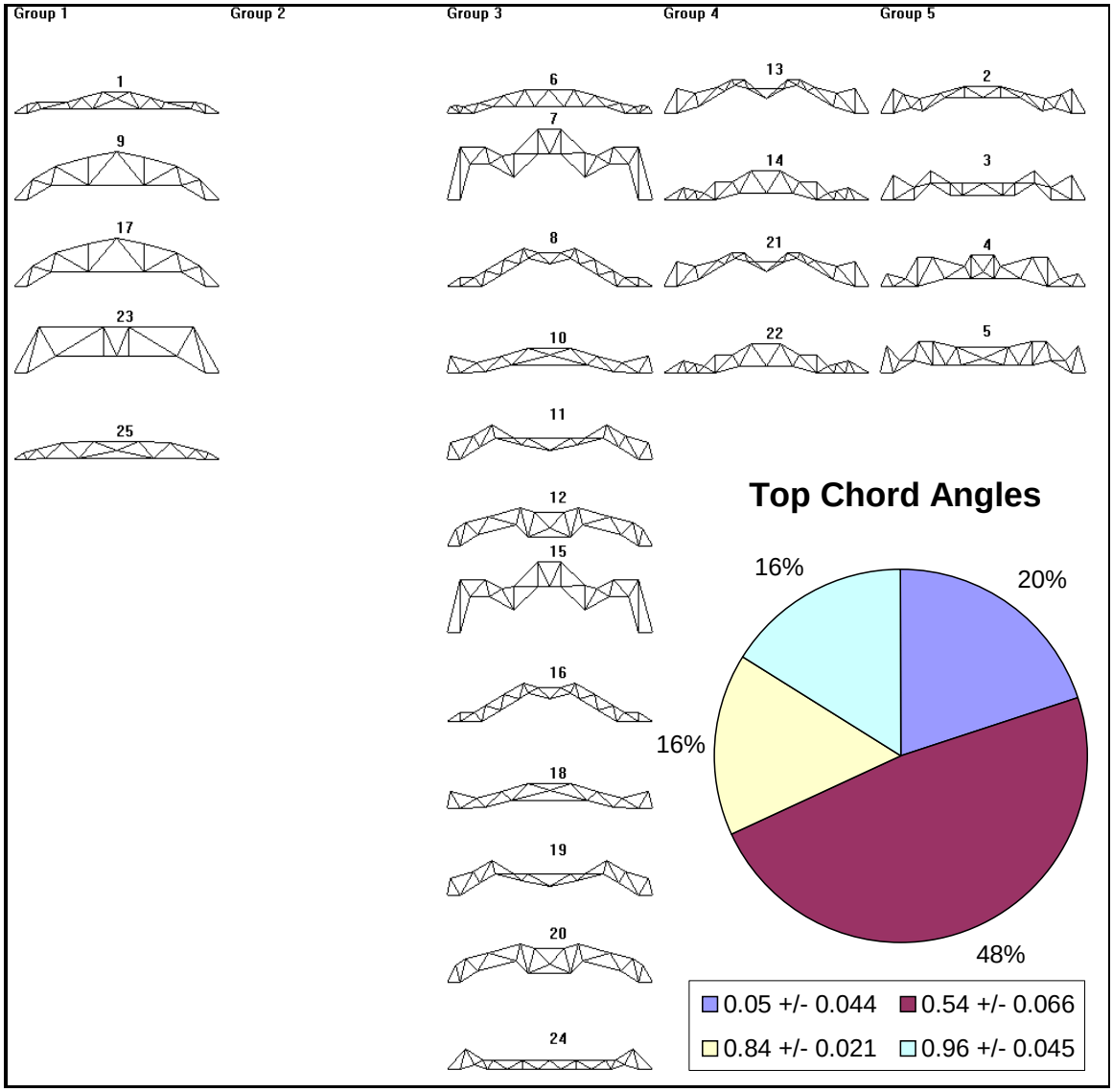


Figure A.62. Feature map created for S25M50 with top chord angles as input.

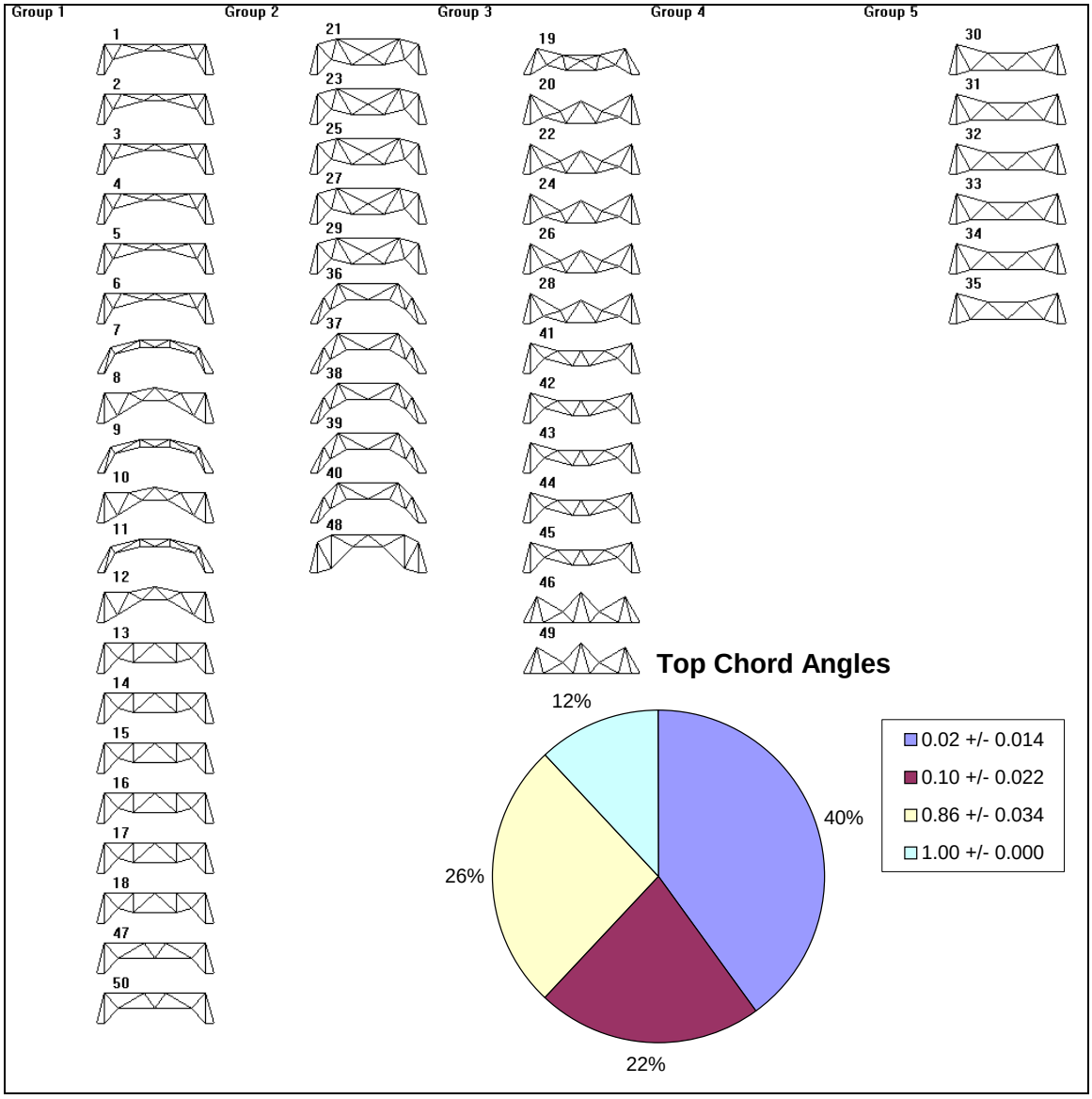


Figure A.63. Feature map created for S50M25 with top chord angles as input.

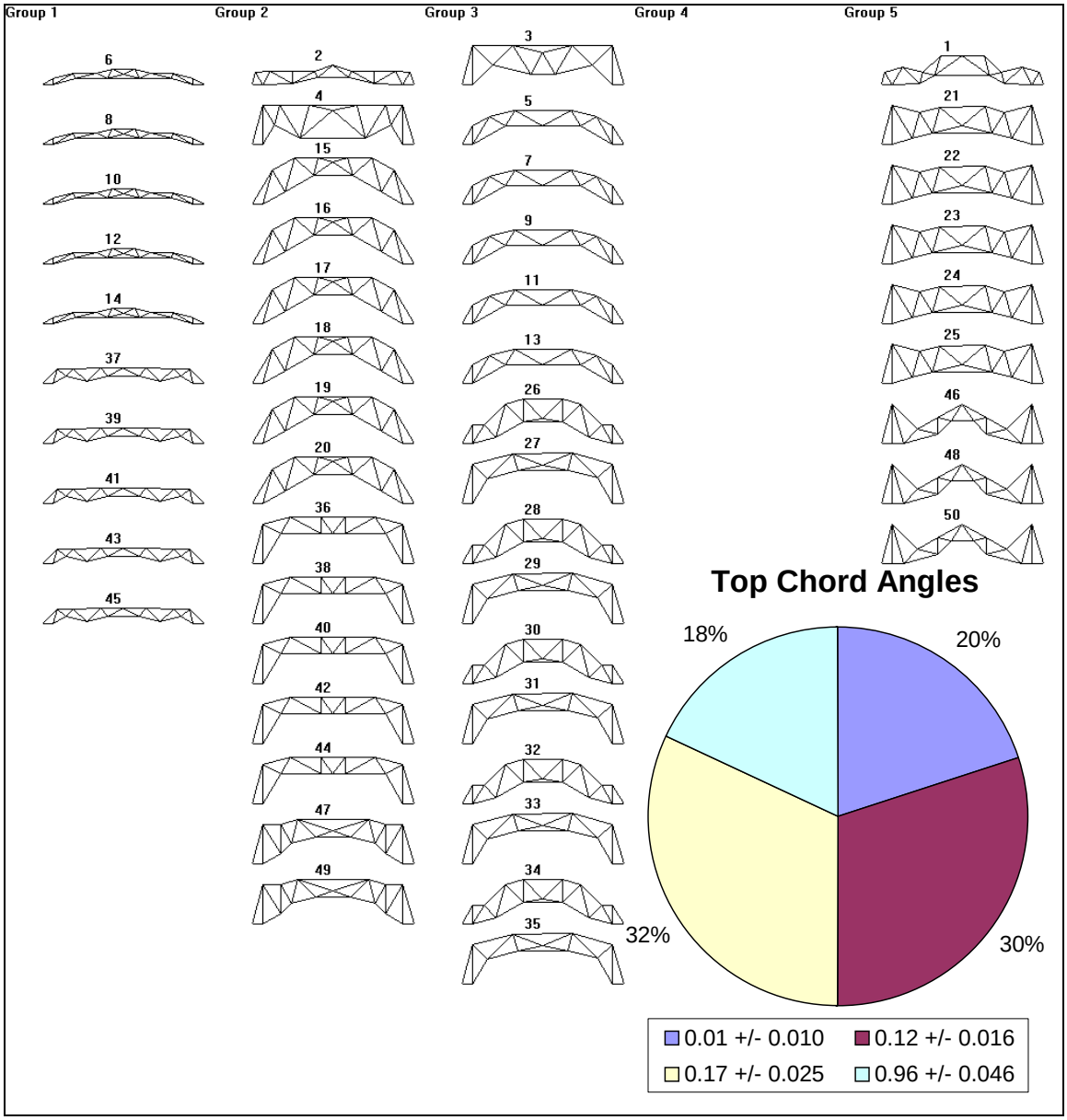


Figure A.64. Feature map created for S50M35 with top chord angles as input.

Bottom Chord Angularity

Figures A.65 through A.68 show feature maps for populations S25M35, S25M50, S50M25, and S50M35, respectively, for bottom chord angles.

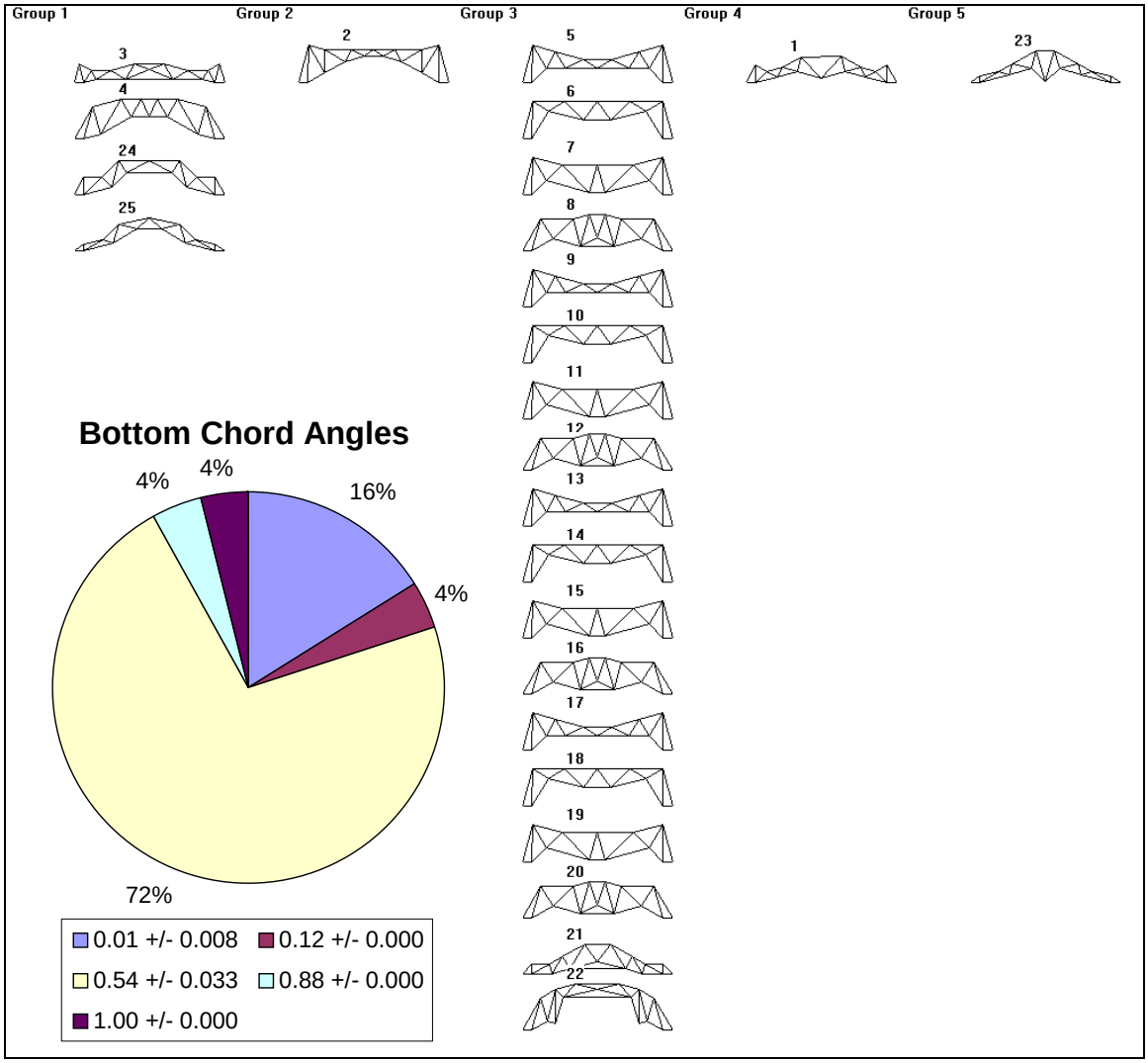


Figure A.65. Feature map created for S25M35 with bottom chord angles as input.

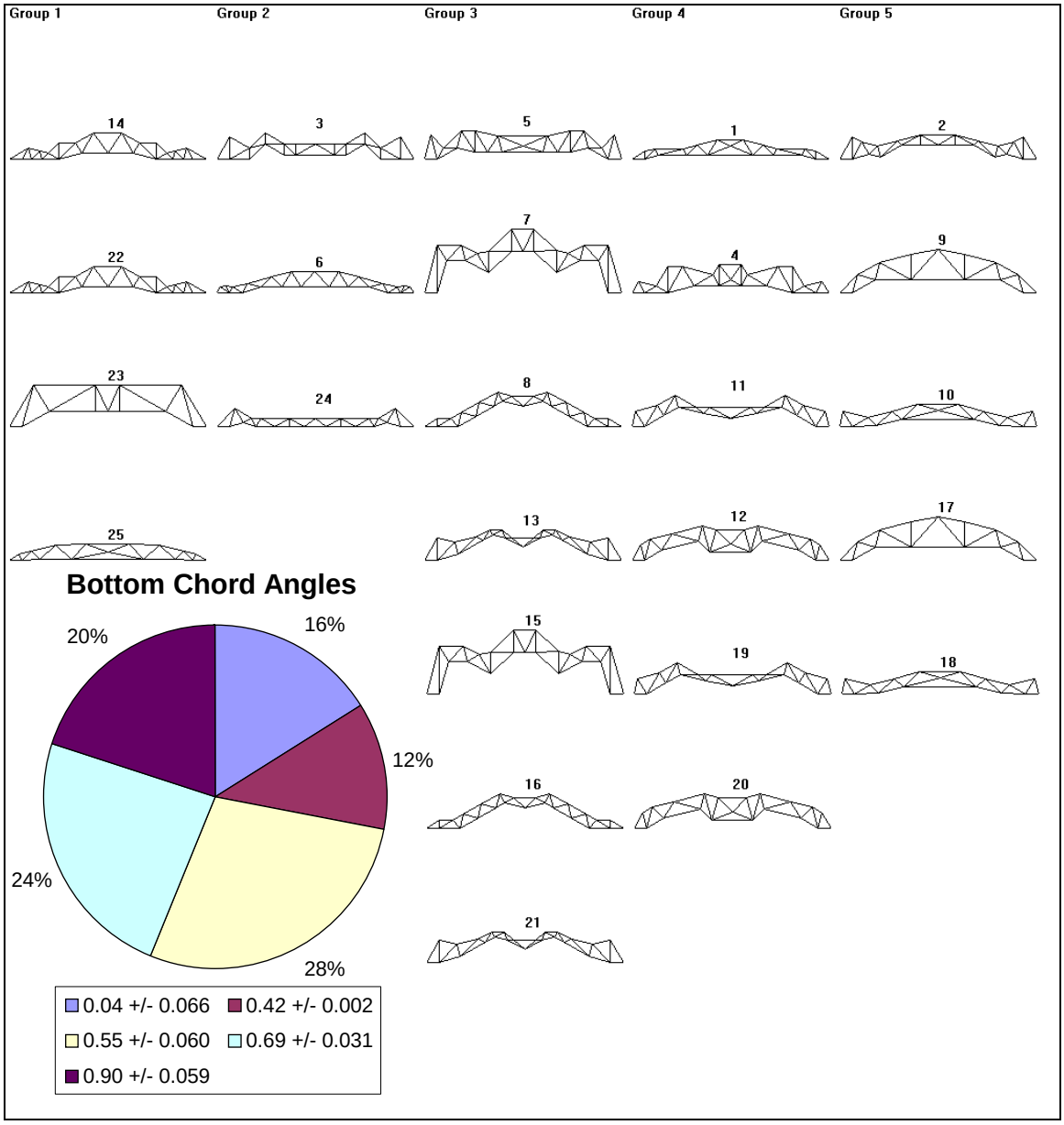


Figure A.66. Feature map created for S25M50 with bottom chord angles as input.

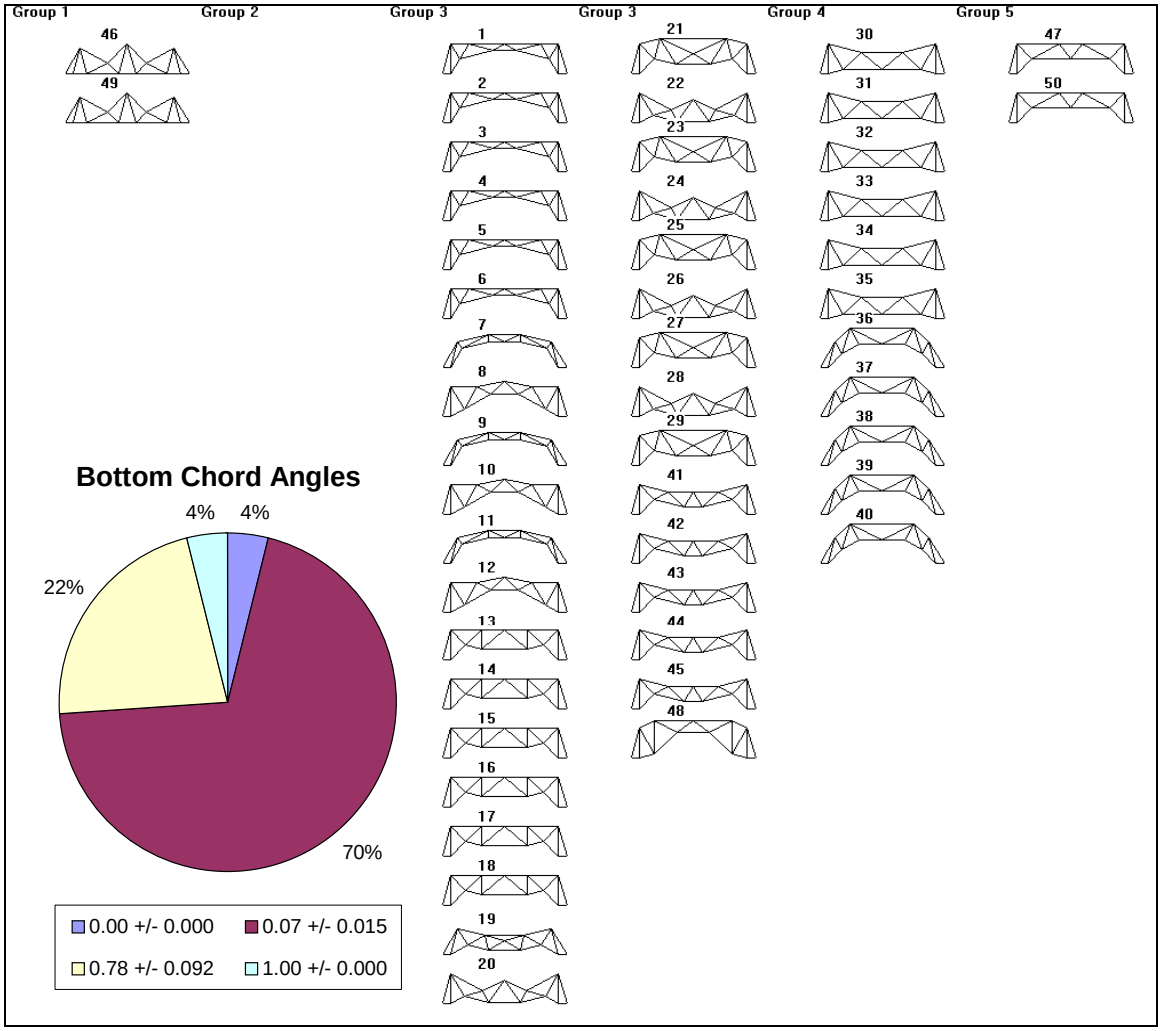


Figure A.67. Feature map created for S50M25 with bottom chord angles as input.

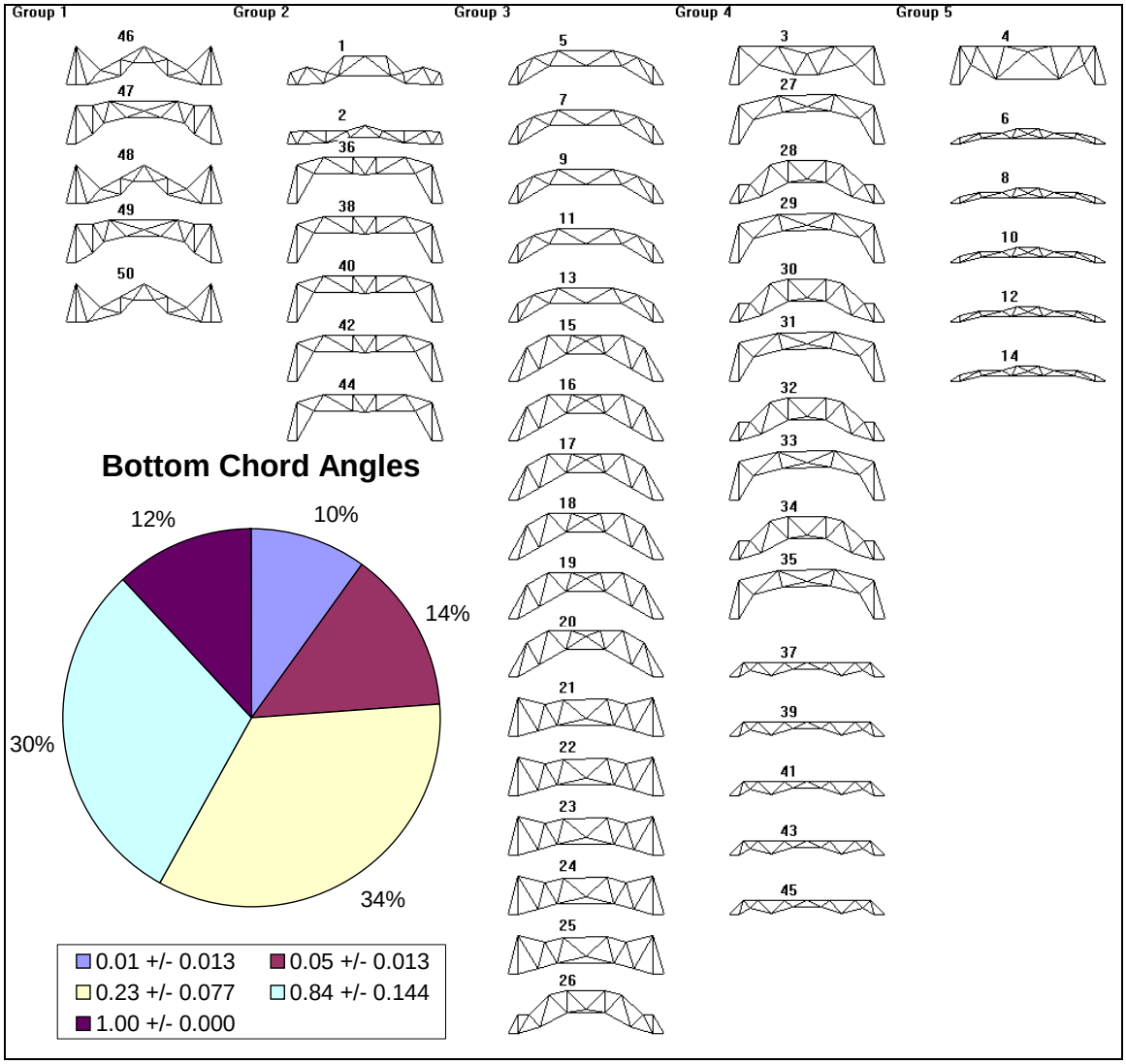


Figure A.68. Feature map created for S50M35 with bottom chord angles as input.

Top Chord Direction Changes

Figures A.69 through A.72 show feature maps for populations S25M35, S25M50, S50M25, and S50M35, respectively, for the number of top chord direction changes.

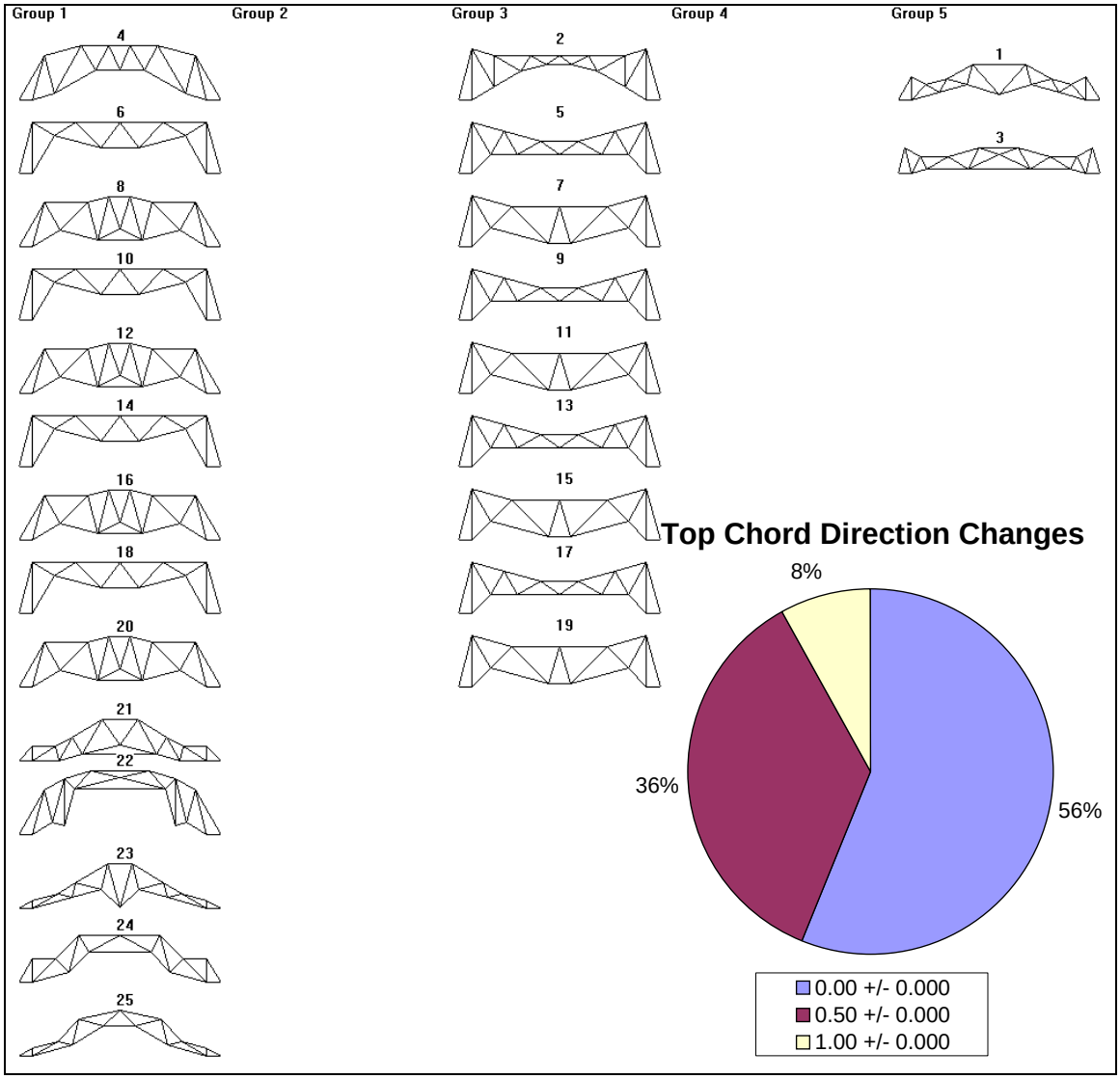


Figure A.69. Feature map created for S25M35 with number of top chord direction changes as input.

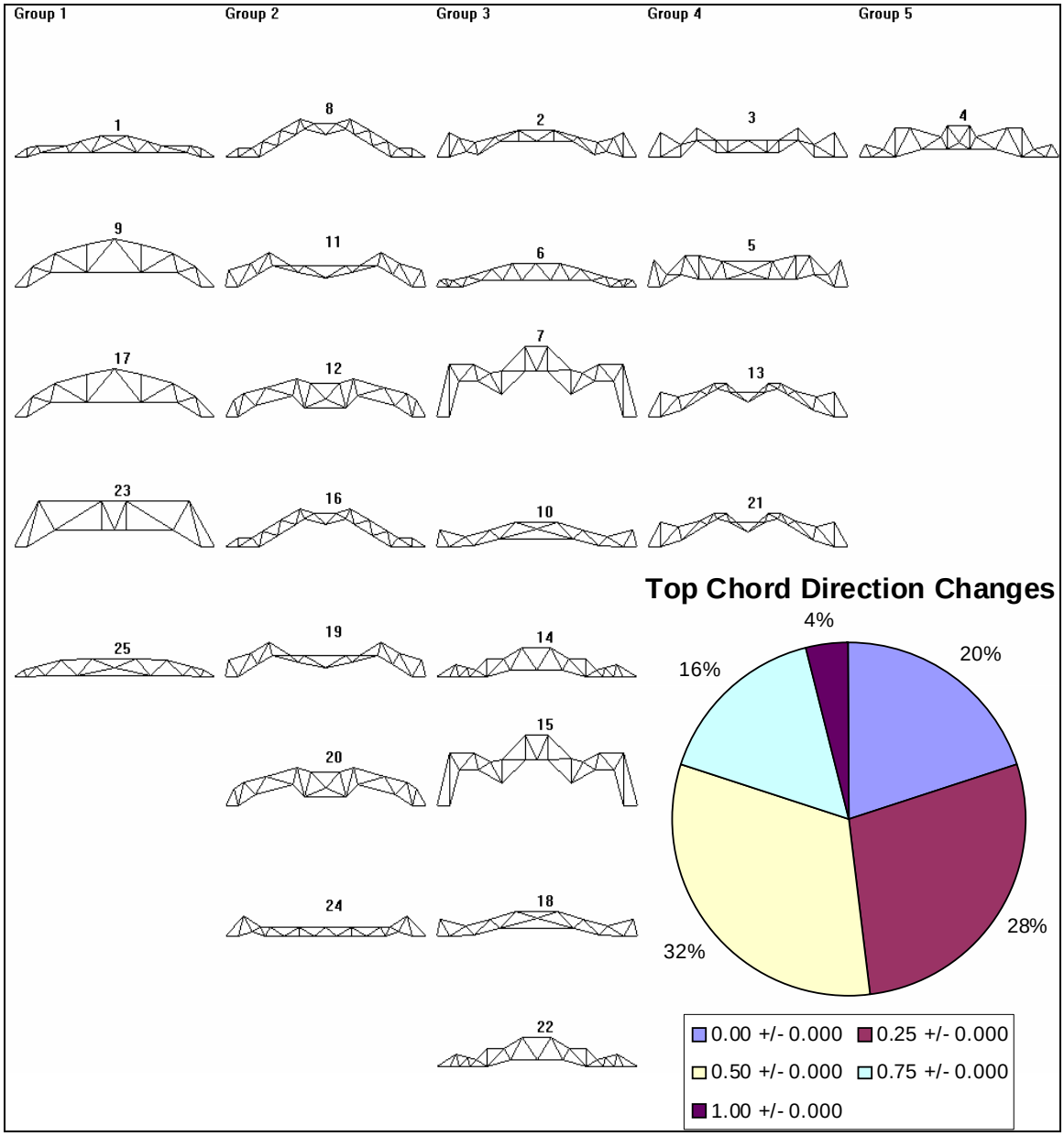


Figure A.70. Feature map created for S25M50 with number of top chord direction changes as input.

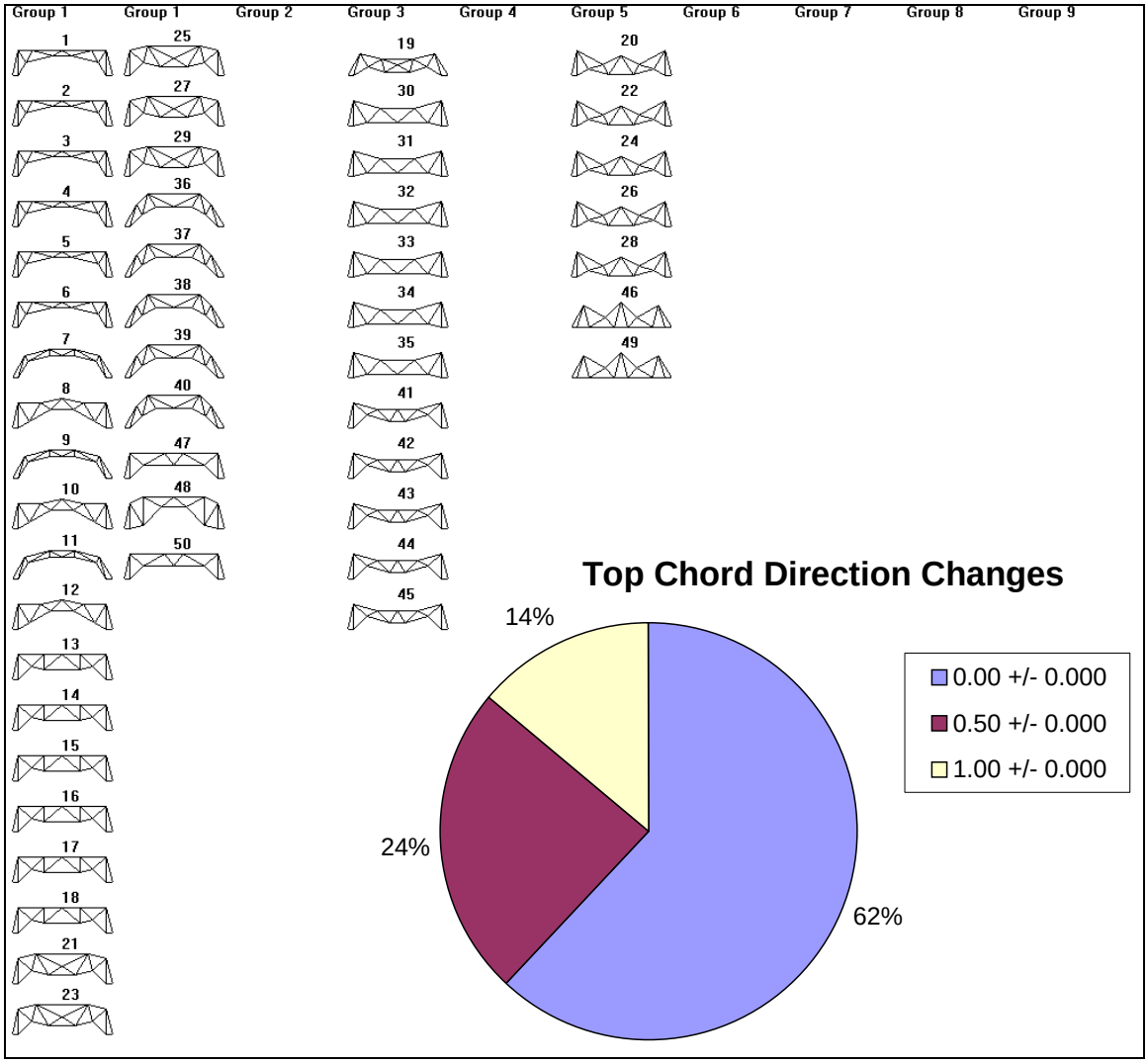


Figure A.71. Feature map created for S50M25 with number of top chord direction changes as input.

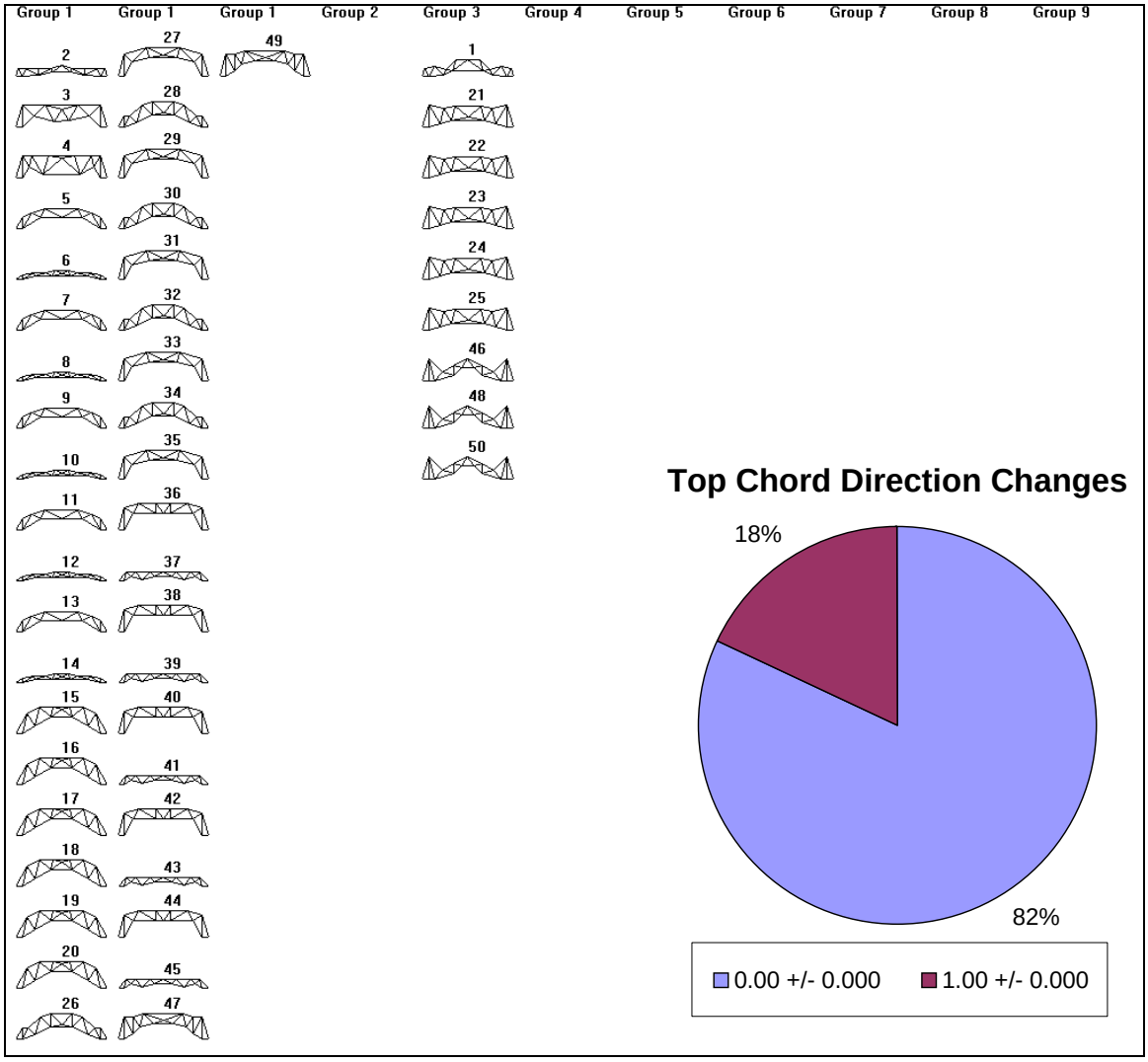


Figure A.72. Feature map created for S50M35 with number of top chord direction changes as input.

Bottom Chord Direction Changes

Figures A.73 through A.76 show feature maps for populations S25M35, S25M50, S50M25, and S50M35, respectively, for the number of bottom chord direction changes.

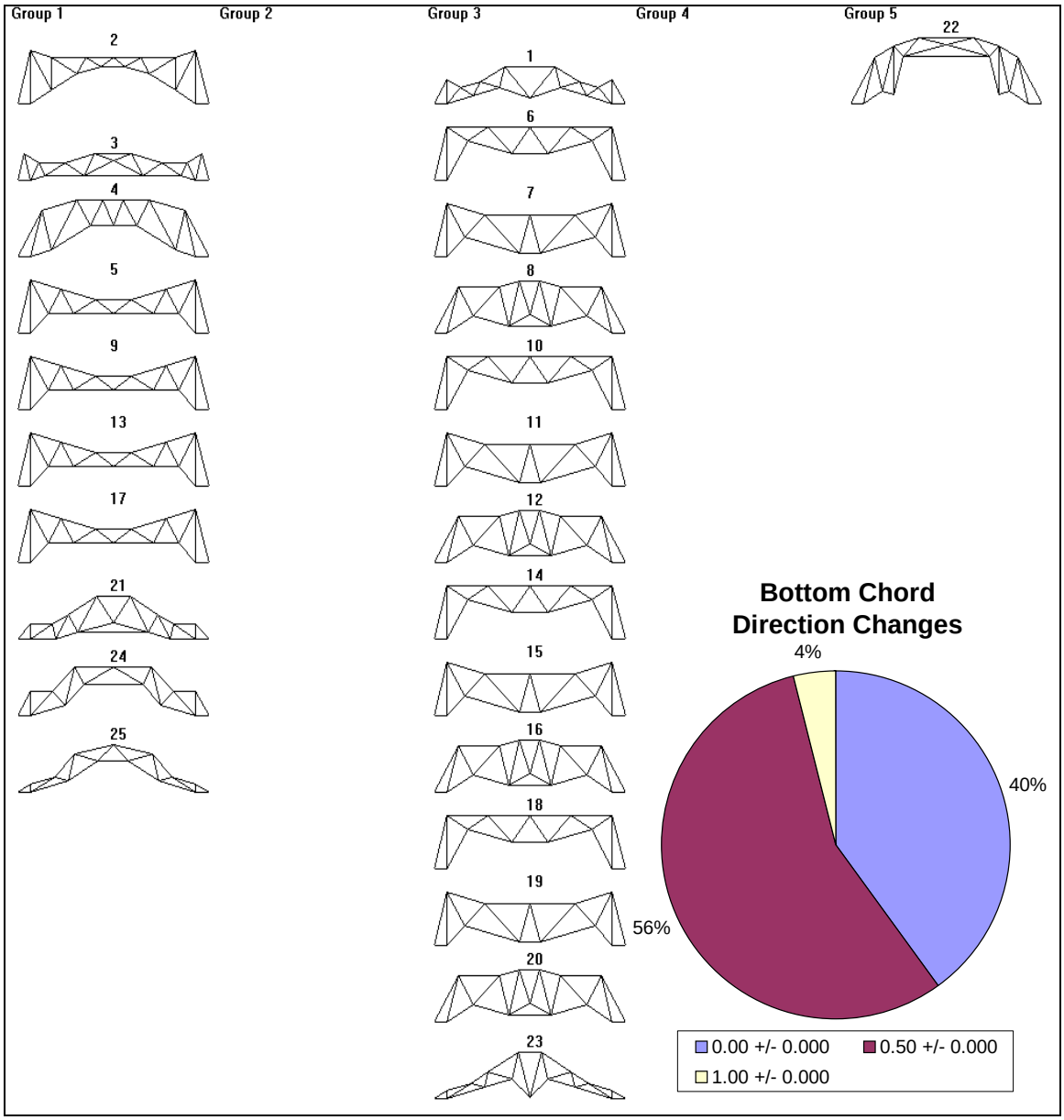


Figure A.73. Feature map created for S25M35 with number of bottom chord direction changes as input.

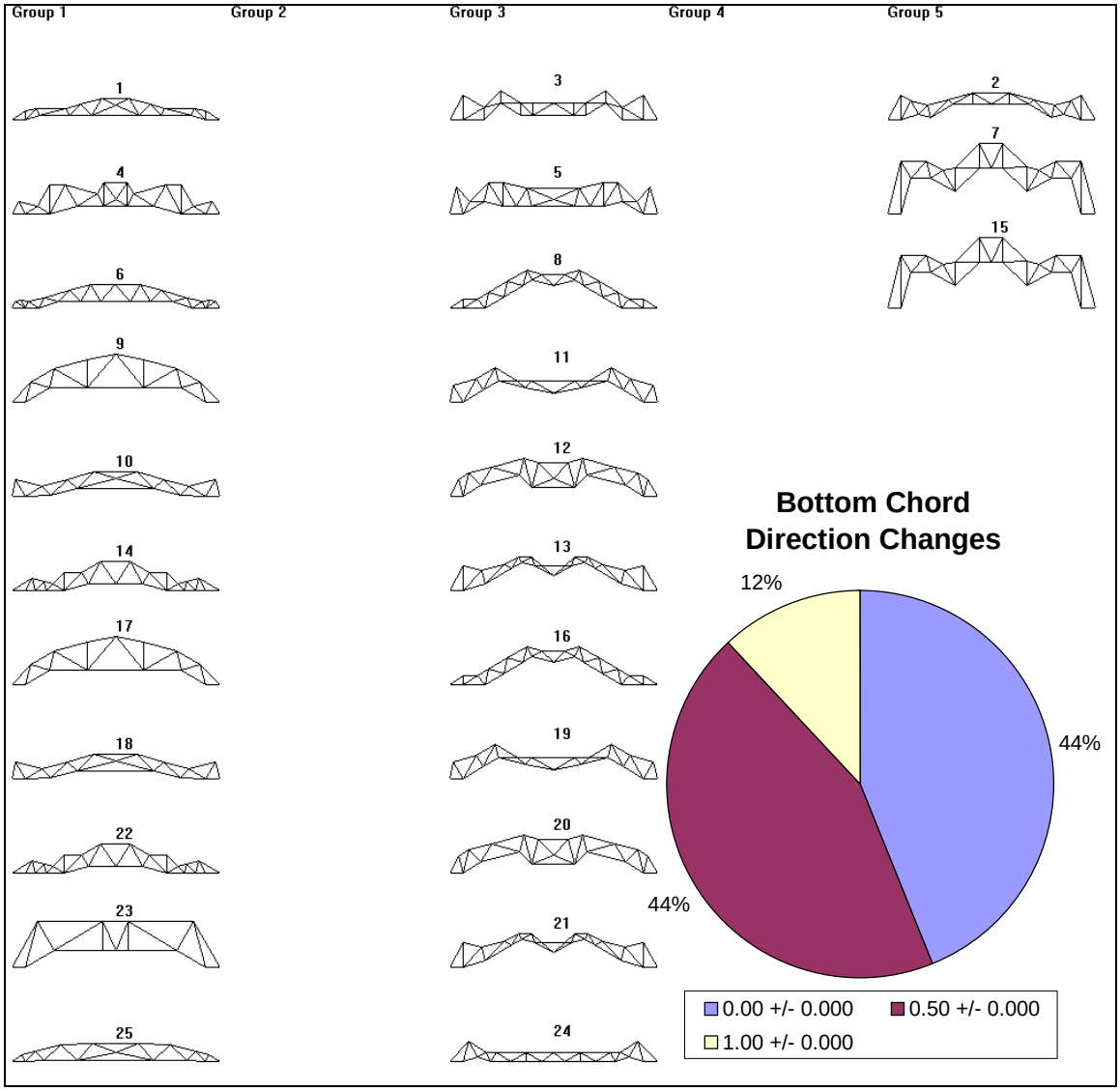


Figure A.74. Feature map created for S25M50 with number of bottom chord direction changes as input.

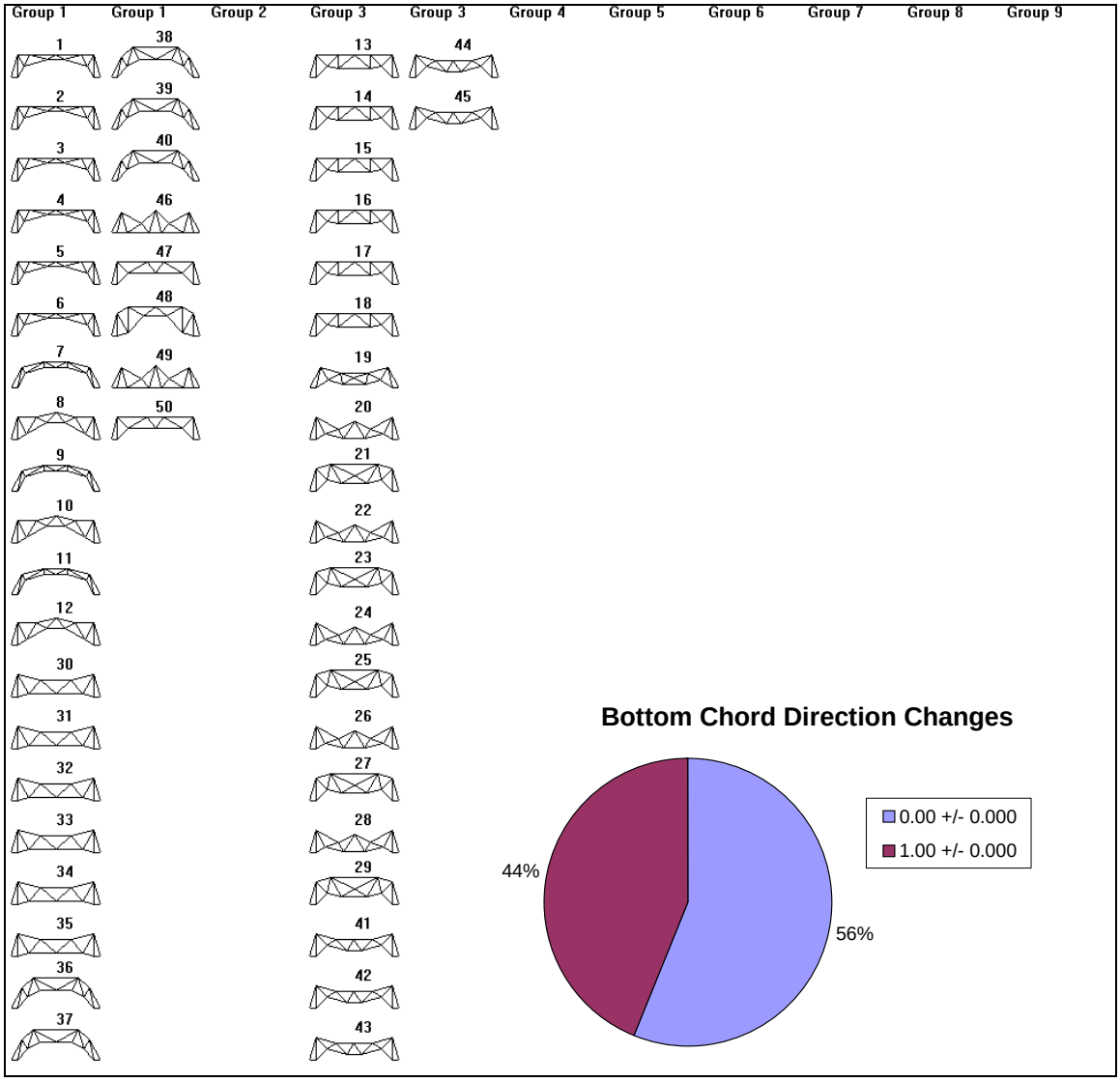


Figure A.75. Feature map created for S50M25 with number of bottom chord direction changes as input.

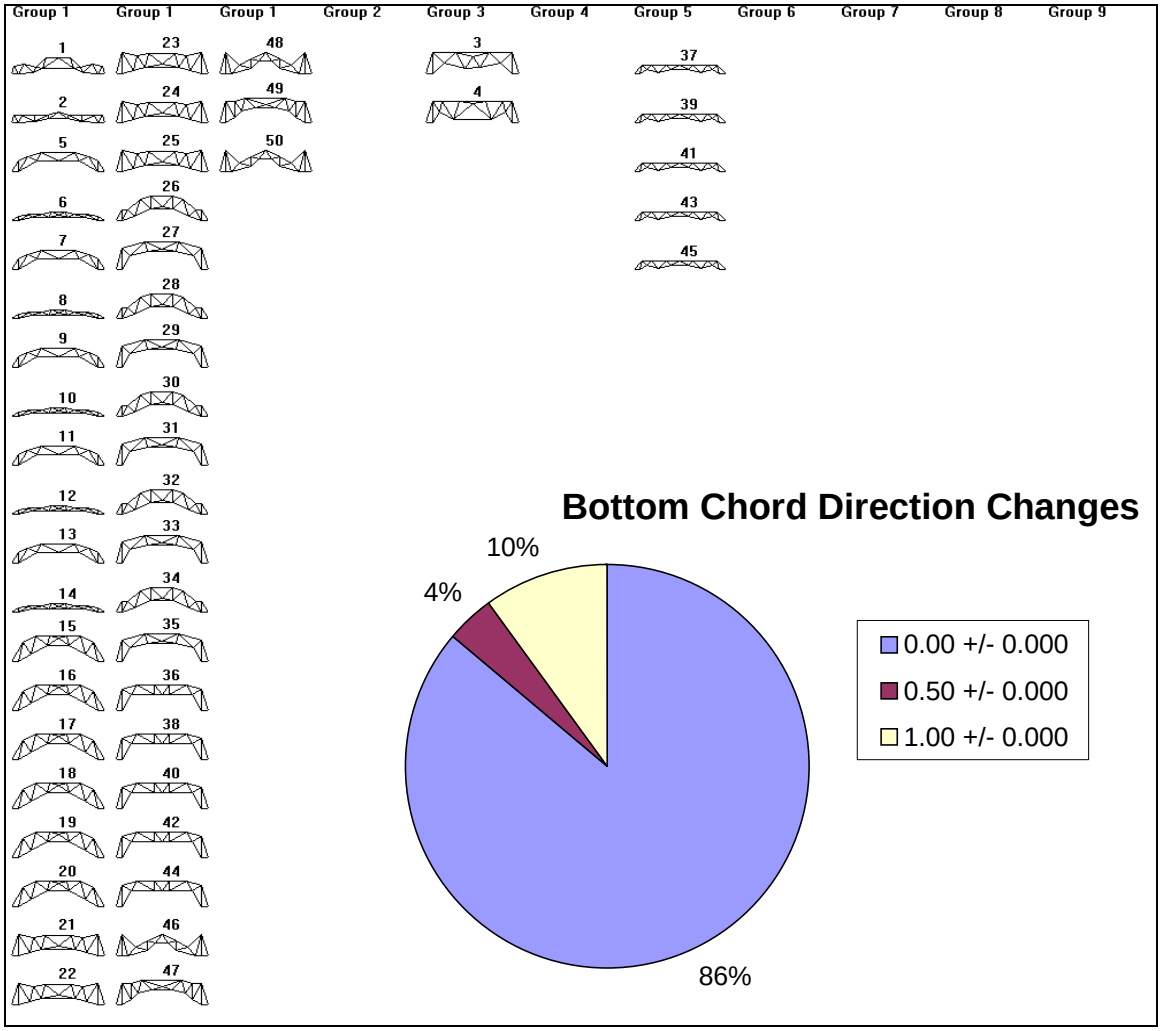


Figure A.76. Feature map created for S50M35 with number of bottom chord direction changes as input.

Truss Depth

Figures A.77 through A.80 show feature maps for populations S25M35, S25M50, S50M25, and S50M35, respectively, for truss depth.

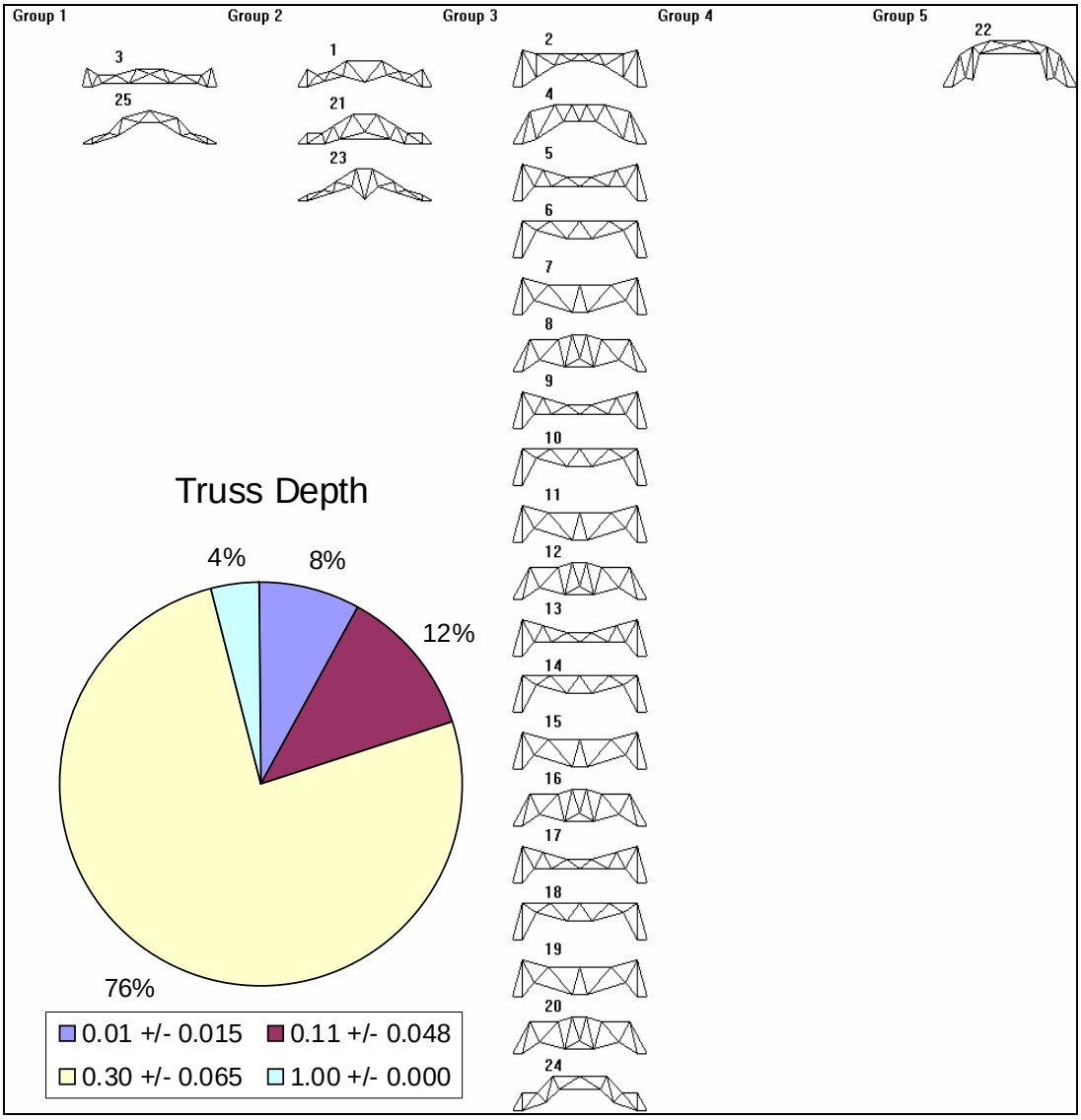


Figure A.77. Feature map created for S25M35 with truss depth as input.

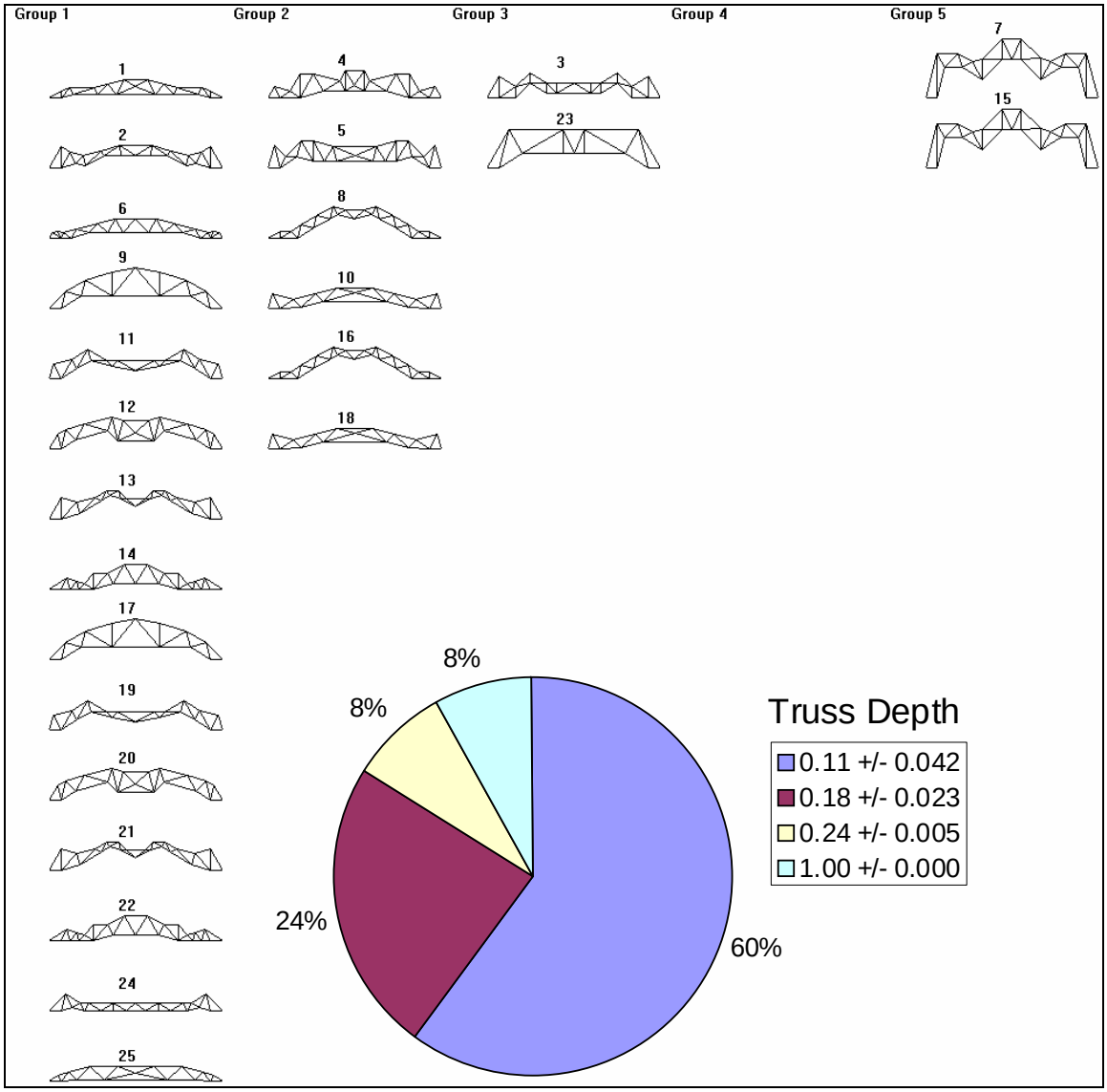


Figure A.78. Feature map created for S25M50 with truss depth as input.

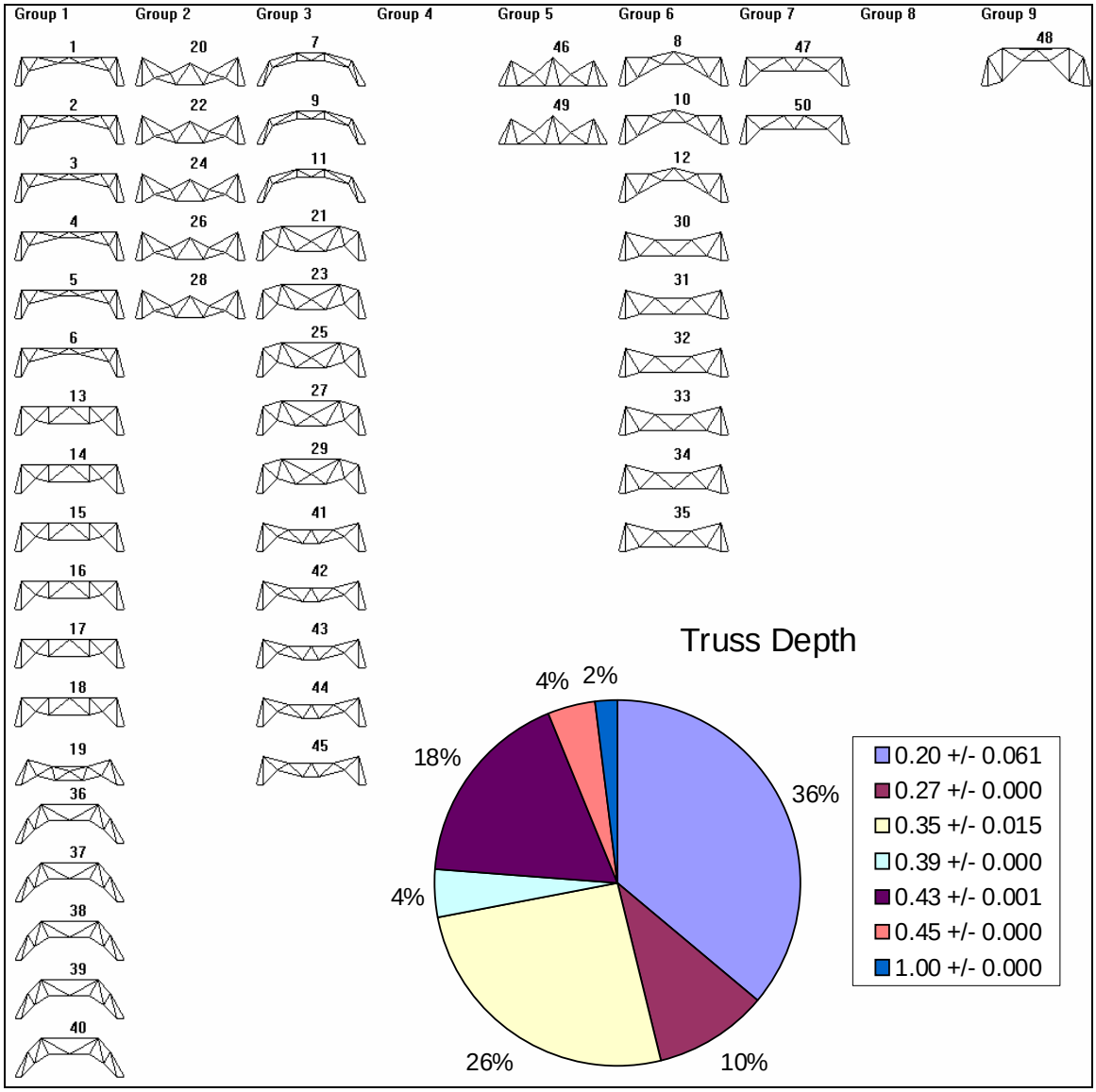


Figure A.79. Feature map created for S50M25 with truss depth as input.

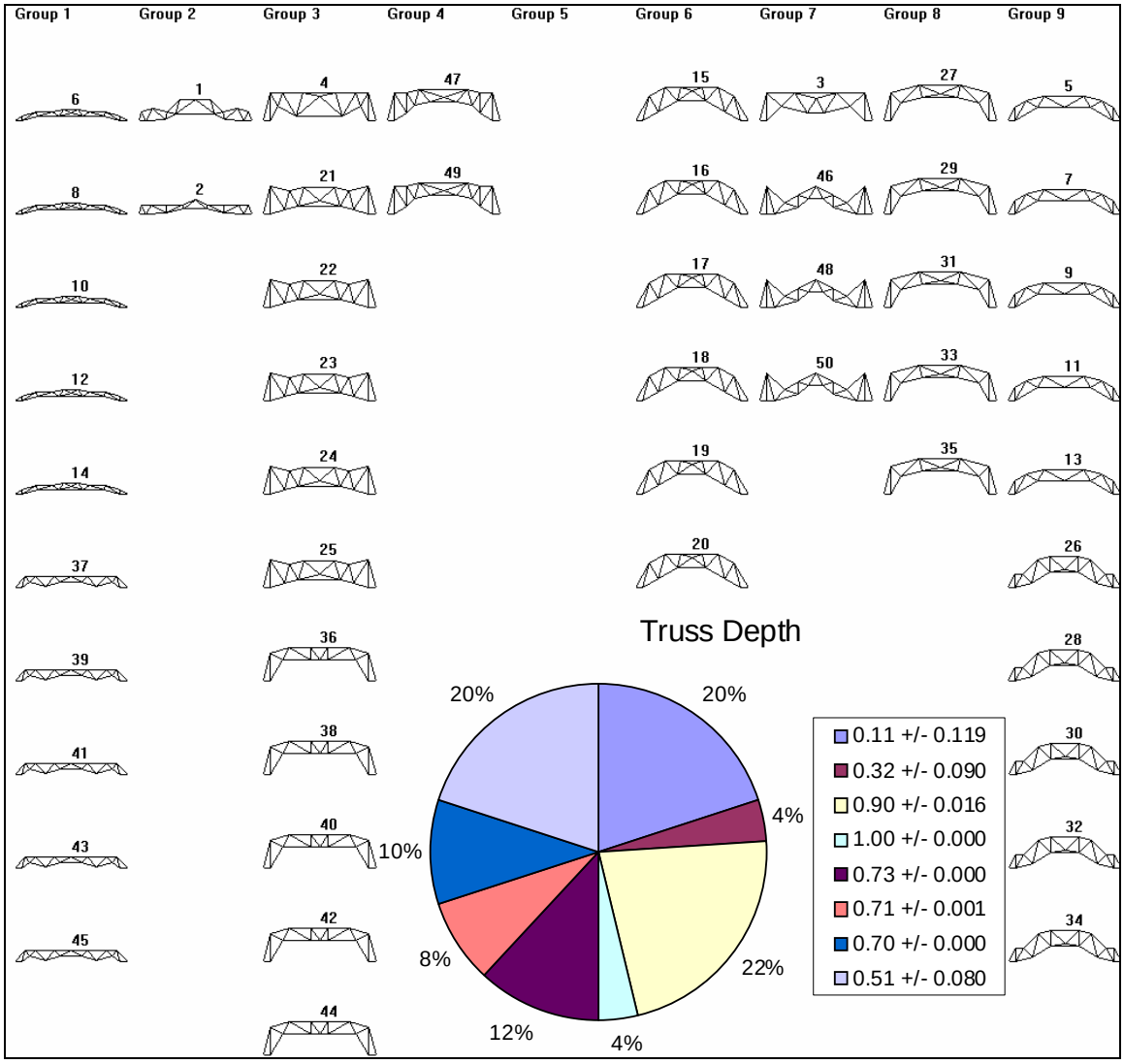


Figure A.80. Feature map created for S50M35 with truss depth as input.

APPENDIX B

This appendix presents additional results for Chapter III, which focuses on the selection of a classification mechanism to subdivide a truss population into manageable groups of trusses with similar design features. Results shown here are of two types. First, clustering maps for populations S25M35, S25M50, S50M25, and S50M50 are shown. These maps represent the classifications proposed by the 2D KSOM, 1D KSOM, k-means, and nearest neighbor algorithms, respectively. The maps are shown in Figures B.1 through B.16.

Statistical tables for each population are presented at the conclusion of the clustering maps. These tables show the averages and standard deviations as well as maximum and minimum data points in the groups identified by the clustering methods.

Clustering Maps

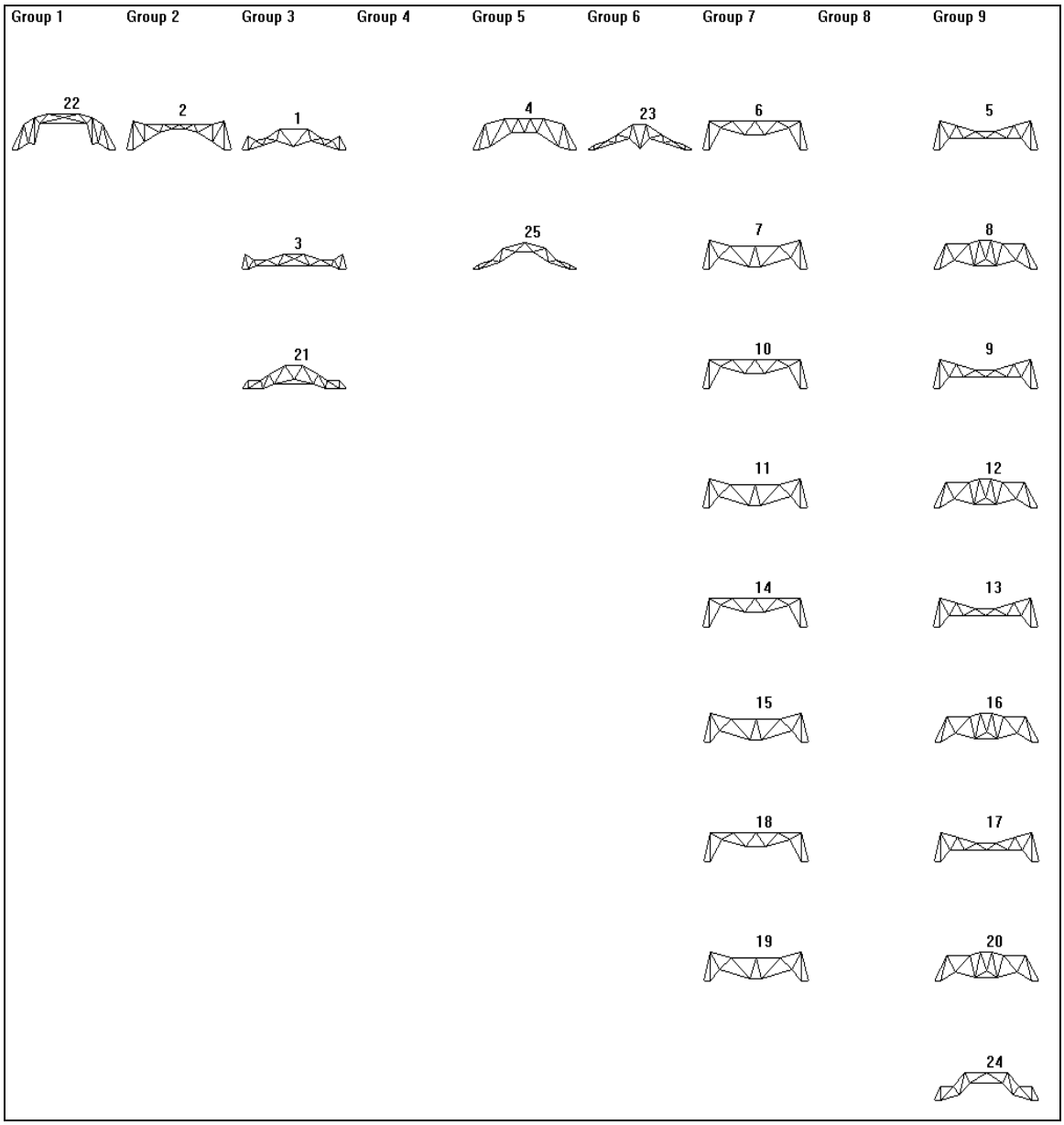


Figure B.1. Clustering map created for S25M35 using 2D KSOM.

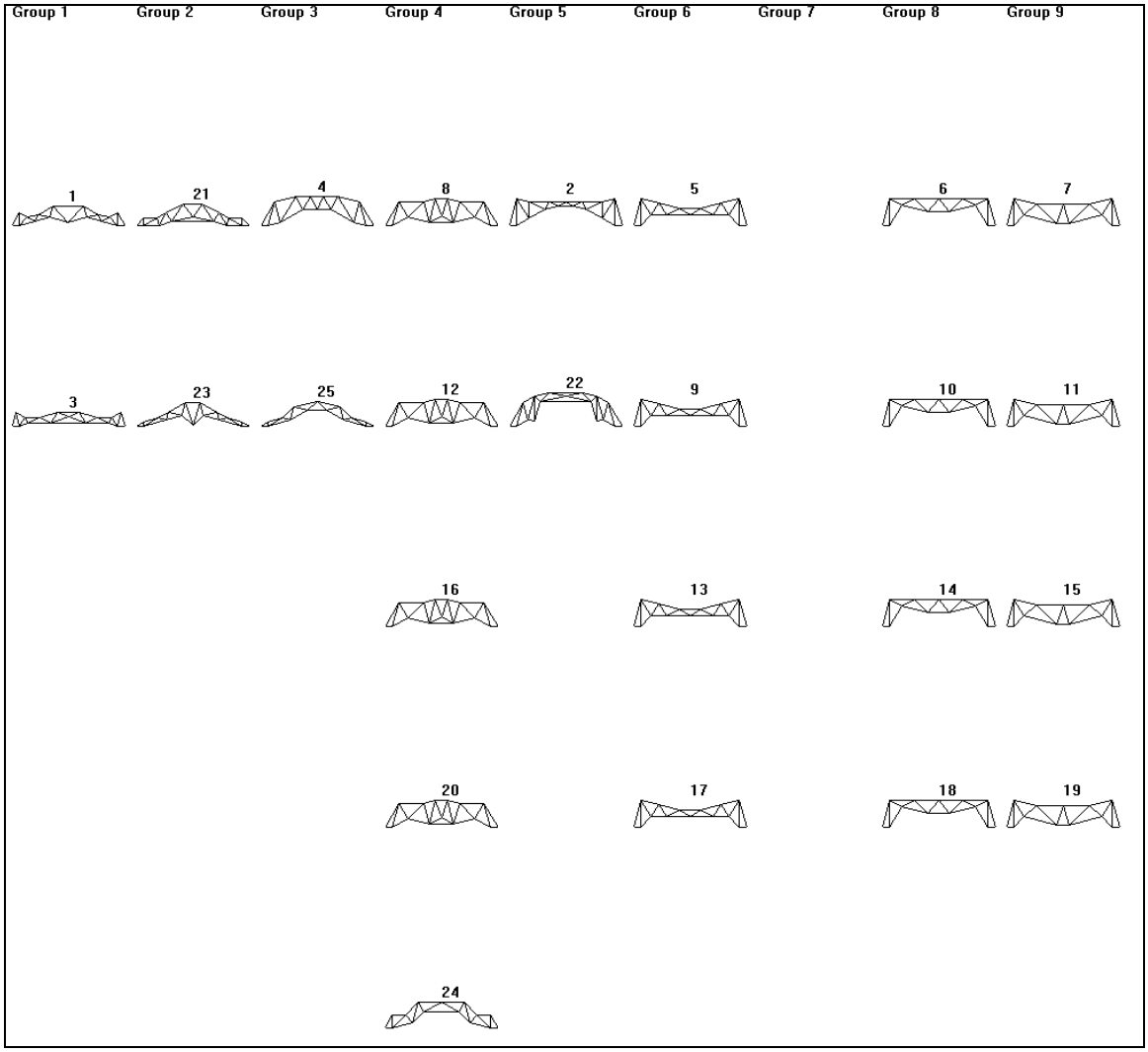


Figure B.2. Clustering map created for S25M35 using 1D KSOM.

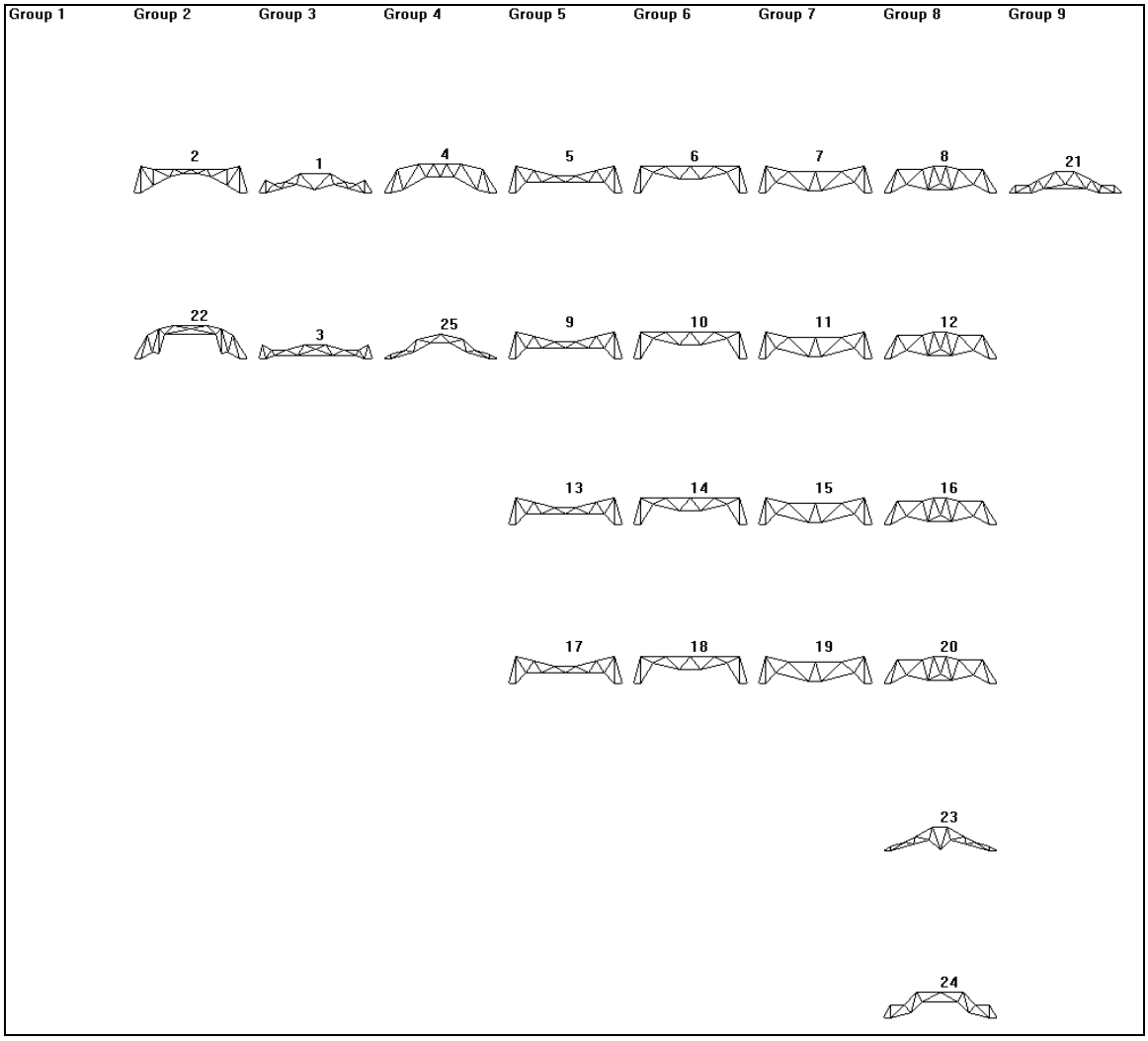


Figure B.3. Clustering map created for S25M35 using k-means.

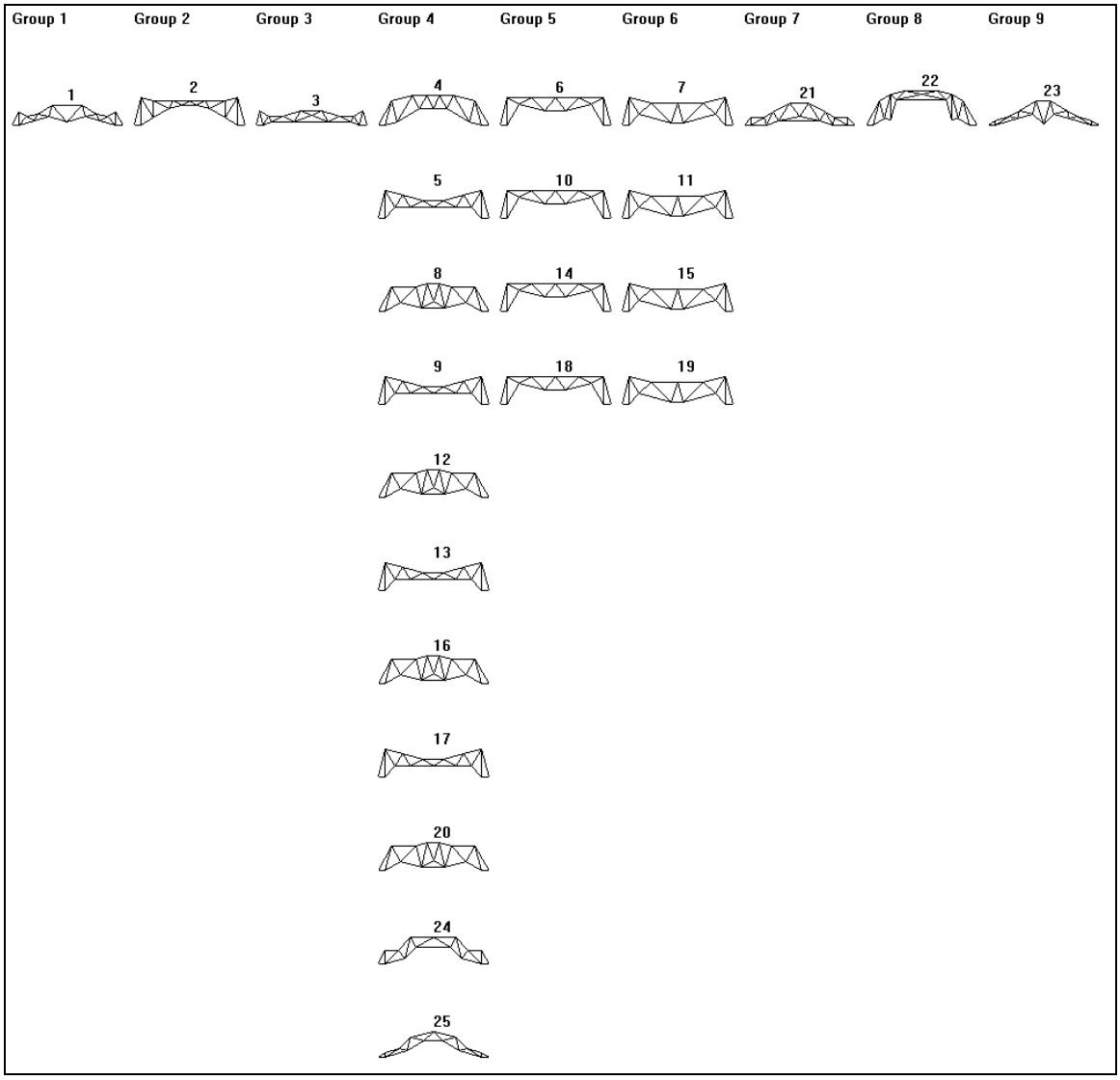


Figure B.4. Clustering map created for S25M35 using nearest neighbors.

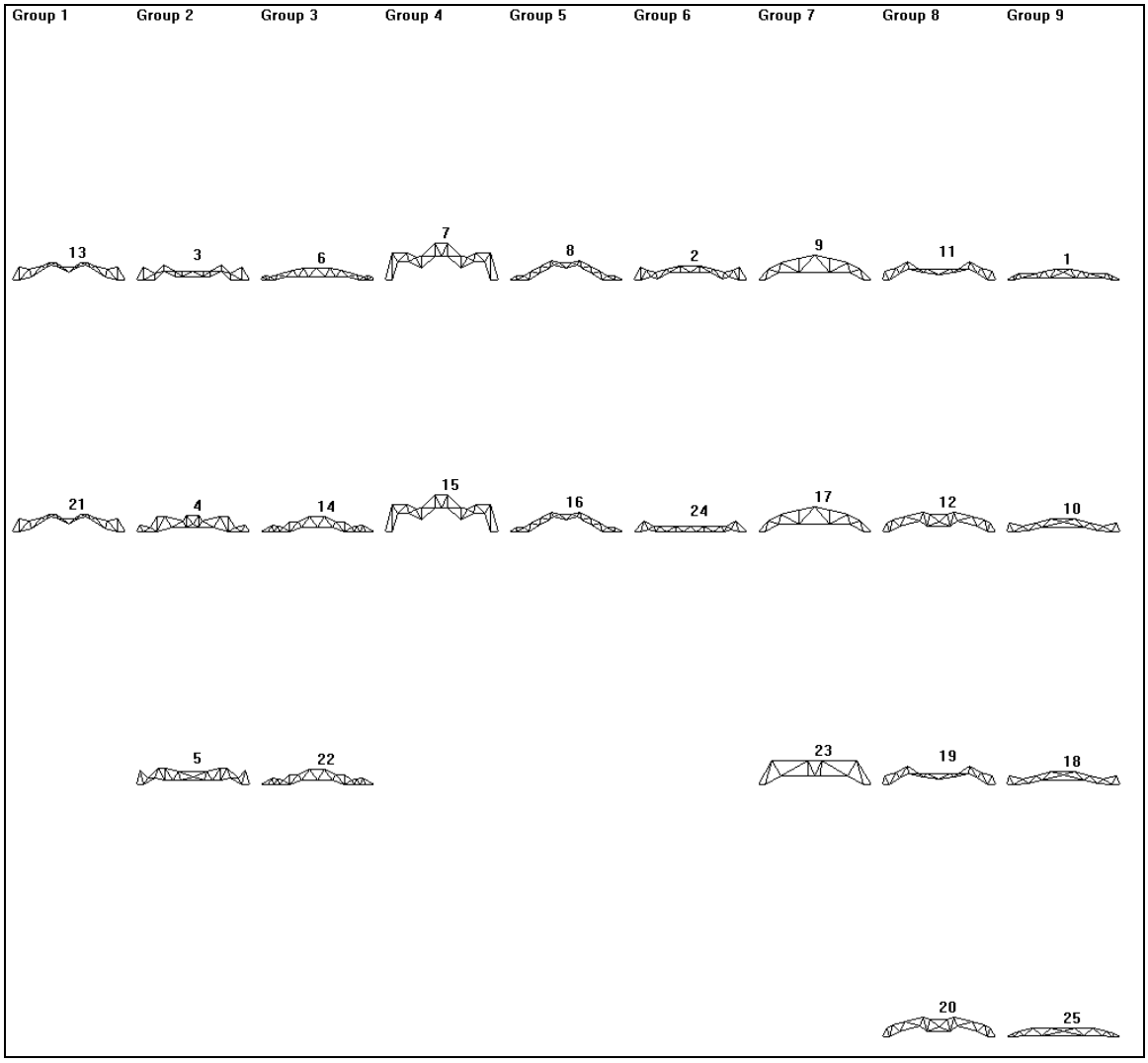


Figure B.5. Clustering map created for S25M50 using 2D KSOM.

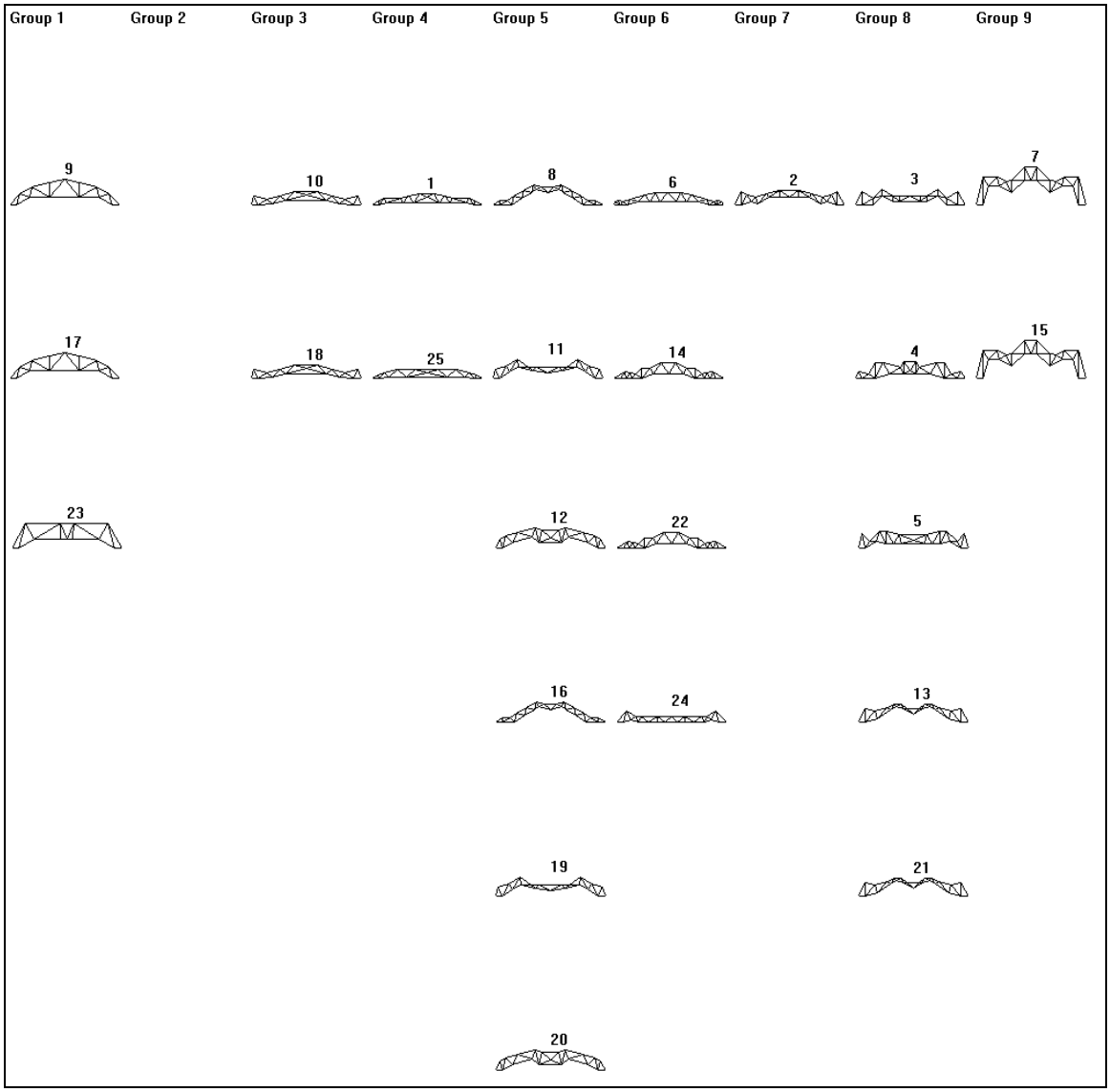


Figure B.6. Clustering map created for S25M50 using 1D KSOM.

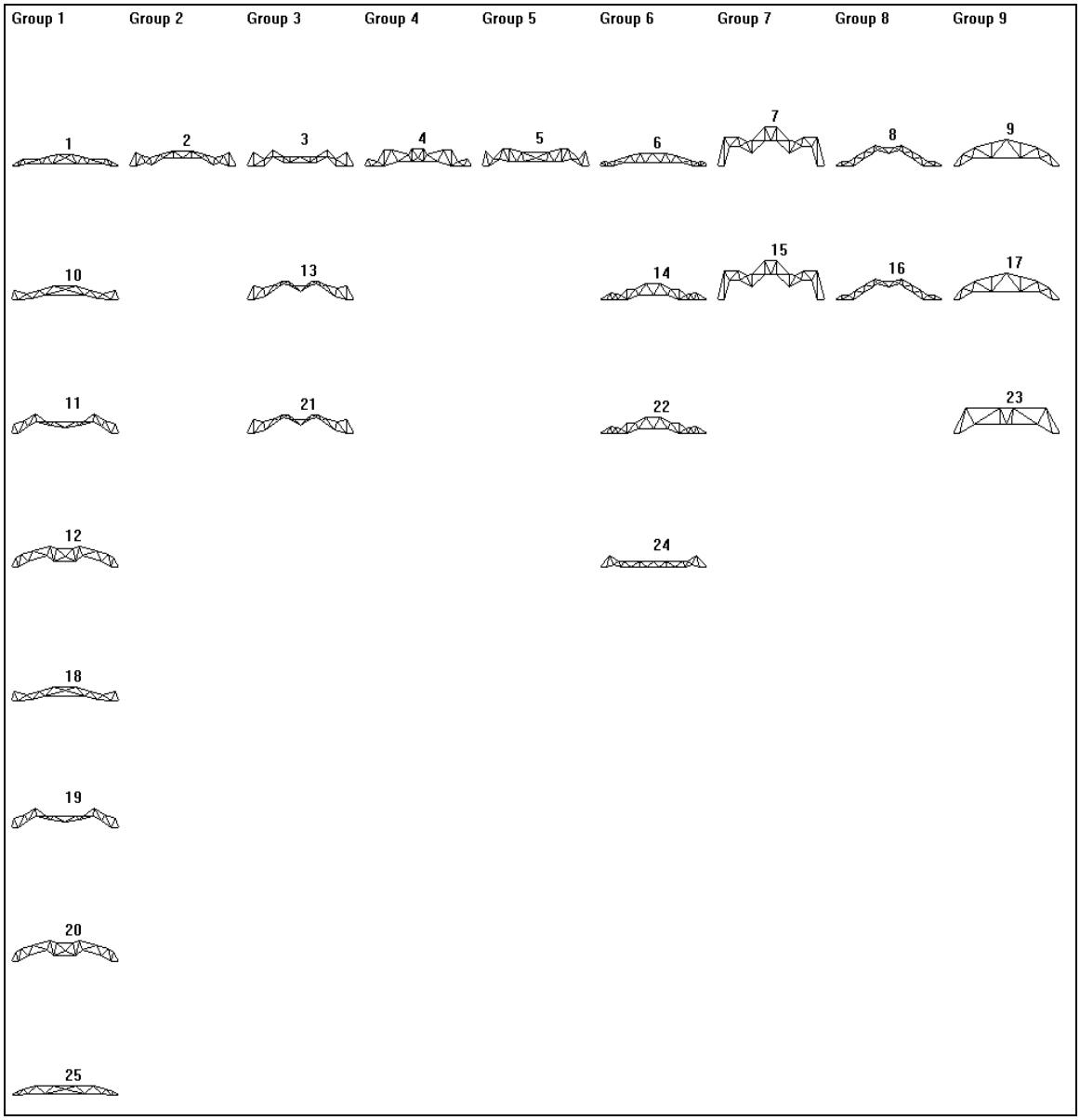


Figure B.7. Clustering map created for S25M50 using k-means.

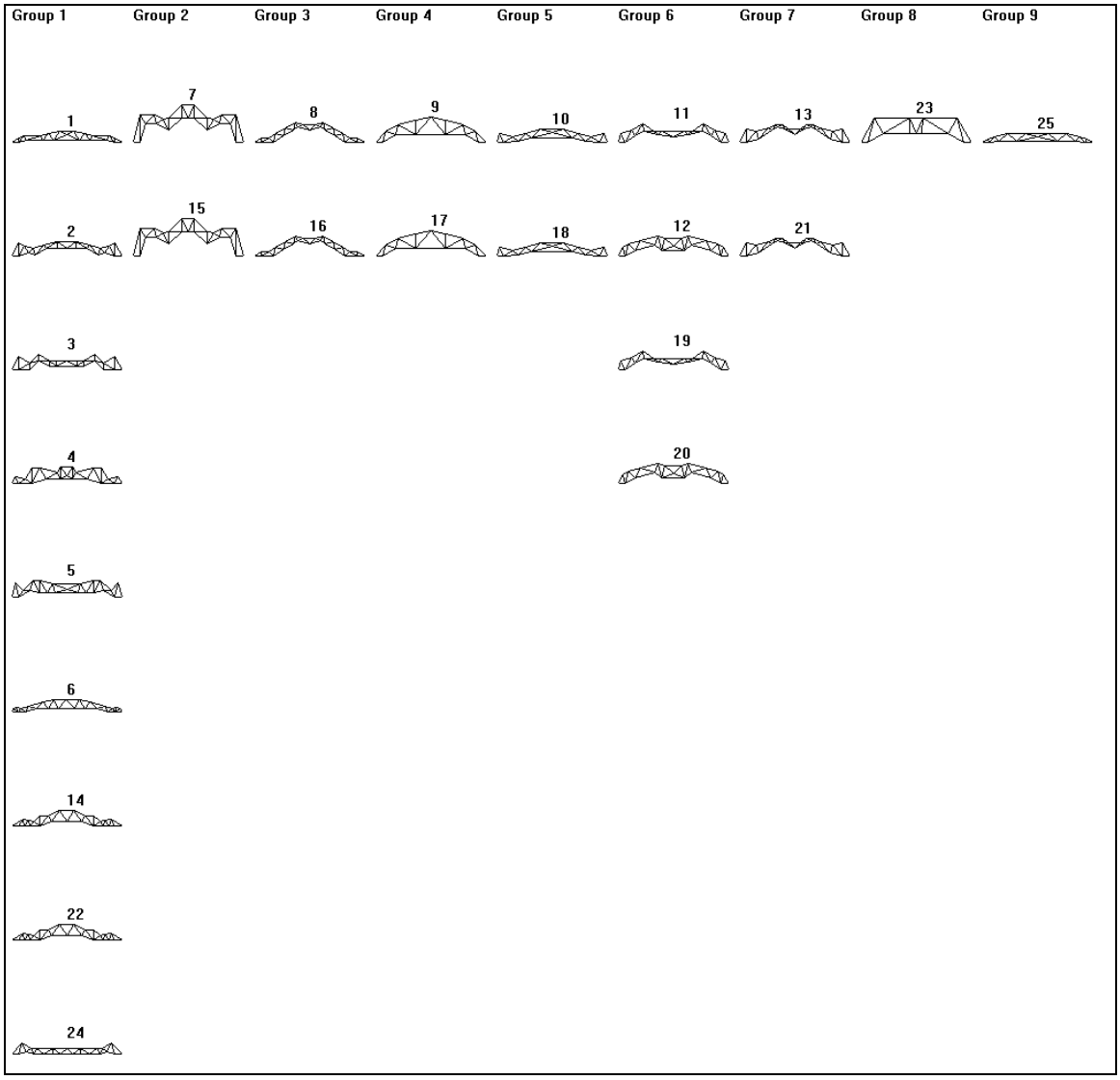


Figure B.8. Clustering map created for S25M50 using nearest neighbors.

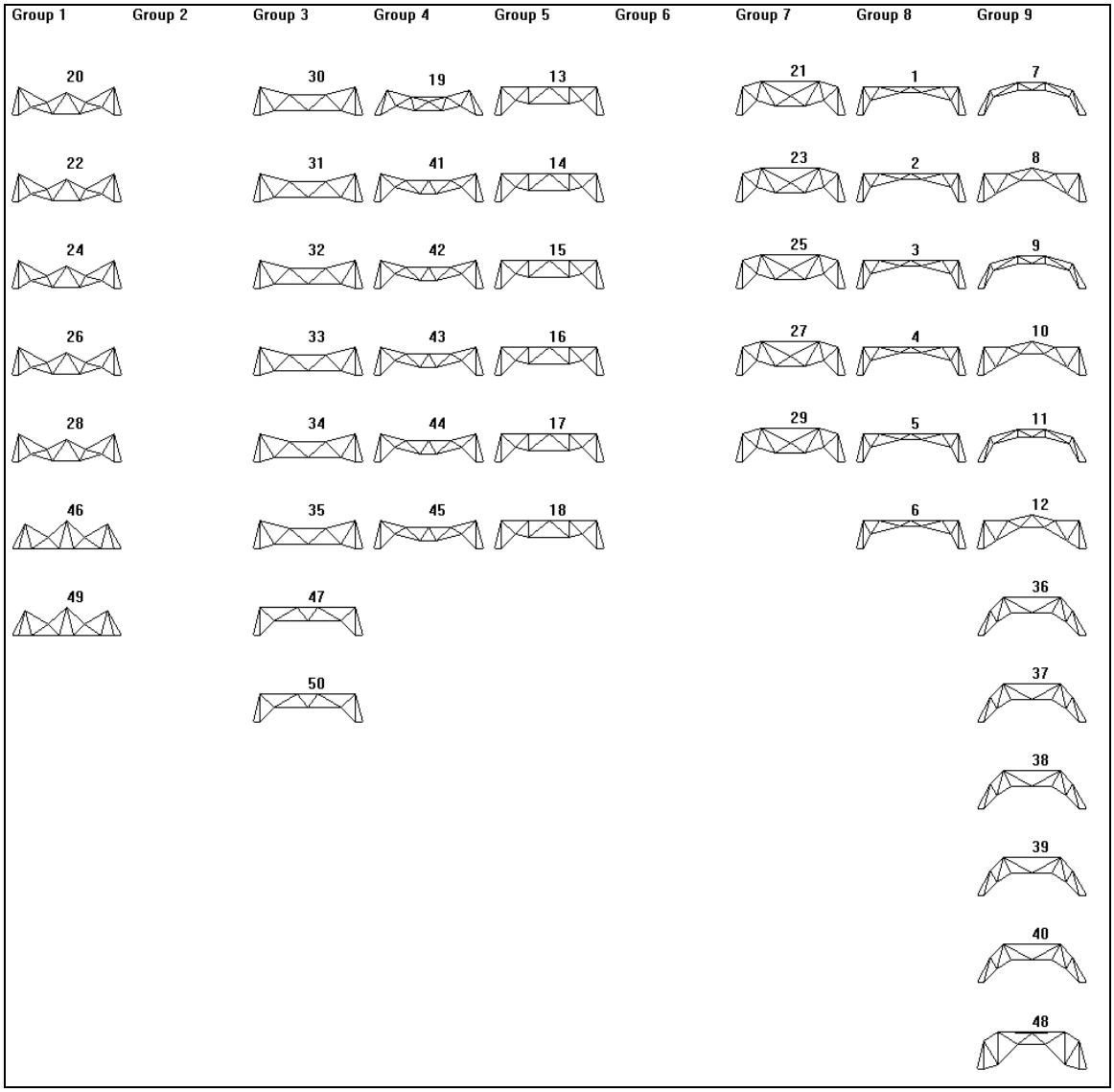


Figure B.9. Clustering map created for S50M25 using 2D KSOM.

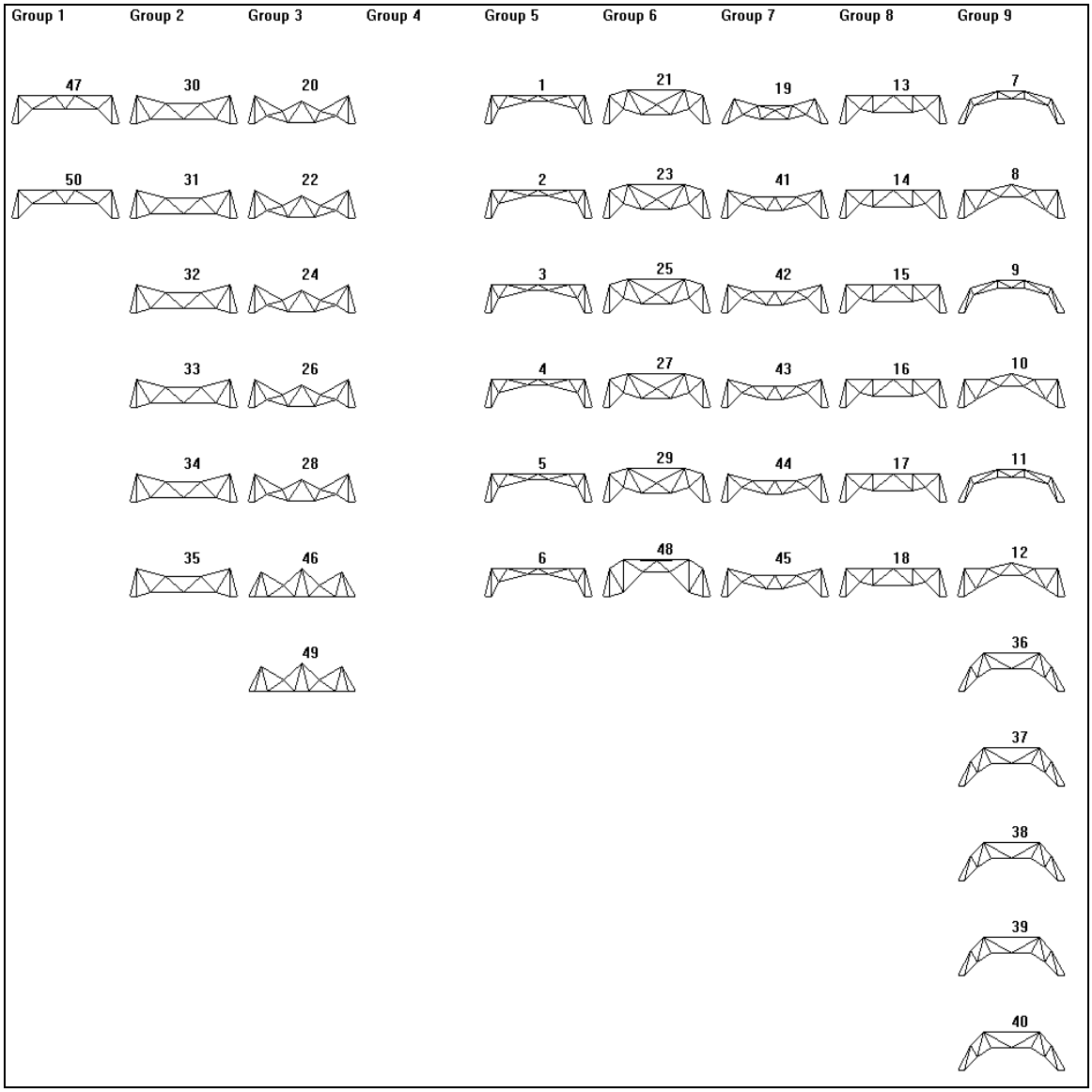


Figure B.10. Clustering map created for S50M25 using 1D KSOM.

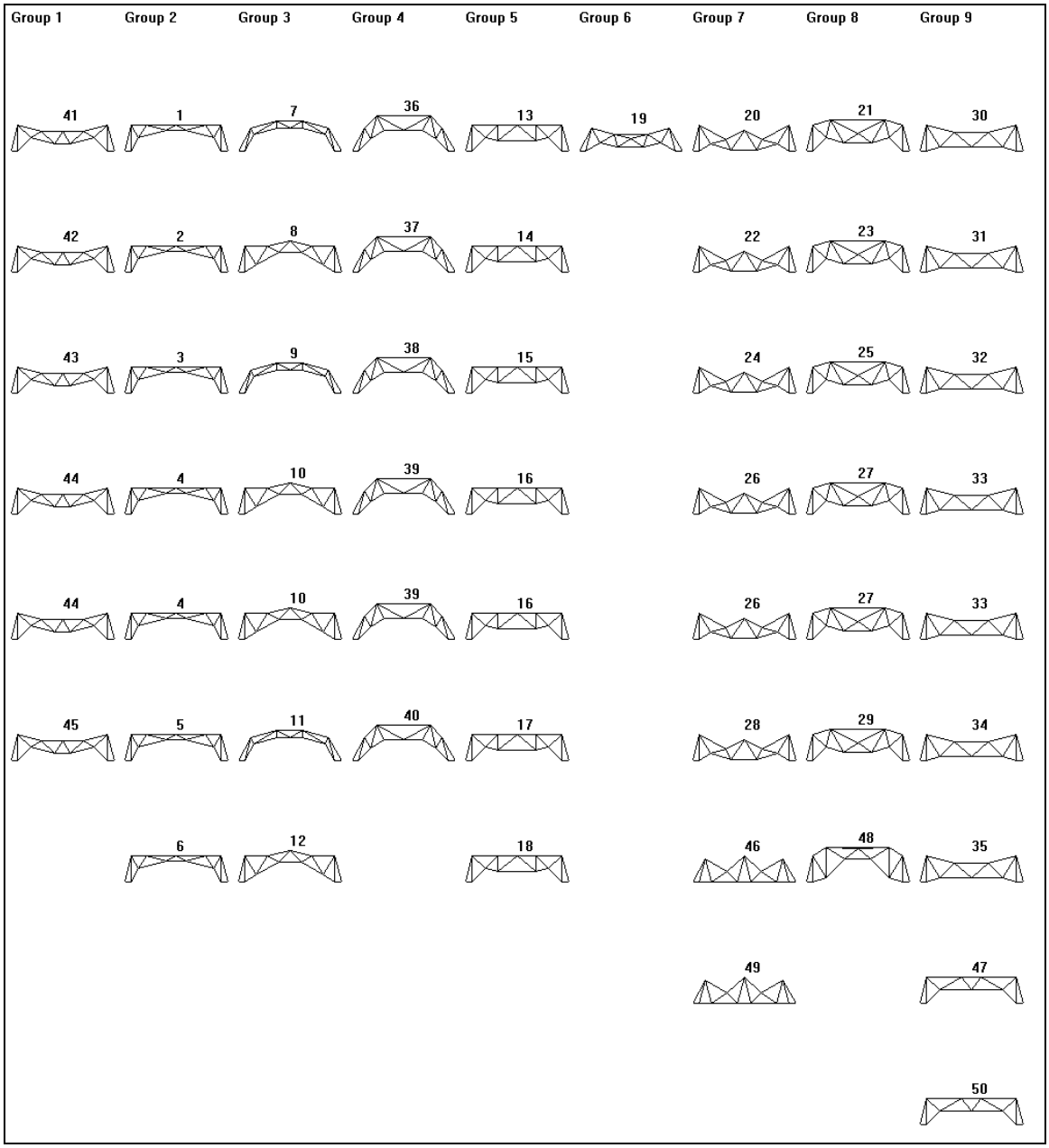


Figure B.11. Clustering map created for S50M25 using k-means.

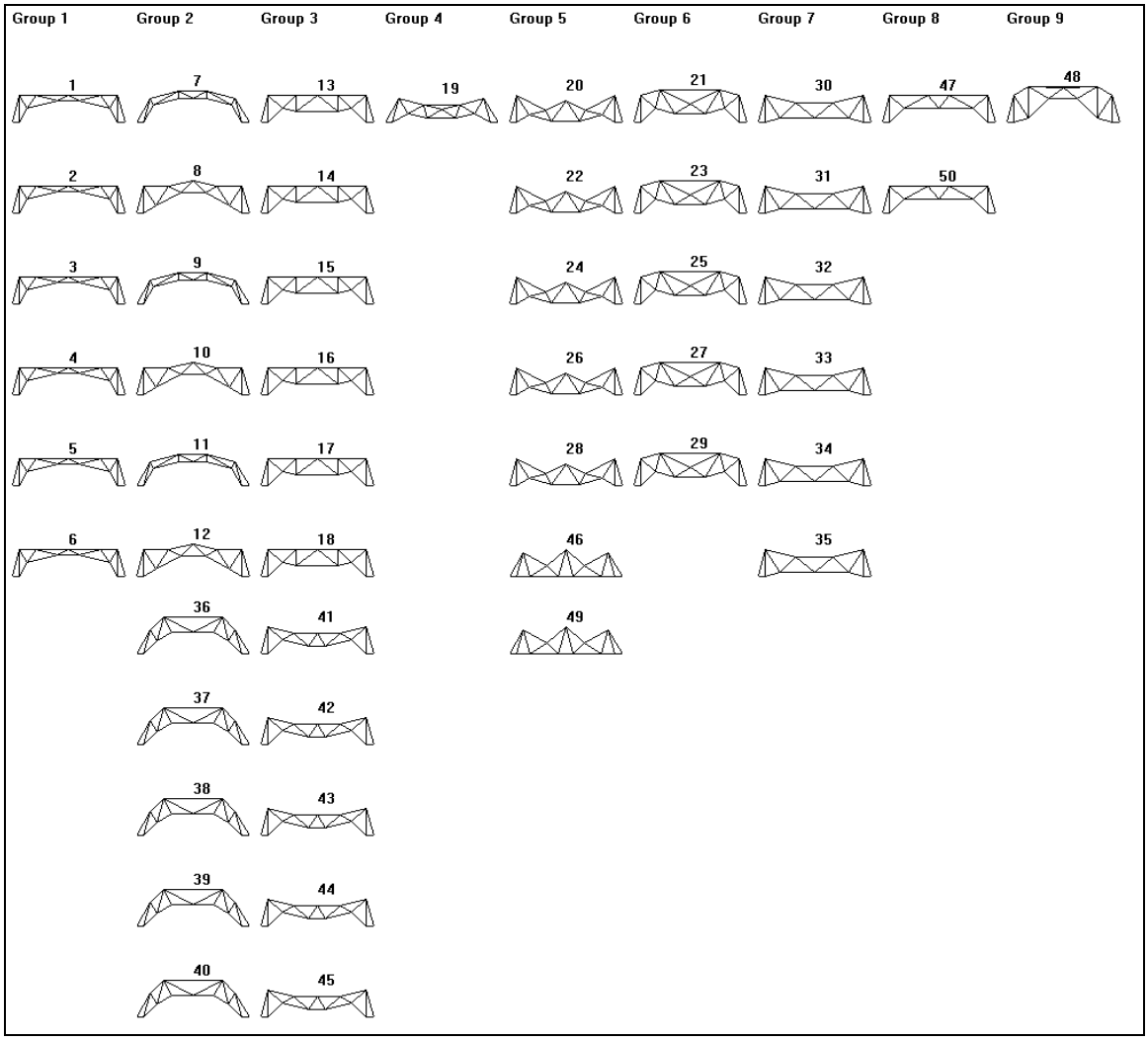


Figure B.12. Clustering map created for S50M25 using nearest neighbors.

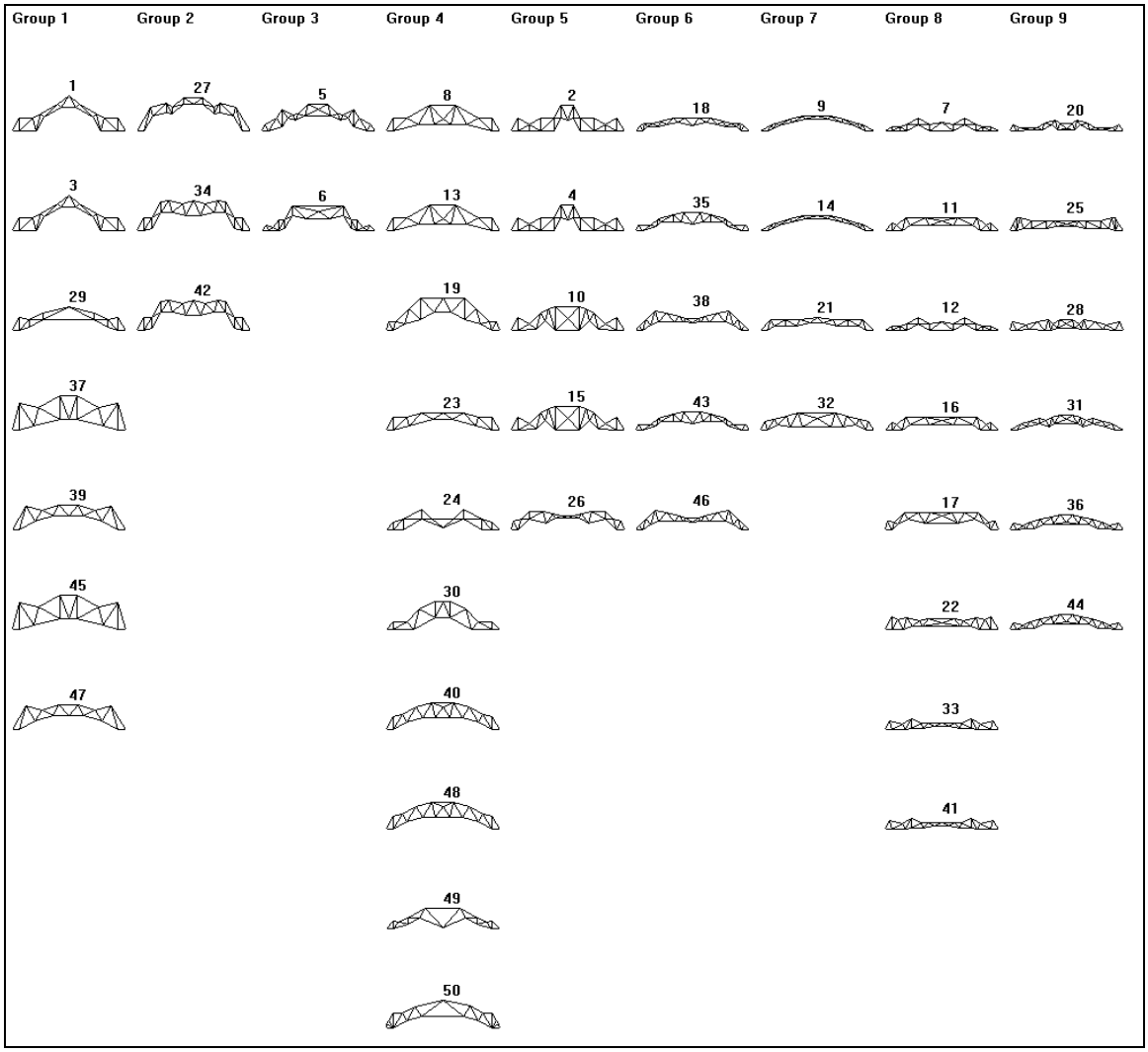


Figure B.13. Clustering map created for S50M50 using 2DKSOM.

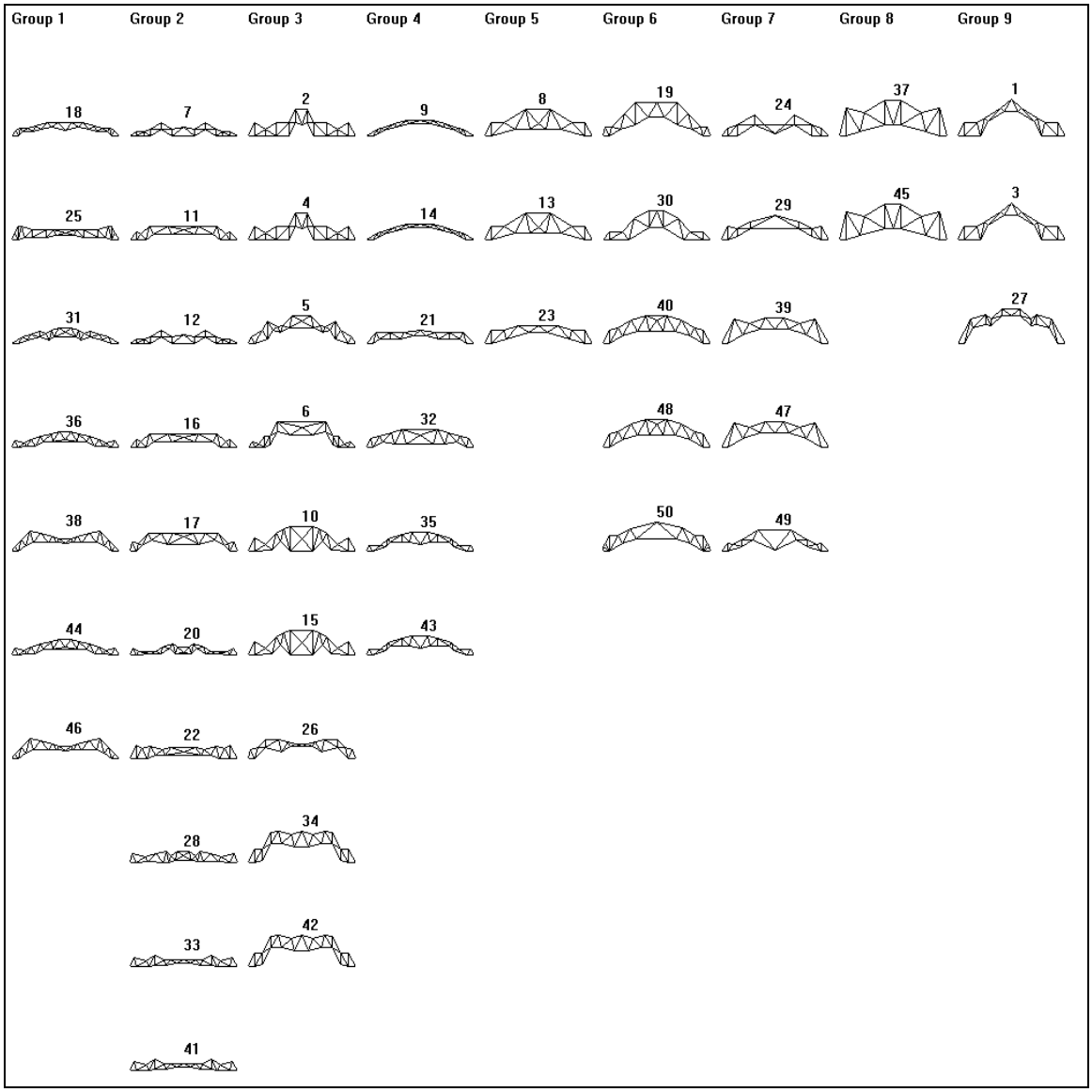


Figure B.14. Clustering map created for S50M50 using 1DKSOM.

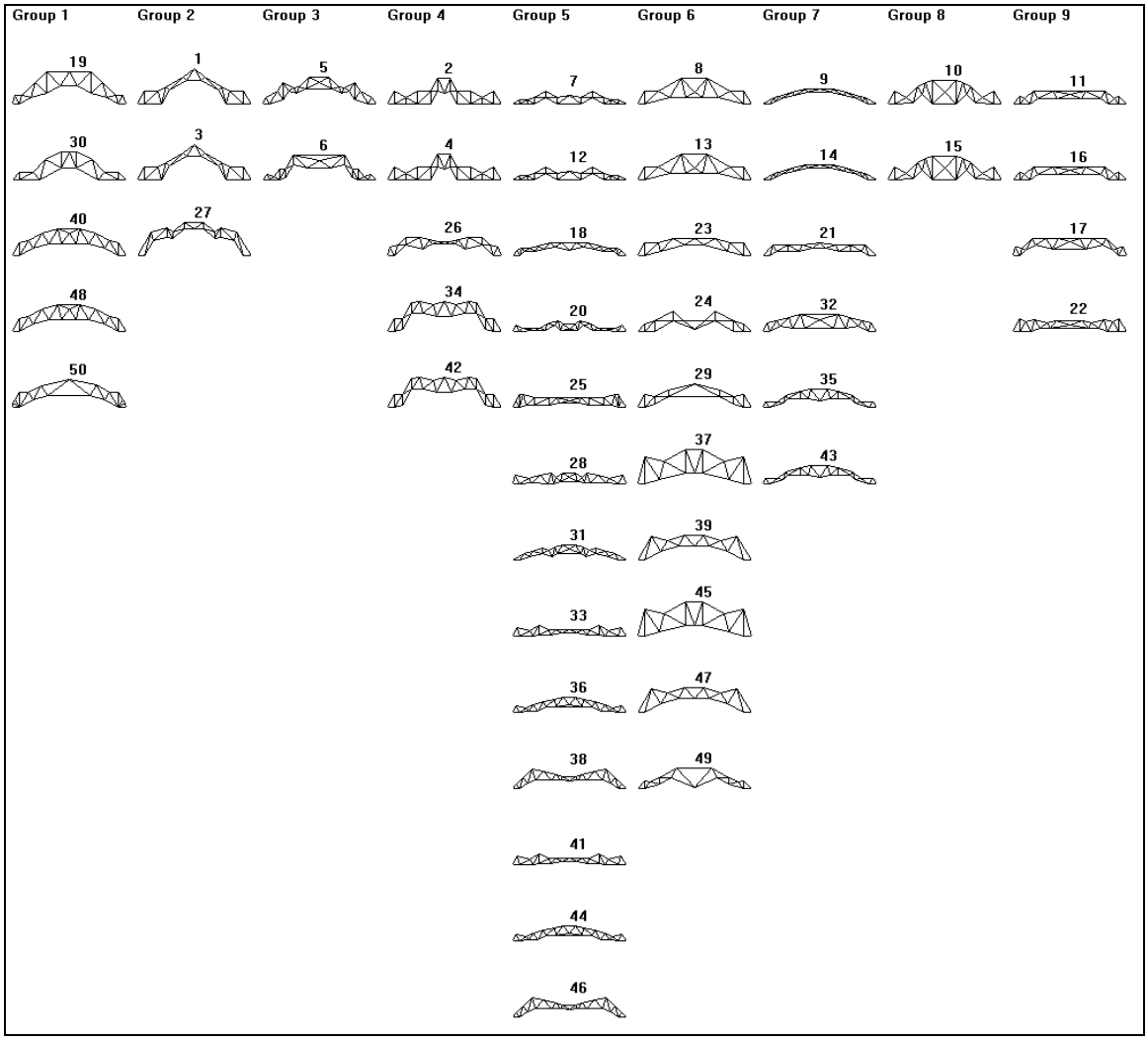


Figure B.15. Clustering map created for S50M50 using k-means.

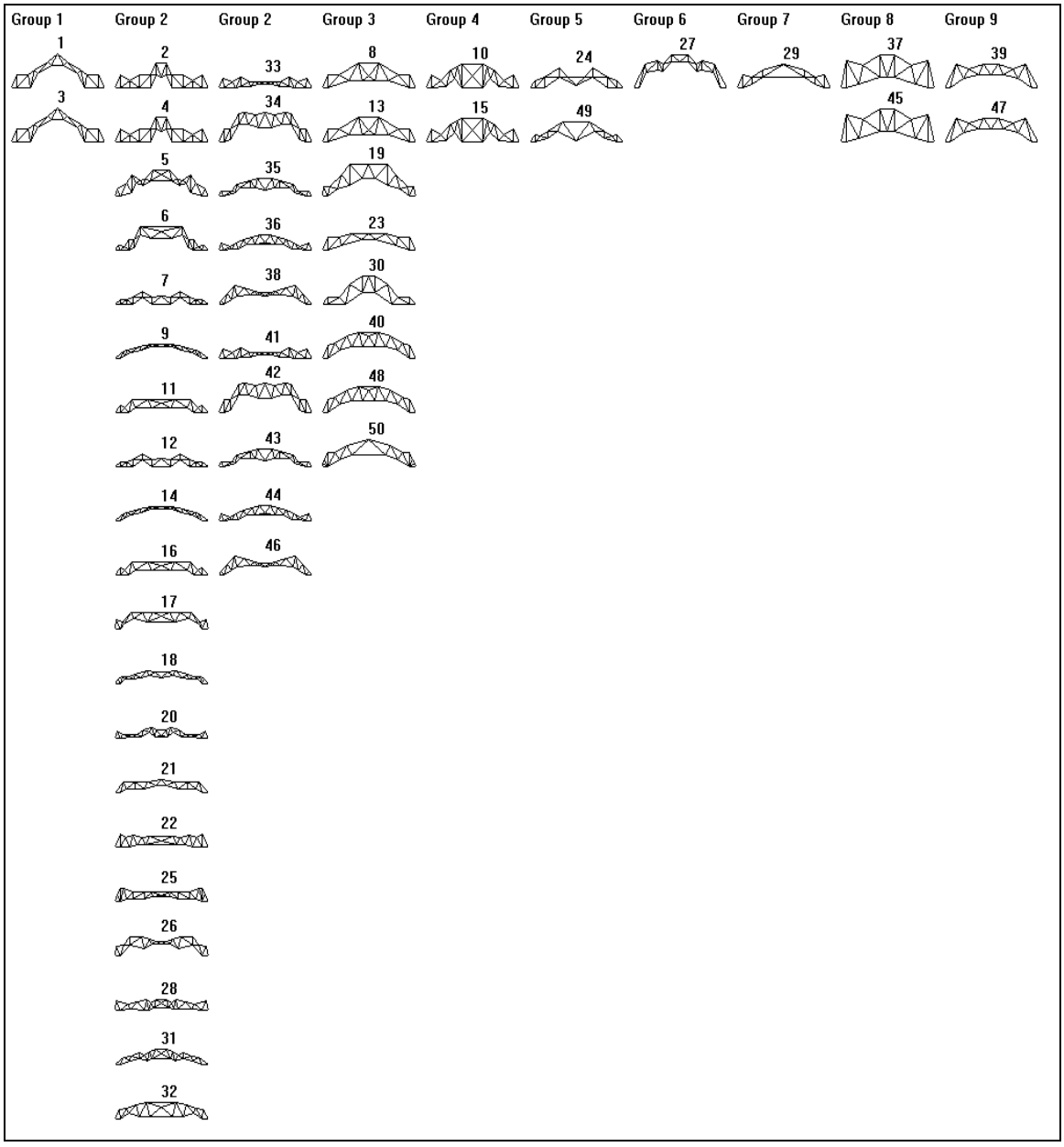


Figure B.16. Clustering map created for S50M50 using nearest neighbor.

Statistical Tables

The following tables present statistical properties for the populations. Tables of averages and standard deviations are shown first, with ranges following.

Table B.1. Averages and Standard Deviations for S25M35 Clusters.

Simulation / Feature		Group 1	Group 2	Group 3	Group 4	Group 5	Group 6	Group 7	Group 8	Group 9
2DKSOM	Joints	1.00 +/- 0.00	1.00 +/- 0.00	1.00 +/- 0.00	- - -	0.50 +/- 0.00	1.00 +/- 0.00	0.00 +/- 0.00	- - -	0.50 +/- 0.00
	Members	1.00 +/- 0.00	1.00 +/- 0.00	0.96 +/- 0.06	- - -	0.44 +/- 0.00	0.89 +/- 0.00	0.00 +/- 0.00	- - -	0.56 +/- 0.00
	Connectivity	1.00 +/- 0.00	1.00 +/- 0.00	0.83 +/- 0.30	- - -	0.27 +/- 0.00	0.48 +/- 0.00	0.00 +/- 0.00	- - -	0.86 +/- 0.00
	Height	1.00 +/- 0.00	0.67 +/- 0.00	0.22 +/- 0.21	- - -	0.65 +/- 0.17	0.49 +/- 0.00	0.67 +/- 0.00	- - -	0.65 +/- 0.03
	Clearance	1.00 +/- 0.00	0.78 +/- 0.00	0.12 +/- 0.02	- - -	0.65 +/- 0.01	0.00 +/- 0.00	0.31 +/- 0.26	- - -	0.31 +/- 0.19
	Depth	1.00 +/- 0.00	0.31 +/- 0.00	0.09 +/- 0.07	- - -	0.18 +/- 0.26	0.07 +/- 0.00	0.35 +/- 0.03	- - -	0.24 +/- 0.03
	Top Flat	0.61 +/- 0.00	0.49 +/- 0.00	0.08 +/- 0.10	- - -	0.35 +/- 0.13	0.05 +/- 0.00	0.86 +/- 0.15	- - -	0.52 +/- 0.21
	Top Directions	0.00 +/- 0.00	0.50 +/- 0.00	0.67 +/- 0.58	- - -	0.00 +/- 0.00	0.00 +/- 0.00	0.25 +/- 0.27	- - -	0.22 +/- 0.26
	Bottom Flat	0.75 +/- 0.00	0.48 +/- 0.00	0.16 +/- 0.24	- - -	0.55 +/- 0.01	0.71 +/- 0.00	0.82 +/- 0.19	- - -	0.41 +/- 0.10
1DKSOM	Joints	1.00 +/- 0.00	1.00 +/- 0.00	0.50 +/- 0.00	0.50 +/- 0.00	1.00 +/- 0.00	0.50 +/- 0.00	- - -	0.00 +/- 0.00	0.00 +/- 0.00
	Members	0.94 +/- 0.08	0.94 +/- 0.08	0.44 +/- 0.00	0.56 +/- 0.00	1.00 +/- 0.00	0.56 +/- 0.00	- - -	0.00 +/- 0.00	0.00 +/- 0.00
	Connectivity	0.74 +/- 0.37	0.74 +/- 0.37	0.27 +/- 0.00	0.86 +/- 0.00	1.00 +/- 0.00	0.86 +/- 0.00	- - -	0.00 +/- 0.00	0.00 +/- 0.00
	Height	0.13 +/- 0.19	0.45 +/- 0.06	0.65 +/- 0.17	0.64 +/- 0.03	0.84 +/- 0.23	0.67 +/- 0.00	- - -	0.67 +/- 0.00	0.67 +/- 0.00
	Clearance	0.11 +/- 0.02	0.07 +/- 0.10	0.65 +/- 0.01	0.23 +/- 0.23	0.89 +/- 0.16	0.40 +/- 0.00	- - -	0.55 +/- 0.00	0.06 +/- 0.00
	Depth	0.06 +/- 0.05	0.11 +/- 0.07	0.18 +/- 0.26	0.21 +/- 0.01	0.65 +/- 0.49	0.27 +/- 0.00	- - -	0.38 +/- 0.00	0.33 +/- 0.00
	Top Flat	0.11 +/- 0.11	0.02 +/- 0.03	0.35 +/- 0.13	0.34 +/- 0.03	0.55 +/- 0.09	0.74 +/- 0.00	- - -	0.71 +/- 0.00	1.00 +/- 0.00
	Top Directions	1.00 +/- 0.00	0.00 +/- 0.00	0.00 +/- 0.00	0.00 +/- 0.00	0.25 +/- 0.35	0.50 +/- 0.00	- - -	0.00 +/- 0.00	0.50 +/- 0.00
	Bottom Flat	0.21 +/- 0.30	0.37 +/- 0.47	0.55 +/- 0.01	0.49 +/- 0.04	0.61 +/- 0.19	0.31 +/- 0.00	- - -	1.00 +/- 0.00	0.64 +/- 0.00
K-Means	Joints	- - -	1.00 +/- 0.00	1.00 +/- 0.00	0.50 +/- 0.00	0.50 +/- 0.00	0.00 +/- 0.00	0.00 +/- 0.00	0.58 +/- 0.20	1.00 +/- 0.00
	Members	- - -	1.00 +/- 0.00	0.94 +/- 0.08	0.44 +/- 0.00	0.56 +/- 0.00	0.00 +/- 0.00	0.00 +/- 0.00	0.61 +/- 0.14	1.00 +/- 0.00
	Connectivity	- - -	1.00 +/- 0.00	0.74 +/- 0.37	0.27 +/- 0.00	0.86 +/- 0.00	0.00 +/- 0.00	0.00 +/- 0.00	0.80 +/- 0.16	1.00 +/- 0.00
	Height	- - -	0.84 +/- 0.23	0.13 +/- 0.19	0.65 +/- 0.17	0.67 +/- 0.00	0.67 +/- 0.00	0.67 +/- 0.00	0.61 +/- 0.07	0.41 +/- 0.00
	Clearance	- - -	0.89 +/- 0.16	0.11 +/- 0.02	0.65 +/- 0.01	0.40 +/- 0.00	0.55 +/- 0.00	0.06 +/- 0.00	0.19 +/- 0.23	0.14 +/- 0.00
	Depth	- - -	0.65 +/- 0.49	0.06 +/- 0.05	0.18 +/- 0.26	0.27 +/- 0.00	0.38 +/- 0.00	0.33 +/- 0.00	0.19 +/- 0.06	0.16 +/- 0.00
	Top Flat	- - -	0.55 +/- 0.09	0.11 +/- 0.11	0.35 +/- 0.13	0.74 +/- 0.00	0.71 +/- 0.00	1.00 +/- 0.00	0.29 +/- 0.12	0.00 +/- 0.00
	Top Directions	- - -	0.25 +/- 0.35	1.00 +/- 0.00	0.00 +/- 0.00	0.50 +/- 0.00	0.00 +/- 0.00	0.50 +/- 0.00	0.00 +/- 0.00	0.00 +/- 0.00
	Bottom Flat	- - -	0.61 +/- 0.19	0.21 +/- 0.30	0.55 +/- 0.01	0.31 +/- 0.00	1.00 +/- 0.00	0.64 +/- 0.00	0.52 +/- 0.10	0.04 +/- 0.00
Nearest Neighbor	Joints	1.00 +/- 0.00	1.00 +/- 0.00	1.00 +/- 0.00	0.50 +/- 0.00	0.00 +/- 0.00	0.00 +/- 0.00	1.00 +/- 0.00	1.00 +/- 0.00	1.00 +/- 0.00
	Members	0.89 +/- 0.00	1.00 +/- 0.00	1.00 +/- 0.00	0.54 +/- 0.04	0.00 +/- 0.00	0.00 +/- 0.00	1.00 +/- 0.00	1.00 +/- 0.00	0.89 +/- 0.00
	Connectivity	0.48 +/- 0.00	1.00 +/- 0.00	1.00 +/- 0.00	0.75 +/- 0.24	0.00 +/- 0.00	0.00 +/- 0.00	1.00 +/- 0.00	1.00 +/- 0.00	0.48 +/- 0.00
	Height	0.26 +/- 0.00	0.67 +/- 0.00	0.00 +/- 0.00	0.65 +/- 0.06	0.67 +/- 0.00	0.67 +/- 0.00	0.41 +/- 0.00	1.00 +/- 0.00	0.49 +/- 0.00
	Clearance	0.12 +/- 0.00	0.78 +/- 0.00	0.09 +/- 0.00	0.37 +/- 0.22	0.55 +/- 0.00	0.06 +/- 0.00	0.14 +/- 0.00	1.00 +/- 0.00	0.00 +/- 0.00
	Depth	0.09 +/- 0.00	0.31 +/- 0.00	0.02 +/- 0.00	0.23 +/- 0.09	0.38 +/- 0.00	0.33 +/- 0.00	0.16 +/- 0.00	1.00 +/- 0.00	0.07 +/- 0.00
	Top Flat	0.19 +/- 0.00	0.49 +/- 0.00	0.04 +/- 0.00	0.49 +/- 0.21	0.71 +/- 0.00	1.00 +/- 0.00	0.00 +/- 0.00	0.61 +/- 0.00	0.05 +/- 0.00
	Top Directions	1.00 +/- 0.00	0.50 +/- 0.00	1.00 +/- 0.00	0.18 +/- 0.25	0.00 +/- 0.00	0.50 +/- 0.00	0.00 +/- 0.00	0.00 +/- 0.00	0.00 +/- 0.00
	Bottom Flat	0.43 +/- 0.00	0.48 +/- 0.00	0.00 +/- 0.00	0.43 +/- 0.10	1.00 +/- 0.00	0.64 +/- 0.00	0.04 +/- 0.00	0.75 +/- 0.00	0.71 +/- 0.00

Table B.2. Maximum and Minimum Values for S25M35 Clusters.

S25M35		Group 1		Group 2		Group 3		Group 4		Group 5		Group 6		Group 7		Group 8		Group 9	
Simulation / Feature		**Max**	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**	**Min**
2DKSOM	Joints	1.00	1.00	1.00	1.00	1.00	1.00	-	-	0.50	0.50	1.00	1.00	0.00	0.00	-	-	0.50	0.50
	Members	1.00	1.00	1.00	1.00	1.00	0.89	-	-	0.44	0.44	0.89	0.89	0.00	0.00	-	-	0.56	0.56
	Connectivity	1.00	1.00	1.00	1.00	1.00	0.48	-	-	0.27	0.27	0.48	0.48	0.00	0.00	-	-	0.86	0.86
	Height	1.00	1.00	0.67	0.67	0.41	0.00	-	-	0.77	0.53	0.49	0.49	0.67	0.67	-	-	0.67	0.58
	Clearance	1.00	1.00	0.78	0.78	0.14	0.09	-	-	0.65	0.64	0.00	0.00	0.55	0.06	-	-	0.65	0.13
	Depth	1.00	1.00	0.31	0.31	0.16	0.02	-	-	0.37	0.00	0.07	0.07	0.38	0.33	-	-	0.27	0.19
	Top Flat	0.61	0.61	0.49	0.49	0.19	0.00	-	-	0.44	0.25	0.05	0.05	1.00	0.71	-	-	0.74	0.29
	Top Directions	0.00	0.00	0.50	0.50	1.00	0.00	-	-	0.00	0.00	0.00	0.00	0.50	0.00	-	-	0.50	0.00
	Bottom Flat	0.75	0.75	0.48	0.48	0.43	0.00	-	-	0.55	0.54	0.71	0.71	1.00	0.64	-	-	0.55	0.31
1DKSOM	Joints	1.00	1.00	1.00	1.00	0.50	0.50	0.50	0.50	1.00	1.00	0.50	0.50	-	-	0.00	0.00	0.00	0.00
	Members	1.00	0.89	1.00	0.89	0.44	0.44	0.56	0.56	1.00	1.00	0.56	0.56	-	-	0.00	0.00	0.00	0.00
	Connectivity	1.00	0.48	1.00	0.48	0.27	0.27	0.86	0.86	1.00	1.00	0.86	0.86	-	-	0.00	0.00	0.00	0.00
	Height	0.26	0.00	0.49	0.41	0.77	0.53	0.65	0.58	1.00	0.67	0.67	0.67	-	-	0.67	0.67	0.67	0.67
	Clearance	0.12	0.09	0.14	0.00	0.65	0.64	0.65	0.13	1.00	0.78	0.40	0.40	-	-	0.55	0.55	0.06	0.06
	Depth	0.09	0.02	0.16	0.07	0.37	0.00	0.21	0.19	1.00	0.31	0.27	0.27	-	-	0.38	0.38	0.33	0.33
	Top Flat	0.19	0.04	0.05	0.00	0.44	0.25	0.35	0.29	0.61	0.49	0.74	0.74	-	-	0.71	0.71	1.00	1.00
	Top Directions	1.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.50	0.00	0.50	0.50	-	-	0.00	0.00	0.50	0.50
	Bottom Flat	0.43	0.00	0.71	0.04	0.55	0.54	0.55	0.47	0.75	0.48	0.31	0.31	-	-	1.00	1.00	0.64	0.64
K-Means	Joints	-	-	1.00	1.00	1.00	1.00	0.50	0.50	0.50	0.50	0.00	0.00	0.00	0.00	1.00	0.50	1.00	1.00
	Members	-	-	1.00	1.00	1.00	0.89	0.44	0.44	0.56	0.56	0.00	0.00	0.00	0.00	0.89	0.56	1.00	1.00
	Connectivity	-	-	1.00	1.00	1.00	0.48	0.27	0.27	0.86	0.86	0.00	0.00	0.00	0.00	0.86	0.48	1.00	1.00
	Height	-	-	1.00	0.67	0.26	0.00	0.77	0.53	0.67	0.67	0.67	0.67	0.67	0.67	0.65	0.49	0.41	0.41
	Clearance	-	-	1.00	0.78	0.12	0.09	0.65	0.64	0.40	0.40	0.55	0.55	0.06	0.06	0.65	0.00	0.14	0.14
	Depth	-	-	1.00	0.31	0.09	0.02	0.37	0.00	0.27	0.27	0.38	0.38	0.33	0.33	0.21	0.07	0.16	0.16
	Top Flat	-	-	0.61	0.49	0.19	0.04	0.44	0.25	0.74	0.74	0.71	0.71	1.00	1.00	0.35	0.05	0.00	0.00
	Top Directions	-	-	0.50	0.00	1.00	1.00	0.00	0.00	0.50	0.50	0.00	0.00	0.50	0.50	0.00	0.00	0.00	0.00
	Bottom Flat	-	-	0.75	0.48	0.43	0.00	0.55	0.54	0.31	0.31	1.00	1.00	0.64	0.64	0.71	0.47	0.04	0.04
Nearest	Joints	1.00	1.00	1.00	1.00	1.00	1.00	0.50	0.50	0.00	0.00	0.00	0.00	1.00	1.00	1.00	1.00	1.00	1.00
Neighbor	Members	0.89	0.89	1.00	1.00	1.00	1.00	0.56	0.44	0.00	0.00	0.00	0.00	1.00	1.00	1.00	1.00	0.89	0.89
	Connectivity	0.48	0.48	1.00	1.00	1.00	1.00	0.86	0.27	0.00	0.00	0.00	0.00	1.00	1.00	1.00	1.00	0.48	0.48
	Height	0.26	0.26	0.67	0.67	0.00	0.00	0.77	0.53	0.67	0.67	0.67	0.67	0.41	0.41	1.00	1.00	0.49	0.49
	Clearance	0.12	0.12	0.78	0.78	0.09	0.09	0.65	0.13	0.55	0.55	0.06	0.06	0.14	0.14	1.00	1.00	0.00	0.00
	Depth	0.09	0.09	0.31	0.31	0.02	0.02	0.37	0.00	0.38	0.38	0.33	0.33	0.16	0.16	1.00	1.00	0.07	0.07
	Top Flat	0.19	0.19	0.49	0.49	0.04	0.04	0.74	0.25	0.71	0.71	1.00	1.00	0.00	0.00	0.61	0.61	0.05	0.05
	Top Directions	1.00	1.00	0.50	0.50	1.00	1.00	0.50	0.00	0.00	0.00	0.50	0.50	0.00	0.00	0.00	0.00	0.00	0.00
	Bottom Flat	0.43	0.43	0.48	0.48	0.00	0.00	0.55	0.31	1.00	1.00	0.64	0.64	0.04	0.04	0.75	0.75	0.71	0.71

Table B.3. Averages and Standard Deviations for S25M50 Clusters.

Simulation / Feature		Group 1	Group 2	Group 3	Group 4	Group 5	Group 6	Group 7	Group 8	Group 9
2DKSOM	Joints	1.00 +/- 0.00	0.83 +/- 0.00	1.00 +/- 0.00	0.83 +/- 0.00	1.00 +/- 0.00	0.83 +/- 0.00	0.11 +/- 0.10	0.67 +/- 0.00	0.42 +/- 0.17
	Members	0.96 +/- 0.00	0.83 +/- 0.02	1.00 +/- 0.00	0.80 +/- 0.00	0.96 +/- 0.00	0.84 +/- 0.00	0.11 +/- 0.09	0.66 +/- 0.02	0.44 +/- 0.16
	Connectivity	0.73 +/- 0.00	0.86 +/- 0.17	1.00 +/- 0.00	0.67 +/- 0.00	0.73 +/- 0.00	0.95 +/- 0.00	0.14 +/- 0.12	0.74 +/- 0.18	0.79 +/- 0.07
	Height	0.31 +/- 0.00	0.26 +/- 0.03	0.19 +/- 0.07	1.00 +/- 0.00	0.38 +/- 0.00	0.11 +/- 0.09	0.56 +/- 0.02	0.36 +/- 0.04	0.08 +/- 0.06
	Clearance	0.31 +/- 0.00	0.14 +/- 0.04	0.14 +/- 0.01	1.00 +/- 0.00	0.48 +/- 0.00	0.16 +/- 0.23	0.73 +/- 0.32	0.19 +/- 0.00	0.10 +/- 0.07
	Depth	0.14 +/- 0.00	0.20 +/- 0.04	0.11 +/- 0.04	1.00 +/- 0.00	0.20 +/- 0.00	0.09 +/- 0.07	0.15 +/- 0.08	0.12 +/- 0.01	0.10 +/- 0.08
	Top Flat	0.36 +/- 0.00	0.57 +/- 0.12	0.15 +/- 0.10	0.98 +/- 0.00	0.22 +/- 0.00	0.30 +/- 0.11	0.81 +/- 0.16	0.34 +/- 0.05	0.18 +/- 0.20
	Top Directions	0.75 +/- 0.00	0.83 +/- 0.14	0.50 +/- 0.00	0.50 +/- 0.00	0.25 +/- 0.00	0.38 +/- 0.18	0.00 +/- 0.00	0.25 +/- 0.00	0.25 +/- 0.29
	Bottom Flat	0.54 +/- 0.00	0.25 +/- 0.13	0.08 +/- 0.01	1.00 +/- 0.00	0.45 +/- 0.00	0.18 +/- 0.14	0.34 +/- 0.03	0.55 +/- 0.00	0.10 +/- 0.08
1DKSOM	Joints	0.11 +/- 0.10	- - -	0.33 +/- 0.00	0.50 +/- 0.24	0.78 +/- 0.17	0.96 +/- 0.08	0.83 +/- 0.00	0.90 +/- 0.09	0.83 +/- 0.00
	Members	0.11 +/- 0.09	- - -	0.36 +/- 0.00	0.52 +/- 0.23	0.76 +/- 0.16	0.96 +/- 0.08	0.84 +/- 0.00	0.88 +/- 0.07	0.80 +/- 0.00
	Connectivity	0.14 +/- 0.12	- - -	0.75 +/- 0.00	0.82 +/- 0.11	0.74 +/- 0.14	0.99 +/- 0.02	0.95 +/- 0.00	0.81 +/- 0.14	0.67 +/- 0.00
	Height	0.56 +/- 0.02	- - -	0.13 +/- 0.00	0.03 +/- 0.05	0.36 +/- 0.03	0.15 +/- 0.09	0.17 +/- 0.00	0.28 +/- 0.04	1.00 +/- 0.00
	Clearance	0.73 +/- 0.32	- - -	0.15 +/- 0.00	0.05 +/- 0.06	0.29 +/- 0.15	0.10 +/- 0.07	0.32 +/- 0.00	0.21 +/- 0.10	1.00 +/- 0.00
	Depth	0.15 +/- 0.08	- - -	0.16 +/- 0.00	0.04 +/- 0.05	0.15 +/- 0.04	0.09 +/- 0.05	0.14 +/- 0.00	0.17 +/- 0.04	1.00 +/- 0.00
	Top Flat	0.81 +/- 0.16	- - -	0.35 +/- 0.00	0.00 +/- 0.00	0.30 +/- 0.07	0.17 +/- 0.09	0.38 +/- 0.00	0.49 +/- 0.14	0.98 +/- 0.00
	Top Directions	0.00 +/- 0.00	- - -	0.50 +/- 0.00	0.00 +/- 0.00	0.25 +/- 0.00	0.44 +/- 0.13	0.50 +/- 0.00	0.80 +/- 0.11	0.50 +/- 0.00
	Bottom Flat	0.34 +/- 0.03	- - -	0.16 +/- 0.00	0.03 +/- 0.05	0.52 +/- 0.06	0.08 +/- 0.01	0.28 +/- 0.00	0.36 +/- 0.19	1.00 +/- 0.00
K-Means	Joints	0.54 +/- 0.17	0.83 +/- 0.00	0.94 +/- 0.10	0.83 +/- 0.00	0.83 +/- 0.00	0.96 +/- 0.08	0.83 +/- 0.00	1.00 +/- 0.00	0.11 +/- 0.10
	Members	0.55 +/- 0.16	0.84 +/- 0.00	0.91 +/- 0.09	0.84 +/- 0.00	0.84 +/- 0.00	0.96 +/- 0.08	0.80 +/- 0.00	0.96 +/- 0.00	0.11 +/- 0.09
	Connectivity	0.76 +/- 0.13	0.95 +/- 0.00	0.71 +/- 0.04	0.95 +/- 0.00	0.95 +/- 0.00	0.99 +/- 0.02	0.67 +/- 0.00	0.73 +/- 0.00	0.14 +/- 0.12
	Height	0.22 +/- 0.15	0.17 +/- 0.00	0.28 +/- 0.05	0.27 +/- 0.00	0.27 +/- 0.00	0.15 +/- 0.09	1.00 +/- 0.00	0.38 +/- 0.00	0.56 +/- 0.02
	Clearance	0.15 +/- 0.07	0.32 +/- 0.00	0.24 +/- 0.12	0.17 +/- 0.00	0.15 +/- 0.00	0.10 +/- 0.07	1.00 +/- 0.00	0.48 +/- 0.00	0.73 +/- 0.32
	Depth	0.11 +/- 0.05	0.14 +/- 0.00	0.17 +/- 0.05	0.20 +/- 0.00	0.16 +/- 0.00	0.09 +/- 0.05	1.00 +/- 0.00	0.20 +/- 0.00	0.15 +/- 0.08
	Top Flat	0.26 +/- 0.16	0.38 +/- 0.00	0.45 +/- 0.16	0.43 +/- 0.00	0.65 +/- 0.00	0.17 +/- 0.09	0.98 +/- 0.00	0.22 +/- 0.00	0.81 +/- 0.16
	Top Directions	0.25 +/- 0.19	0.50 +/- 0.00	0.75 +/- 0.00	1.00 +/- 0.00	0.75 +/- 0.00	0.44 +/- 0.13	0.50 +/- 0.00	0.25 +/- 0.00	0.00 +/- 0.00
	Bottom Flat	0.32 +/- 0.25	0.28 +/- 0.00	0.49 +/- 0.09	0.13 +/- 0.00	0.23 +/- 0.00	0.08 +/- 0.01	1.00 +/- 0.00	0.45 +/- 0.00	0.34 +/- 0.03
Nearest	Joints	0.87 +/- 0.11	0.83 +/- 0.00	1.00 +/- 0.00	0.17 +/- 0.00	0.33 +/- 0.00	0.67 +/- 0.00	1.00 +/- 0.00	0.00 +/- 0.00	0.33 +/- 0.00
Neighbor	Members	0.87 +/- 0.11	0.80 +/- 0.00	0.96 +/- 0.00	0.16 +/- 0.00	0.36 +/- 0.00	0.66 +/- 0.02	0.96 +/- 0.00	0.00 +/- 0.00	0.36 +/- 0.00
	Connectivity	0.93 +/- 0.10	0.67 +/- 0.00	0.73 +/- 0.00	0.20 +/- 0.00	0.75 +/- 0.00	0.74 +/- 0.18	0.73 +/- 0.00	0.00 +/- 0.00	0.75 +/- 0.00
	Height	0.18 +/- 0.09	1.00 +/- 0.00	0.38 +/- 0.00	0.57 +/- 0.00	0.13 +/- 0.00	0.36 +/- 0.04	0.31 +/- 0.00	0.54 +/- 0.00	0.00 +/- 0.00
	Clearance	0.14 +/- 0.09	1.00 +/- 0.00	0.48 +/- 0.00	0.92 +/- 0.00	0.15 +/- 0.00	0.19 +/- 0.00	0.31 +/- 0.00	0.36 +/- 0.00	0.00 +/- 0.00
	Depth	0.12 +/- 0.07	1.00 +/- 0.00	0.20 +/- 0.00	0.11 +/- 0.00	0.16 +/- 0.00	0.12 +/- 0.01	0.14 +/- 0.00	0.24 +/- 0.00	0.08 +/- 0.00
	Top Flat	0.31 +/- 0.23	0.98 +/- 0.00	0.22 +/- 0.00	0.71 +/- 0.00	0.35 +/- 0.00	0.34 +/- 0.05	0.36 +/- 0.00	1.00 +/- 0.00	0.00 +/- 0.00
	Top Directions	0.53 +/- 0.29	0.50 +/- 0.00	0.25 +/- 0.00	0.00 +/- 0.00	0.50 +/- 0.00	0.25 +/- 0.00	0.75 +/- 0.00	0.00 +/- 0.00	0.00 +/- 0.00
	Bottom Flat	0.16 +/- 0.11	1.00 +/- 0.00	0.45 +/- 0.00	0.32 +/- 0.00	0.16 +/- 0.00	0.55 +/- 0.00	0.54 +/- 0.00	0.37 +/- 0.00	0.00 +/- 0.00

Table B.4. Maximum and Minimum Values for S25M50 Clusters.

S25M50		Group 1		Group 2		Group 3		Group 4		Group 5		Group 6		Group 7		Group 8		Group 9	
Simulation / Feature		**Max**	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**	**Min**
2DKSOM	Joints	1.00	1.00	0.83	0.83	1.00	1.00	0.83	0.83	1.00	1.00	0.83	0.83	0.17	0.00	0.67	0.67	0.67	0.33
	Members	0.96	0.96	0.84	0.80	1.00	1.00	0.80	0.80	0.96	0.96	0.84	0.84	0.16	0.00	0.68	0.64	0.68	0.36
	Connectivity	0.73	0.73	0.95	0.67	1.00	1.00	0.67	0.67	0.73	0.73	0.95	0.95	0.20	0.00	0.90	0.58	0.90	0.75
	Height	0.31	0.31	0.27	0.22	0.23	0.11	1.00	1.00	0.38	0.38	0.17	0.04	0.57	0.54	0.39	0.32	0.13	0.00
	Clearance	0.31	0.31	0.17	0.10	0.14	0.12	1.00	1.00	0.48	0.48	0.32	0.00	0.92	0.36	0.19	0.19	0.15	0.00
	Depth	0.14	0.14	0.24	0.16	0.13	0.07	1.00	1.00	0.20	0.20	0.14	0.04	0.24	0.11	0.13	0.12	0.16	0.00
	Top Flat	0.36	0.36	0.65	0.43	0.21	0.04	0.98	0.98	0.22	0.22	0.38	0.22	1.00	0.71	0.38	0.30	0.35	0.00
	Top Directions	0.75	0.75	1.00	0.75	0.50	0.50	0.50	0.50	0.25	0.25	0.50	0.25	0.00	0.00	0.25	0.25	0.50	0.00
	Bottom Flat	0.54	0.54	0.39	0.13	0.09	0.08	1.00	1.00	0.45	0.45	0.28	0.08	0.37	0.32	0.55	0.55	0.16	0.00
1DKSOM	Joints	0.17	0.00	-	-	0.33	0.33	0.67	0.33	1.00	0.67	1.00	0.83	0.83	0.83	1.00	0.83	0.83	0.83
	Members	0.16	0.00	-	-	0.36	0.36	0.68	0.36	0.96	0.64	1.00	0.84	0.84	0.84	0.96	0.80	0.80	0.80
	Connectivity	0.20	0.00	-	-	0.75	0.75	0.90	0.75	0.90	0.58	1.00	0.95	0.95	0.95	0.95	0.67	0.67	0.67
	Height	0.57	0.54	-	-	0.13	0.13	0.07	0.00	0.39	0.32	0.23	0.04	0.17	0.17	0.31	0.22	1.00	1.00
	Clearance	0.92	0.36	-	-	0.15	0.15	0.09	0.00	0.48	0.19	0.14	0.00	0.32	0.32	0.31	0.10	1.00	1.00
	Depth	0.24	0.11	-	-	0.16	0.16	0.08	0.00	0.20	0.12	0.13	0.04	0.14	0.14	0.24	0.14	1.00	1.00
	Top Flat	1.00	0.71	-	-	0.35	0.35	0.00	0.00	0.38	0.22	0.22	0.04	0.38	0.38	0.65	0.36	0.98	0.98
	Top Directions	0.00	0.00	-	-	0.50	0.50	0.00	0.00	0.25	0.25	0.50	0.25	0.50	0.50	1.00	0.75	0.50	0.50
	Bottom Flat	0.37	0.32	-	-	0.16	0.16	0.07	0.00	0.55	0.45	0.09	0.08	0.28	0.28	0.54	0.13	1.00	1.00
K-Means	Joints	0.67	0.33	0.83	0.83	1.00	0.83	0.83	0.83	0.83	0.83	1.00	0.83	0.83	0.83	1.00	1.00	0.17	0.00
	Members	0.68	0.36	0.84	0.84	0.96	0.80	0.84	0.84	0.84	0.84	1.00	0.84	0.80	0.80	0.96	0.96	0.16	0.00
	Connectivity	0.90	0.58	0.95	0.95	0.73	0.67	0.95	0.95	0.95	0.95	1.00	0.95	0.67	0.67	0.73	0.73	0.20	0.00
	Height	0.39	0.00	0.17	0.17	0.31	0.22	0.27	0.27	0.27	0.27	0.23	0.04	1.00	1.00	0.38	0.38	0.57	0.54
	Clearance	0.19	0.00	0.32	0.32	0.31	0.10	0.17	0.17	0.15	0.15	0.14	0.00	1.00	1.00	0.48	0.48	0.92	0.36
	Depth	0.16	0.00	0.14	0.14	0.24	0.14	0.20	0.20	0.16	0.16	0.13	0.04	1.00	1.00	0.20	0.20	0.24	0.11
	Top Flat	0.38	0.00	0.38	0.38	0.64	0.36	0.43	0.43	0.65	0.65	0.22	0.04	0.98	0.98	0.22	0.22	1.00	0.71
	Top Directions	0.50	0.00	0.50	0.50	0.75	0.75	1.00	1.00	0.75	0.75	0.50	0.25	0.50	0.50	0.25	0.25	0.00	0.00
	Bottom Flat	0.55	0.00	0.28	0.28	0.54	0.39	0.13	0.13	0.23	0.23	0.09	0.08	1.00	1.00	0.45	0.45	0.37	0.32
Nearest	Joints	1.00	0.67	0.83	0.83	1.00	1.00	0.17	0.17	0.33	0.33	0.67	0.67	1.00	1.00	0.00	0.00	0.33	0.33
Neighbor	Members	1.00	0.68	0.80	0.80	0.96	0.96	0.16	0.16	0.36	0.36	0.68	0.64	0.96	0.96	0.00	0.00	0.36	0.36
	Connectivity	1.00	0.67	0.67	0.67	0.73	0.73	0.20	0.20	0.75	0.75	0.90	0.58	0.73	0.73	0.00	0.00	0.75	0.75
	Height	0.27	0.04	1.00	1.00	0.38	0.38	0.57	0.57	0.13	0.13	0.39	0.32	0.31	0.31	0.54	0.54	0.00	0.00
	Clearance	0.32	0.00	1.00	1.00	0.48	0.48	0.92	0.92	0.15	0.15	0.19	0.19	0.31	0.31	0.36	0.36	0.00	0.00
	Depth	0.24	0.00	1.00	1.00	0.20	0.20	0.11	0.11	0.16	0.16	0.13	0.12	0.14	0.14	0.24	0.24	0.08	0.08
	Top Flat	0.65	0.00	0.98	0.98	0.22	0.22	0.71	0.71	0.35	0.35	0.38	0.30	0.36	0.36	1.00	1.00	0.00	0.00
	Top Directions	1.00	0.00	0.50	0.50	0.25	0.25	0.00	0.00	0.50	0.50	0.25	0.25	0.75	0.75	0.00	0.00	0.00	0.00
	Bottom Flat	0.39	0.07	1.00	1.00	0.45	0.45	0.32	0.32	0.16	0.16	0.55	0.55	0.54	0.54	0.37	0.37	0.00	0.00

Table B.5. Averages and Standard Deviations for S50M25 Clusters.

Simulation / Feature		Group 1	Group 2	Group 3	Group 4	Group 5	Group 6	Group 7	Group 8	Group 9
2DKSOM	Joints	1.00 +/- 0.00	- - -	0.00 +/- 0.00	1.00 +/- 0.00	1.00 +/- 0.00	- - -	1.00 +/- 0.00	1.00 +/- 0.00	1.00 +/- 0.00
	Members	0.80 +/- 0.00	- - -	0.00 +/- 0.00	0.83 +/- 0.08	0.80 +/- 0.00	- - -	1.00 +/- 0.00	1.00 +/- 0.00	0.82 +/- 0.06
	Connectivity	0.35 +/- 0.00	- - -	0.00 +/- 0.00	0.46 +/- 0.26	0.35 +/- 0.00	- - -	1.00 +/- 0.00	1.00 +/- 0.00	0.41 +/- 0.19
	Height	0.29 +/- 0.00	- - -	0.29 +/- 0.00	0.24 +/- 0.12	0.29 +/- 0.00	- - -	0.68 +/- 0.00	0.29 +/- 0.00	0.81 +/- 0.19
	Clearance	0.03 +/- 0.02	- - -	0.26 +/- 0.20	0.27 +/- 0.05	0.42 +/- 0.00	- - -	0.37 +/- 0.00	0.91 +/- 0.00	0.93 +/- 0.06
	Depth	0.30 +/- 0.06	- - -	0.43 +/- 0.01	0.29 +/- 0.14	0.17 +/- 0.00	- - -	0.36 +/- 0.00	0.22 +/- 0.00	0.38 +/- 0.21
	Top Flat	0.98 +/- 0.02	- - -	0.22 +/- 0.13	0.22 +/- 0.06	0.00 +/- 0.00	- - -	0.18 +/- 0.00	0.00 +/- 0.00	0.24 +/- 0.09
	Top Directions	1.00 +/- 0.00	- - -	0.38 +/- 0.23	0.50 +/- 0.00	0.00 +/- 0.00	- - -	0.00 +/- 0.00	0.00 +/- 0.00	0.00 +/- 0.00
	Bottom Flat	0.38 +/- 0.26	- - -	0.39 +/- 0.29	0.82 +/- 0.12	0.73 +/- 0.00	- - -	0.79 +/- 0.00	0.91 +/- 0.00	0.93 +/- 0.06
1DKSOM	Joints	0.00 +/- 0.00	0.00 +/- 0.00	1.00 +/- 0.00	- - -	1.00 +/- 0.00	1.00 +/- 0.00	1.00 +/- 0.00	1.00 +/- 0.00	1.00 +/- 0.00
	Members	0.00 +/- 0.00	0.00 +/- 0.00	0.80 +/- 0.00	- - -	1.00 +/- 0.00	1.00 +/- 0.00	0.83 +/- 0.08	0.80 +/- 0.00	0.80 +/- 0.00
	Connectivity	0.00 +/- 0.00	0.00 +/- 0.00	0.35 +/- 0.00	- - -	1.00 +/- 0.00	1.00 +/- 0.00	0.46 +/- 0.26	0.35 +/- 0.00	0.35 +/- 0.00
	Height	0.29 +/- 0.00	0.29 +/- 0.00	0.29 +/- 0.00	- - -	0.29 +/- 0.00	0.72 +/- 0.09	0.24 +/- 0.12	0.29 +/- 0.00	0.80 +/- 0.20
	Clearance	0.58 +/- 0.00	0.15 +/- 0.00	0.03 +/- 0.02	- - -	0.91 +/- 0.00	0.47 +/- 0.26	0.27 +/- 0.05	0.42 +/- 0.00	0.92 +/- 0.05
	Depth	0.45 +/- 0.00	0.43 +/- 0.00	0.30 +/- 0.06	- - -	0.22 +/- 0.00	0.46 +/- 0.26	0.29 +/- 0.14	0.17 +/- 0.00	0.32 +/- 0.07
	Top Flat	0.00 +/- 0.00	0.29 +/- 0.00	0.98 +/- 0.02	- - -	0.00 +/- 0.00	0.20 +/- 0.04	0.22 +/- 0.06	0.00 +/- 0.00	0.24 +/- 0.09
	Top Directions	0.00 +/- 0.00	0.50 +/- 0.00	1.00 +/- 0.00	- - -	0.00 +/- 0.00	0.00 +/- 0.00	0.50 +/- 0.00	0.00 +/- 0.00	0.00 +/- 0.00
	Bottom Flat	0.87 +/- 0.00	0.23 +/- 0.00	0.38 +/- 0.26	- - -	0.91 +/- 0.00	0.82 +/- 0.09	0.82 +/- 0.12	0.73 +/- 0.00	0.92 +/- 0.05
K-Means	Joints	1.00 +/- 0.00	1.00 +/- 0.00	1.00 +/- 0.00	1.00 +/- 0.00	1.00 +/- 0.00	1.00 +/- 0.00	1.00 +/- 0.00	0.00 +/- 0.00	1.00 +/- 0.00
	Members	0.80 +/- 0.00	0.80 +/- 0.00	0.80 +/- 0.00	0.80 +/- 0.00	1.00 +/- 0.00	0.80 +/- 0.00	1.00 +/- 0.00	0.00 +/- 0.00	0.80 +/- 0.00
	Connectivity	0.35 +/- 0.00	0.35 +/- 0.00	0.35 +/- 0.00	0.35 +/- 0.00	1.00 +/- 0.00	0.35 +/- 0.00	1.00 +/- 0.00	0.00 +/- 0.00	0.35 +/- 0.00
	Height	0.29 +/- 0.00	0.58 +/- 0.00	0.68 +/- 0.00	0.29 +/- 0.00	0.00 +/- 0.00	0.29 +/- 0.00	0.50 +/- 0.23	0.29 +/- 0.00	1.00 +/- 0.00
	Clearance	0.29 +/- 0.00	1.00 +/- 0.00	0.87 +/- 0.00	0.42 +/- 0.00	0.15 +/- 0.00	0.03 +/- 0.02	0.69 +/- 0.29	0.26 +/- 0.20	0.90 +/- 0.00
	Depth	0.35 +/- 0.00	0.32 +/- 0.00	0.43 +/- 0.00	0.17 +/- 0.00	0.00 +/- 0.00	0.30 +/- 0.06	0.34 +/- 0.22	0.43 +/- 0.01	0.26 +/- 0.00
	Top Flat	0.25 +/- 0.00	0.13 +/- 0.00	0.18 +/- 0.00	0.00 +/- 0.00	0.10 +/- 0.00	0.98 +/- 0.02	0.10 +/- 0.11	0.22 +/- 0.13	0.33 +/- 0.00
	Top Directions	0.50 +/- 0.00	0.00 +/- 0.00	0.00 +/- 0.00	0.00 +/- 0.00	0.50 +/- 0.00	1.00 +/- 0.00	0.00 +/- 0.00	0.38 +/- 0.23	0.00 +/- 0.00
	Bottom Flat	0.87 +/- 0.00	1.00 +/- 0.00	0.87 +/- 0.00	0.73 +/- 0.00	0.57 +/- 0.00	0.38 +/- 0.26	0.87 +/- 0.07	0.39 +/- 0.29	0.90 +/- 0.00
Nearest	Joints	1.00 +/- 0.00	1.00 +/- 0.00	1.00 +/- 0.00	1.00 +/- 0.00	1.00 +/- 0.00	1.00 +/- 0.00	0.00 +/- 0.00	0.00 +/- 0.00	1.00 +/- 0.00
Neighbor	Members	1.00 +/- 0.00	0.80 +/- 0.00	0.80 +/- 0.00	1.00 +/- 0.00	0.80 +/- 0.00	1.00 +/- 0.00	0.00 +/- 0.00	0.00 +/- 0.00	1.00 +/- 0.00
	Connectivity	1.00 +/- 0.00	0.35 +/- 0.00	0.35 +/- 0.00	1.00 +/- 0.00	0.35 +/- 0.00	1.00 +/- 0.00	0.00 +/- 0.00	0.00 +/- 0.00	1.00 +/- 0.00
	Height	0.29 +/- 0.00	0.80 +/- 0.20	0.29 +/- 0.00	0.00 +/- 0.00	0.29 +/- 0.00	0.68 +/- 0.00	0.29 +/- 0.00	0.29 +/- 0.00	0.91 +/- 0.00
	Clearance	0.91 +/- 0.00	0.92 +/- 0.05	0.36 +/- 0.07	0.15 +/- 0.00	0.03 +/- 0.02	0.37 +/- 0.00	0.15 +/- 0.00	0.58 +/- 0.00	1.00 +/- 0.00
	Depth	0.22 +/- 0.00	0.32 +/- 0.07	0.25 +/- 0.09	0.00 +/- 0.00	0.30 +/- 0.06	0.36 +/- 0.00	0.43 +/- 0.00	0.45 +/- 0.00	1.00 +/- 0.00
	Top Flat	0.00 +/- 0.00	0.24 +/- 0.09	0.11 +/- 0.13	0.10 +/- 0.00	0.98 +/- 0.02	0.18 +/- 0.00	0.29 +/- 0.00	0.00 +/- 0.00	0.29 +/- 0.00
	Top Directions	0.00 +/- 0.00	0.00 +/- 0.00	0.23 +/- 0.26	0.50 +/- 0.00	1.00 +/- 0.00	0.00 +/- 0.00	0.50 +/- 0.00	0.00 +/- 0.00	0.00 +/- 0.00
	Bottom Flat	0.91 +/- 0.00	0.92 +/- 0.05	0.79 +/- 0.07	0.57 +/- 0.00	0.38 +/- 0.26	0.79 +/- 0.00	0.23 +/- 0.00	0.87 +/- 0.00	1.00 +/- 0.00

Table B.6. Maximum and Minimum Values for S50M25 Clusters.

S50M25		Group 1		Group 2		Group 3		Group 4		Group 5		Group 6		Group 7		Group 8		Group 9	
Simulation / Feature		**Max**	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**	**Min**
2DKSOM	Joints	1.00	1.00	-	-	0.00	0.00	1.00	1.00	1.00	1.00	-	-	1.00	1.00	1.00	1.00	1.00	1.00
	Members	0.80	0.80	-	-	0.00	0.00	1.00	0.80	0.80	0.80	-	-	1.00	1.00	1.00	1.00	1.00	0.80
	Connectivity	0.35	0.35	-	-	0.00	0.00	1.00	0.35	0.35	0.35	-	-	1.00	1.00	1.00	1.00	1.00	0.35
	Height	0.29	0.29	-	-	0.29	0.29	0.29	0.00	0.29	0.29	-	-	0.68	0.68	0.29	0.29	1.00	0.58
	Clearance	0.04	0.00	-	-	0.58	0.15	0.29	0.15	0.42	0.42	-	-	0.37	0.37	0.91	0.91	1.00	0.87
	Depth	0.39	0.27	-	-	0.45	0.43	0.35	0.00	0.17	0.17	-	-	0.36	0.36	0.22	0.22	1.00	0.26
	Top Flat	1.00	0.97	-	-	0.29	0.00	0.25	0.10	0.00	0.00	-	-	0.18	0.18	0.00	0.00	0.33	0.13
	Top Directions	1.00	1.00	-	-	0.50	0.00	0.50	0.50	0.00	0.00	-	-	0.00	0.00	0.00	0.00	0.00	0.00
	Bottom Flat	0.54	0.00	-	-	0.87	0.23	0.87	0.57	0.73	0.73	-	-	0.79	0.79	0.91	0.91	1.00	0.87
1DKSOM	Joints	0.00	0.00	0.00	0.00	1.00	1.00	-	-	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	Members	0.00	0.00	0.00	0.00	0.80	0.80	-	-	1.00	1.00	1.00	1.00	1.00	0.80	0.80	0.80	0.80	0.80
	Connectivity	0.00	0.00	0.00	0.00	0.35	0.35	-	-	1.00	1.00	1.00	1.00	1.00	0.35	0.35	0.35	0.35	0.35
	Height	0.29	0.29	0.29	0.29	0.29	0.29	-	-	0.29	0.29	0.91	0.68	0.29	0.00	0.29	0.29	1.00	0.58
	Clearance	0.58	0.58	0.15	0.15	0.04	0.00	-	-	0.91	0.91	1.00	0.37	0.29	0.15	0.42	0.42	1.00	0.87
	Depth	0.45	0.45	0.43	0.43	0.39	0.27	-	-	0.22	0.22	1.00	0.36	0.35	0.00	0.17	0.17	0.43	0.26
	Top Flat	0.00	0.00	0.29	0.29	1.00	0.97	-	-	0.00	0.00	0.29	0.18	0.25	0.10	0.00	0.00	0.33	0.13
	Top Directions	0.00	0.00	0.50	0.50	1.00	1.00	-	-	0.00	0.00	0.00	0.00	0.50	0.50	0.00	0.00	0.00	0.00
	Bottom Flat	0.87	0.87	0.23	0.23	0.54	0.00	-	-	0.91	0.91	1.00	0.79	0.87	0.57	0.73	0.73	1.00	0.87
K-Means	Joints	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.00	0.00	1.00	1.00
	Members	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80	1.00	1.00	0.80	0.80	1.00	1.00	0.00	0.00	0.80	0.80
	Connectivity	0.35	0.35	0.35	0.35	0.35	0.35	0.35	0.35	1.00	1.00	0.35	0.35	1.00	1.00	0.00	0.00	0.35	0.35
	Height	0.29	0.29	0.58	0.58	0.68	0.68	0.29	0.29	0.00	0.00	0.29	0.29	0.91	0.29	0.29	0.29	1.00	1.00
	Clearance	0.29	0.29	1.00	1.00	0.87	0.87	0.42	0.42	0.15	0.15	0.04	0.00	1.00	0.37	0.58	0.15	0.90	0.90
	Depth	0.35	0.35	0.32	0.32	0.43	0.43	0.17	0.17	0.00	0.00	0.39	0.27	1.00	0.22	0.45	0.43	0.26	0.26
	Top Flat	0.25	0.25	0.13	0.13	0.18	0.18	0.00	0.00	0.10	0.10	1.00	0.97	0.29	0.00	0.29	0.00	0.33	0.33
	Top Directions	0.50	0.50	0.00	0.00	0.00	0.00	0.00	0.00	0.50	0.50	1.00	1.00	0.00	0.00	0.50	0.00	0.00	0.00
	Bottom Flat	0.87	0.87	1.00	1.00	0.87	0.87	0.73	0.73	0.57	0.57	0.54	0.00	1.00	0.79	0.87	0.23	0.90	0.90
Nearest	Joints	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.00	0.00	0.00	0.00	1.00	1.00
Neighbor	Members	1.00	1.00	0.80	0.80	0.80	0.80	1.00	1.00	0.80	0.80	1.00	1.00	0.00	0.00	0.00	0.00	1.00	1.00
	Connectivity	1.00	1.00	0.35	0.35	0.35	0.35	1.00	1.00	0.35	0.35	1.00	1.00	0.00	0.00	0.00	0.00	1.00	1.00
	Height	0.29	0.29	1.00	0.58	0.29	0.29	0.00	0.00	0.29	0.29	0.68	0.68	0.29	0.29	0.29	0.29	0.91	0.91
	Clearance	0.91	0.91	1.00	0.87	0.42	0.29	0.15	0.15	0.04	0.00	0.37	0.37	0.15	0.15	0.58	0.58	1.00	1.00
	Depth	0.22	0.22	0.43	0.26	0.35	0.17	0.00	0.00	0.39	0.27	0.36	0.36	0.43	0.43	0.45	0.45	1.00	1.00
	Top Flat	0.00	0.00	0.33	0.13	0.25	0.00	0.10	0.10	1.00	0.97	0.18	0.18	0.29	0.29	0.00	0.00	0.29	0.29
	Top Directions	0.00	0.00	0.00	0.00	0.50	0.00	0.50	0.50	1.00	1.00	0.00	0.00	0.50	0.50	0.00	0.00	0.00	0.00
	Bottom Flat	0.91	0.91	1.00	0.87	0.87	0.73	0.57	0.57	0.54	0.00	0.79	0.79	0.23	0.23	0.87	0.87	1.00	1.00

Table B.7. Averages and Standard Deviations for S50M35 Clusters.

Simulation / Feature		Group 1	Group 2	Group 3	Group 4	Group 5	Group 6	Group 7	Group 8	Group 9
2DKSOM	Joints	0.16 +/- 0.24	0.50 +/- 0.00	1.00 +/- 0.00	0.50 +/- 0.00	1.00 +/- 0.00	- - -	1.00 +/- 0.00	1.00 +/- 0.00	0.69 +/- 0.26
	Members	0.18 +/- 0.19	0.56 +/- 0.00	1.00 +/- 0.00	0.56 +/- 0.00	1.00 +/- 0.00	- - -	0.94 +/- 0.06	0.89 +/- 0.00	0.68 +/- 0.17
	Connectivity	0.34 +/- 0.29	0.86 +/- 0.00	1.00 +/- 0.00	0.86 +/- 0.00	1.00 +/- 0.00	- - -	0.72 +/- 0.27	0.48 +/- 0.00	0.72 +/- 0.20
	Height	0.81 +/- 0.20	0.91 +/- 0.00	0.83 +/- 0.00	0.70 +/- 0.00	0.83 +/- 0.00	- - -	0.02 +/- 0.04	0.40 +/- 0.00	0.70 +/- 0.00
	Clearance	0.75 +/- 0.27	0.84 +/- 0.00	0.78 +/- 0.00	0.00 +/- 0.00	0.59 +/- 0.00	- - -	0.04 +/- 0.02	0.12 +/- 0.00	0.35 +/- 0.22
	Depth	0.68 +/- 0.19	0.73 +/- 0.00	1.00 +/- 0.00	0.95 +/- 0.00	0.58 +/- 0.00	- - -	0.13 +/- 0.12	0.39 +/- 0.00	0.83 +/- 0.10
	Top Flat	0.48 +/- 0.18	0.45 +/- 0.00	0.42 +/- 0.00	0.36 +/- 0.00	0.30 +/- 0.00	- - -	0.02 +/- 0.02	0.35 +/- 0.00	0.70 +/- 0.25
	Top Directions	0.00 +/- 0.00	0.00 +/- 0.00	0.00 +/- 0.00	0.00 +/- 0.00	0.00 +/- 0.00	- - -	0.00 +/- 0.00	1.00 +/- 0.00	1.00 +/- 0.00
	Bottom Flat	0.85 +/- 0.08	0.80 +/- 0.00	0.51 +/- 0.00	1.00 +/- 0.00	0.56 +/- 0.00	- - -	0.13 +/- 0.14	0.10 +/- 0.00	0.25 +/- 0.11
1DKSOM	Joints	1.00 +/- 0.00	0.58 +/- 0.20	1.00 +/- 0.00	- - -	1.00 +/- 0.00	1.00 +/- 0.00	0.34 +/- 0.24	0.00 +/- 0.00	0.25 +/- 0.35
	Members	0.94 +/- 0.06	0.61 +/- 0.14	0.89 +/- 0.00	- - -	1.00 +/- 0.00	1.00 +/- 0.00	0.38 +/- 0.19	0.00 +/- 0.00	0.33 +/- 0.31
	Connectivity	0.72 +/- 0.27	0.80 +/- 0.16	0.48 +/- 0.00	- - -	1.00 +/- 0.00	1.00 +/- 0.00	0.62 +/- 0.25	0.00 +/- 0.00	0.77 +/- 0.13
	Height	0.02 +/- 0.04	0.65 +/- 0.12	0.70 +/- 0.00	- - -	0.83 +/- 0.00	0.83 +/- 0.00	0.94 +/- 0.05	0.55 +/- 0.00	0.70 +/- 0.00
	Clearance	0.04 +/- 0.02	0.18 +/- 0.03	0.61 +/- 0.00	- - -	0.59 +/- 0.00	0.78 +/- 0.00	0.90 +/- 0.07	0.47 +/- 0.00	0.08 +/- 0.12
	Depth	0.13 +/- 0.12	0.81 +/- 0.21	0.71 +/- 0.00	- - -	0.58 +/- 0.00	1.00 +/- 0.00	0.77 +/- 0.09	0.43 +/- 0.00	0.83 +/- 0.17
	Top Flat	0.02 +/- 0.02	0.49 +/- 0.07	1.00 +/- 0.00	- - -	0.30 +/- 0.00	0.42 +/- 0.00	0.53 +/- 0.13	0.28 +/- 0.00	0.44 +/- 0.12
	Top Directions	0.00 +/- 0.00	1.00 +/- 0.00	1.00 +/- 0.00	- - -	0.00 +/- 0.00	0.00 +/- 0.00	0.00 +/- 0.00	0.00 +/- 0.00	0.00 +/- 0.00
	Bottom Flat	0.13 +/- 0.14	0.16 +/- 0.03	0.39 +/- 0.00	- - -	0.56 +/- 0.00	0.51 +/- 0.00	0.86 +/- 0.07	0.77 +/- 0.00	0.92 +/- 0.12
K-Means	Joints	- - -	1.00 +/- 0.00	0.00 +/- 0.00	0.50 +/- 0.00	0.00 +/- 0.00	1.00 +/- 0.00	0.34 +/- 0.24	0.72 +/- 0.26	1.00 +/- 0.00
	Members	- - -	0.89 +/- 0.00	0.11 +/- 0.00	0.56 +/- 0.00	0.00 +/- 0.00	1.00 +/- 0.00	0.38 +/- 0.19	0.70 +/- 0.18	1.00 +/- 0.00
	Connectivity	- - -	0.48 +/- 0.00	0.68 +/- 0.00	0.86 +/- 0.00	0.00 +/- 0.00	1.00 +/- 0.00	0.62 +/- 0.25	0.69 +/- 0.20	1.00 +/- 0.00
	Height	- - -	0.02 +/- 0.05	0.70 +/- 0.00	0.70 +/- 0.00	0.55 +/- 0.00	0.02 +/- 0.00	0.94 +/- 0.05	0.67 +/- 0.10	0.83 +/- 0.00
	Clearance	- - -	0.06 +/- 0.00	0.17 +/- 0.00	0.00 +/- 0.00	0.47 +/- 0.00	0.01 +/- 0.00	0.90 +/- 0.07	0.32 +/- 0.22	0.65 +/- 0.09
	Depth	- - -	0.23 +/- 0.01	0.71 +/- 0.00	0.95 +/- 0.00	0.43 +/- 0.00	0.00 +/- 0.00	0.77 +/- 0.09	0.78 +/- 0.17	0.70 +/- 0.20
	Top Flat	- - -	0.03 +/- 0.00	0.53 +/- 0.00	0.36 +/- 0.00	0.28 +/- 0.00	0.00 +/- 0.00	0.53 +/- 0.13	0.66 +/- 0.26	0.33 +/- 0.06
	Top Directions	- - -	0.00 +/- 0.00	0.00 +/- 0.00	0.00 +/- 0.00	0.00 +/- 0.00	0.00 +/- 0.00	0.00 +/- 0.00	1.00 +/- 0.00	0.00 +/- 0.00
	Bottom Flat	- - -	0.23 +/- 0.10	0.84 +/- 0.00	1.00 +/- 0.00	0.77 +/- 0.00	0.00 +/- 0.00	0.86 +/- 0.07	0.23 +/- 0.12	0.54 +/- 0.02
Nearest	Joints	1.00 +/- 0.00	1.00 +/- 0.00	0.00 +/- 0.00	0.50 +/- 0.00	0.00 +/- 0.00	0.34 +/- 0.24	0.50 +/- 0.00	1.00 +/- 0.00	1.00 +/- 0.00
Neighbor	Members	0.89 +/- 0.00	0.94 +/- 0.06	0.11 +/- 0.00	0.56 +/- 0.00	0.00 +/- 0.00	0.38 +/- 0.19	0.56 +/- 0.00	1.00 +/- 0.00	0.89 +/- 0.00
	Connectivity	0.48 +/- 0.00	0.72 +/- 0.27	0.68 +/- 0.00	0.86 +/- 0.00	0.00 +/- 0.00	0.62 +/- 0.25	0.86 +/- 0.00	1.00 +/- 0.00	0.48 +/- 0.00
	Height	0.40 +/- 0.00	0.02 +/- 0.04	0.70 +/- 0.00	0.70 +/- 0.00	0.55 +/- 0.00	0.94 +/- 0.05	0.70 +/- 0.00	0.83 +/- 0.00	0.70 +/- 0.00
	Clearance	0.12 +/- 0.00	0.04 +/- 0.02	0.17 +/- 0.00	0.00 +/- 0.00	0.47 +/- 0.00	0.90 +/- 0.07	0.19 +/- 0.00	0.65 +/- 0.09	0.61 +/- 0.00
	Depth	0.39 +/- 0.00	0.13 +/- 0.12	0.71 +/- 0.00	0.95 +/- 0.00	0.43 +/- 0.00	0.77 +/- 0.09	0.90 +/- 0.00	0.70 +/- 0.20	0.71 +/- 0.00
	Top Flat	0.35 +/- 0.00	0.02 +/- 0.02	0.53 +/- 0.00	0.36 +/- 0.00	0.28 +/- 0.00	0.53 +/- 0.13	0.52 +/- 0.00	0.33 +/- 0.06	1.00 +/- 0.00
	Top Directions	1.00 +/- 0.00	0.00 +/- 0.00	0.00 +/- 0.00	0.00 +/- 0.00	0.00 +/- 0.00	0.00 +/- 0.00	1.00 +/- 0.00	0.00 +/- 0.00	1.00 +/- 0.00
	Bottom Flat	0.10 +/- 0.00	0.13 +/- 0.14	0.84 +/- 0.00	1.00 +/- 0.00	0.77 +/- 0.00	0.86 +/- 0.07	0.17 +/- 0.00	0.54 +/- 0.02	0.39 +/- 0.00

Table B.8. Maximum and Minimum Values for S50M35 Clusters.

S50M35		Group 1		Group 2		Group 3		Group 4		Group 5		Group 6		Group 7		Group 8		Group 9	
Simulation / Feature		**Max**	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**	**Min**
2DKSOM	Joints	0.50	0.00	0.50	0.50	1.00	1.00	0.50	0.50	1.00	1.00	-	-	1.00	1.00	1.00	1.00	1.00	0.50
	Members	0.44	0.00	0.56	0.56	1.00	1.00	0.56	0.56	1.00	1.00	-	-	1.00	0.89	0.89	0.89	0.89	0.56
	Connectivity	0.68	0.00	0.86	0.86	1.00	1.00	0.86	0.86	1.00	1.00	-	-	1.00	0.48	0.48	0.48	0.86	0.48
	Height	1.00	0.55	0.91	0.91	0.83	0.83	0.70	0.70	0.83	0.83	-	-	0.13	0.00	0.40	0.40	0.70	0.70
	Clearance	1.00	0.17	0.84	0.84	0.78	0.78	0.00	0.00	0.59	0.59	-	-	0.06	0.01	0.12	0.12	0.61	0.19
	Depth	0.89	0.43	0.73	0.73	1.00	1.00	0.95	0.95	0.58	0.58	-	-	0.26	0.00	0.39	0.39	0.90	0.71
	Top Flat	0.71	0.28	0.45	0.45	0.42	0.42	0.36	0.36	0.30	0.30	-	-	0.04	0.00	0.35	0.35	1.00	0.52
	Top Directions	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	-	-	0.00	0.00	1.00	1.00	1.00	1.00
	Bottom Flat	0.96	0.77	0.80	0.80	0.51	0.51	1.00	1.00	0.56	0.56	-	-	0.27	0.00	0.10	0.10	0.39	0.17
1DKSOM	Joints	1.00	1.00	1.00	0.50	1.00	1.00	-	-	1.00	1.00	1.00	1.00	0.50	0.00	0.00	0.00	0.50	0.00
	Members	1.00	0.89	0.89	0.56	0.89	0.89	-	-	1.00	1.00	1.00	1.00	0.56	0.11	0.00	0.00	0.56	0.11
	Connectivity	1.00	0.48	0.86	0.48	0.48	0.48	-	-	1.00	1.00	1.00	1.00	0.86	0.27	0.00	0.00	0.86	0.68
	Height	0.13	0.00	0.70	0.40	0.70	0.70	-	-	0.83	0.83	0.83	0.83	1.00	0.90	0.55	0.55	0.70	0.70
	Clearance	0.06	0.01	0.19	0.12	0.61	0.61	-	-	0.59	0.59	0.78	0.78	1.00	0.84	0.47	0.47	0.17	0.00
	Depth	0.26	0.00	0.90	0.39	0.71	0.71	-	-	0.58	0.58	1.00	1.00	0.89	0.70	0.43	0.43	0.95	0.71
	Top Flat	0.04	0.00	0.52	0.35	1.00	1.00	-	-	0.30	0.30	0.42	0.42	0.71	0.45	0.28	0.28	0.53	0.36
	Top Directions	0.00	0.00	1.00	1.00	1.00	1.00	-	-	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Bottom Flat	0.27	0.00	0.17	0.10	0.39	0.39	-	-	0.56	0.56	0.51	0.51	0.96	0.80	0.77	0.77	1.00	0.84
K-Means	Joints	-	-	1.00	1.00	0.00	0.00	0.50	0.50	0.00	0.00	1.00	1.00	0.50	0.00	1.00	0.50	1.00	1.00
	Members	-	-	0.89	0.89	0.11	0.11	0.56	0.56	0.00	0.00	1.00	1.00	0.56	0.11	0.89	0.56	1.00	1.00
	Connectivity	-	-	0.48	0.48	0.68	0.68	0.86	0.86	0.00	0.00	1.00	1.00	0.86	0.27	0.86	0.48	1.00	1.00
	Height	-	-	0.13	0.00	0.70	0.70	0.70	0.70	0.55	0.55	0.02	0.02	1.00	0.90	0.70	0.40	0.83	0.83
	Clearance	-	-	0.06	0.06	0.17	0.17	0.00	0.00	0.47	0.47	0.01	0.01	1.00	0.84	0.61	0.12	0.78	0.59
	Depth	-	-	0.26	0.22	0.71	0.71	0.95	0.95	0.43	0.43	0.00	0.00	0.89	0.70	0.90	0.39	1.00	0.58
	Top Flat	-	-	0.04	0.03	0.53	0.53	0.36	0.36	0.28	0.28	0.00	0.00	0.71	0.45	1.00	0.35	0.42	0.30
	Top Directions	-	-	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	1.00	0.00	0.00
	Bottom Flat	-	-	0.27	0.04	0.84	0.84	1.00	1.00	0.77	0.77	0.00	0.00	0.96	0.80	0.39	0.10	0.56	0.51
Nearest	Joints	1.00	1.00	1.00	1.00	0.00	0.00	0.50	0.50	0.00	0.00	0.50	0.00	0.50	0.50	1.00	1.00	1.00	1.00
Neighbor	Members	0.89	0.89	1.00	0.89	0.11	0.11	0.56	0.56	0.00	0.00	0.56	0.11	0.56	0.56	1.00	1.00	0.89	0.89
	Connectivity	0.48	0.48	1.00	0.48	0.68	0.68	0.86	0.86	0.00	0.00	0.86	0.27	0.86	0.86	1.00	1.00	0.48	0.48
	Height	0.40	0.40	0.13	0.00	0.70	0.70	0.70	0.70	0.55	0.55	1.00	0.90	0.70	0.70	0.83	0.83	0.70	0.70
	Clearance	0.12	0.12	0.06	0.01	0.17	0.17	0.00	0.00	0.47	0.47	1.00	0.84	0.19	0.19	0.78	0.59	0.61	0.61
	Depth	0.39	0.39	0.26	0.00	0.71	0.71	0.95	0.95	0.43	0.43	0.89	0.70	0.90	0.90	1.00	0.58	0.71	0.71
	Top Flat	0.35	0.35	0.04	0.00	0.53	0.53	0.36	0.36	0.28	0.28	0.71	0.45	0.52	0.52	0.42	0.30	1.00	1.00
	Top Directions	1.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	1.00	0.00	0.00	1.00	1.00
	Bottom Flat	0.10	0.10	0.27	0.00	0.84	0.84	1.00	1.00	0.77	0.77	0.96	0.80	0.17	0.17	0.56	0.51	0.39	0.39

Table B.9. Averages and Standard Deviations for S50M50 Clusters.

Simulation / Feature		Group 1	Group 2	Group 3	Group 4	Group 5	Group 6	Group 7	Group 8	Group 9
2DKSOM	Joints	0.11 +/- 0.20	0.73 +/- 0.12	0.90 +/- 0.14	0.22 +/- 0.18	0.72 +/- 0.11	0.92 +/- 0.11	0.50 +/- 0.12	0.65 +/- 0.18	1.00 +/- 0.00
	Members	0.12 +/- 0.18	0.70 +/- 0.11	0.90 +/- 0.13	0.24 +/- 0.16	0.70 +/- 0.08	0.88 +/- 0.10	0.49 +/- 0.10	0.64 +/- 0.15	1.00 +/- 0.00
	Connectivity	0.26 +/- 0.26	0.55 +/- 0.06	0.97 +/- 0.04	0.55 +/- 0.23	0.70 +/- 0.16	0.63 +/- 0.05	0.52 +/- 0.19	0.71 +/- 0.14	1.00 +/- 0.00
	Height	0.79 +/- 0.23	0.85 +/- 0.08	0.60 +/- 0.07	0.58 +/- 0.18	0.51 +/- 0.11	0.28 +/- 0.11	0.18 +/- 0.07	0.09 +/- 0.07	0.11 +/- 0.09
	Clearance	0.60 +/- 0.21	0.70 +/- 0.26	0.51 +/- 0.09	0.33 +/- 0.20	0.24 +/- 0.22	0.24 +/- 0.04	0.31 +/- 0.17	0.11 +/- 0.09	0.12 +/- 0.09
	Depth	0.73 +/- 0.33	0.66 +/- 0.08	0.47 +/- 0.07	0.42 +/- 0.09	0.54 +/- 0.08	0.26 +/- 0.05	0.07 +/- 0.09	0.27 +/- 0.09	0.22 +/- 0.08
	Top Flat	0.73 +/- 0.19	0.56 +/- 0.13	0.36 +/- 0.07	0.40 +/- 0.09	0.41 +/- 0.14	0.14 +/- 0.09	0.15 +/- 0.03	0.20 +/- 0.02	0.12 +/- 0.05
	Top Directions	0.36 +/- 0.24	0.50 +/- 0.00	0.50 +/- 0.00	0.08 +/- 0.17	0.45 +/- 0.11	0.10 +/- 0.14	0.00 +/- 0.00	0.63 +/- 0.19	0.58 +/- 0.26
	Bottom Flat	0.57 +/- 0.15	0.70 +/- 0.26	0.43 +/- 0.10	0.49 +/- 0.22	0.52 +/- 0.14	0.25 +/- 0.02	0.24 +/- 0.05	0.17 +/- 0.17	0.13 +/- 0.11
1DKSOM	Joints	1.00 +/- 0.00	0.72 +/- 0.21	0.78 +/- 0.12	0.60 +/- 0.18	0.00 +/- 0.00	0.36 +/- 0.09	0.08 +/- 0.11	0.00 +/- 0.00	0.47 +/- 0.12
	Members	0.98 +/- 0.03	0.71 +/- 0.20	0.76 +/- 0.11	0.58 +/- 0.16	0.05 +/- 0.00	0.38 +/- 0.08	0.10 +/- 0.09	0.00 +/- 0.00	0.44 +/- 0.11
	Connectivity	0.86 +/- 0.18	0.76 +/- 0.17	0.73 +/- 0.18	0.54 +/- 0.15	0.56 +/- 0.00	0.68 +/- 0.19	0.30 +/- 0.25	0.00 +/- 0.00	0.39 +/- 0.07
	Height	0.20 +/- 0.11	0.07 +/- 0.07	0.59 +/- 0.15	0.22 +/- 0.08	0.48 +/- 0.20	0.71 +/- 0.09	0.48 +/- 0.09	0.94 +/- 0.00	0.98 +/- 0.04
	Clearance	0.20 +/- 0.07	0.10 +/- 0.08	0.37 +/- 0.22	0.28 +/- 0.13	0.29 +/- 0.09	0.48 +/- 0.12	0.34 +/- 0.29	0.39 +/- 0.00	0.91 +/- 0.08
	Depth	0.23 +/- 0.07	0.25 +/- 0.09	0.56 +/- 0.11	0.15 +/- 0.14	0.43 +/- 0.09	0.38 +/- 0.06	0.43 +/- 0.11	0.99 +/- 0.00	0.85 +/- 0.25
	Top Flat	0.12 +/- 0.08	0.19 +/- 0.04	0.42 +/- 0.11	0.14 +/- 0.02	0.42 +/- 0.14	0.40 +/- 0.07	0.55 +/- 0.16	1.00 +/- 0.00	0.62 +/- 0.07
	Top Directions	0.32 +/- 0.19	0.68 +/- 0.21	0.47 +/- 0.08	0.00 +/- 0.00	0.00 +/- 0.00	0.00 +/- 0.00	0.45 +/- 0.11	0.50 +/- 0.00	0.17 +/- 0.29
	Bottom Flat	0.18 +/- 0.09	0.17 +/- 0.16	0.51 +/- 0.11	0.24 +/- 0.04	0.31 +/- 0.11	0.45 +/- 0.10	0.65 +/- 0.20	0.44 +/- 0.00	0.84 +/- 0.14
K-Means	Joints	0.36 +/- 0.09	0.47 +/- 0.12	0.90 +/- 0.14	0.80 +/- 0.00	0.94 +/- 0.10	0.04 +/- 0.08	0.60 +/- 0.18	0.60 +/- 0.00	0.50 +/- 0.12
	Members	0.38 +/- 0.08	0.44 +/- 0.11	0.90 +/- 0.13	0.76 +/- 0.00	0.92 +/- 0.11	0.06 +/- 0.07	0.58 +/- 0.16	0.62 +/- 0.00	0.52 +/- 0.11
	Connectivity	0.68 +/- 0.19	0.39 +/- 0.07	0.97 +/- 0.04	0.58 +/- 0.00	0.79 +/- 0.20	0.32 +/- 0.26	0.54 +/- 0.15	0.87 +/- 0.00	0.83 +/- 0.05
	Height	0.71 +/- 0.09	0.98 +/- 0.04	0.60 +/- 0.07	0.62 +/- 0.20	0.13 +/- 0.12	0.57 +/- 0.22	0.22 +/- 0.08	0.52 +/- 0.00	0.13 +/- 0.08
	Clearance	0.48 +/- 0.12	0.91 +/- 0.08	0.51 +/- 0.09	0.46 +/- 0.08	0.13 +/- 0.10	0.34 +/- 0.20	0.28 +/- 0.13	0.00 +/- 0.00	0.17 +/- 0.07
	Depth	0.38 +/- 0.06	0.85 +/- 0.25	0.47 +/- 0.07	0.58 +/- 0.13	0.21 +/- 0.06	0.54 +/- 0.25	0.15 +/- 0.14	0.61 +/- 0.00	0.34 +/- 0.04
	Top Flat	0.40 +/- 0.07	0.62 +/- 0.07	0.36 +/- 0.07	0.42 +/- 0.14	0.15 +/- 0.07	0.60 +/- 0.25	0.14 +/- 0.02	0.46 +/- 0.00	0.20 +/- 0.02
	Top Directions	0.00 +/- 0.00	0.17 +/- 0.29	0.50 +/- 0.00	0.45 +/- 0.11	0.50 +/- 0.27	0.33 +/- 0.24	0.00 +/- 0.00	0.50 +/- 0.00	0.63 +/- 0.25
	Bottom Flat	0.45 +/- 0.10	0.84 +/- 0.14	0.43 +/- 0.10	0.50 +/- 0.13	0.17 +/- 0.11	0.51 +/- 0.22	0.24 +/- 0.04	0.60 +/- 0.00	0.19 +/- 0.20
Nearest	Joints	0.40 +/- 0.00	0.79 +/- 0.20	0.23 +/- 0.20	0.60 +/- 0.00	0.20 +/- 0.00	0.60 +/- 0.00	0.00 +/- 0.00	0.00 +/- 0.00	0.00 +/- 0.00
Neighbor	Members	0.38 +/- 0.00	0.77 +/- 0.19	0.26 +/- 0.18	0.62 +/- 0.00	0.19 +/- 0.00	0.57 +/- 0.00	0.00 +/- 0.00	0.00 +/- 0.00	0.05 +/- 0.00
	Connectivity	0.35 +/- 0.00	0.72 +/- 0.20	0.63 +/- 0.16	0.87 +/- 0.00	0.20 +/- 0.00	0.48 +/- 0.00	0.00 +/- 0.00	0.00 +/- 0.00	0.56 +/- 0.00
	Height	1.00 +/- 0.00	0.26 +/- 0.23	0.62 +/- 0.18	0.52 +/- 0.00	0.38 +/- 0.01	0.93 +/- 0.00	0.52 +/- 0.00	0.94 +/- 0.00	0.55 +/- 0.00
	Clearance	0.87 +/- 0.00	0.25 +/- 0.17	0.41 +/- 0.14	0.00 +/- 0.00	0.03 +/- 0.03	1.00 +/- 0.00	0.65 +/- 0.00	0.39 +/- 0.00	0.50 +/- 0.00
	Depth	1.00 +/- 0.00	0.30 +/- 0.18	0.40 +/- 0.07	0.61 +/- 0.00	0.51 +/- 0.15	0.56 +/- 0.00	0.35 +/- 0.00	0.99 +/- 0.00	0.39 +/- 0.00
	Top Flat	0.58 +/- 0.00	0.22 +/- 0.13	0.41 +/- 0.09	0.46 +/- 0.00	0.38 +/- 0.10	0.70 +/- 0.00	0.59 +/- 0.00	1.00 +/- 0.00	0.68 +/- 0.00
	Top Directions	0.00 +/- 0.00	0.41 +/- 0.29	0.00 +/- 0.00	0.50 +/- 0.00	0.38 +/- 0.18	0.50 +/- 0.00	0.50 +/- 0.00	0.50 +/- 0.00	0.50 +/- 0.00
	Bottom Flat	0.76 +/- 0.00	0.26 +/- 0.17	0.40 +/- 0.12	0.60 +/- 0.00	0.86 +/- 0.06	1.00 +/- 0.00	0.42 +/- 0.00	0.44 +/- 0.00	0.57 +/- 0.00

Table B.10. Maximum and Minimum Values for S50M50 Clusters.

S50M50		Group 1		Group 2		Group 3		Group 4		Group 5		Group 6		Group 7		Group 8		Group 9	
Simulation / Feature		**Max**	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**	**Min**
2DKSOM	Joints	0.40	0.00	0.80	0.60	1.00	0.80	0.40	0.00	0.80	0.60	1.00	0.80	0.60	0.40	0.80	0.40	1.00	1.00
	Members	0.38	0.00	0.76	0.57	1.00	0.81	0.43	0.05	0.76	0.62	0.95	0.76	0.57	0.38	0.76	0.43	1.00	1.00
	Connectivity	0.56	0.00	0.58	0.48	1.00	0.94	0.79	0.20	0.87	0.58	0.67	0.58	0.79	0.35	0.87	0.58	1.00	1.00
	Height	1.00	0.52	0.93	0.80	0.65	0.55	0.86	0.25	0.59	0.33	0.35	0.08	0.27	0.10	0.25	0.02	0.18	0.00
	Clearance	0.87	0.39	1.00	0.55	0.57	0.44	0.69	0.01	0.42	0.00	0.28	0.20	0.43	0.09	0.25	0.00	0.22	0.01
	Depth	1.00	0.35	0.71	0.56	0.52	0.42	0.62	0.31	0.61	0.42	0.30	0.20	0.19	0.00	0.38	0.16	0.30	0.13
	Top Flat	1.00	0.58	0.70	0.48	0.41	0.31	0.51	0.26	0.49	0.17	0.23	0.00	0.18	0.12	0.23	0.18	0.22	0.05
	Top Directions	0.50	0.00	0.50	0.50	0.50	0.50	0.50	0.00	0.50	0.25	0.25	0.00	0.00	0.00	1.00	0.50	1.00	0.25
	Bottom Flat	0.76	0.42	1.00	0.55	0.50	0.36	0.90	0.25	0.66	0.37	0.27	0.23	0.27	0.16	0.48	0.03	0.26	0.00
1DKSOM	Joints	1.00	1.00	1.00	0.40	1.00	0.60	0.80	0.40	0.00	0.00	0.40	0.20	0.20	0.00	0.00	0.00	0.60	0.40
	Members	1.00	0.95	1.00	0.43	1.00	0.62	0.76	0.38	0.05	0.05	0.43	0.24	0.19	0.00	0.00	0.00	0.57	0.38
	Connectivity	1.00	0.67	1.00	0.58	1.00	0.58	0.79	0.35	0.56	0.56	0.79	0.35	0.56	0.00	0.00	0.00	0.48	0.35
	Height	0.35	0.08	0.25	0.00	0.80	0.33	0.30	0.10	0.59	0.25	0.86	0.64	0.55	0.38	0.94	0.94	1.00	0.93
	Clearance	0.28	0.07	0.25	0.00	0.57	0.00	0.43	0.09	0.39	0.24	0.69	0.42	0.65	0.01	0.39	0.39	1.00	0.87
	Depth	0.30	0.13	0.38	0.13	0.71	0.42	0.30	0.00	0.47	0.33	0.46	0.31	0.62	0.35	0.99	0.99	1.00	0.56
	Top Flat	0.23	0.00	0.23	0.11	0.49	0.17	0.18	0.12	0.50	0.26	0.51	0.32	0.68	0.31	1.00	1.00	0.70	0.58
	Top Directions	0.50	0.00	1.00	0.50	0.50	0.25	0.00	0.00	0.00	0.00	0.00	0.00	0.50	0.25	0.50	0.50	0.50	0.00
	Bottom Flat	0.27	0.07	0.48	0.00	0.66	0.36	0.27	0.16	0.43	0.25	0.59	0.35	0.90	0.42	0.44	0.44	1.00	0.76
K-Means	Joints	0.40	0.20	0.60	0.40	1.00	0.80	0.80	0.80	1.00	0.80	0.20	0.00	0.80	0.40	0.60	0.60	0.60	0.40
	Members	0.43	0.24	0.57	0.38	1.00	0.81	0.76	0.76	1.00	0.76	0.19	0.00	0.76	0.38	0.62	0.62	0.62	0.43
	Connectivity	0.79	0.35	0.48	0.35	1.00	0.94	0.58	0.58	1.00	0.58	0.56	0.00	0.79	0.35	0.87	0.87	0.87	0.79
	Height	0.86	0.64	1.00	0.93	0.65	0.55	0.80	0.33	0.35	0.00	0.94	0.25	0.30	0.10	0.52	0.52	0.25	0.09
	Clearance	0.69	0.42	1.00	0.87	0.57	0.44	0.55	0.39	0.28	0.00	0.65	0.01	0.43	0.09	0.00	0.00	0.25	0.09
	Depth	0.46	0.31	1.00	0.56	0.52	0.42	0.71	0.42	0.30	0.13	0.99	0.33	0.30	0.00	0.61	0.61	0.38	0.29
	Top Flat	0.51	0.32	0.70	0.58	0.41	0.31	0.49	0.17	0.23	0.00	1.00	0.26	0.18	0.12	0.46	0.46	0.23	0.18
	Top Directions	0.00	0.00	0.50	0.00	0.50	0.50	0.50	0.25	1.00	0.00	0.50	0.00	0.00	0.00	0.50	0.50	1.00	0.50
	Bottom Flat	0.59	0.35	1.00	0.76	0.50	0.36	0.66	0.37	0.30	0.00	0.90	0.25	0.27	0.16	0.60	0.60	0.48	0.03
Nearest	Joints	0.40	0.40	1.00	0.40	0.40	0.00	0.60	0.60	0.20	0.20	0.60	0.60	0.00	0.00	0.00	0.00	0.00	0.00
Neighbor	Members	0.38	0.38	1.00	0.38	0.43	0.05	0.62	0.62	0.19	0.19	0.57	0.57	0.00	0.00	0.00	0.00	0.05	0.05
	Connectivity	0.35	0.35	1.00	0.35	0.79	0.35	0.87	0.87	0.20	0.20	0.48	0.48	0.00	0.00	0.00	0.00	0.56	0.56
	Height	1.00	1.00	0.80	0.00	0.86	0.25	0.52	0.52	0.39	0.38	0.93	0.93	0.52	0.52	0.94	0.94	0.55	0.55
	Clearance	0.87	0.87	0.57	0.00	0.69	0.24	0.00	0.00	0.05	0.01	1.00	1.00	0.65	0.65	0.39	0.39	0.50	0.50
	Depth	1.00	1.00	0.71	0.00	0.47	0.31	0.61	0.61	0.62	0.40	0.56	0.56	0.35	0.35	0.99	0.99	0.39	0.39
	Top Flat	0.58	0.58	0.49	0.00	0.51	0.26	0.46	0.46	0.46	0.31	0.70	0.70	0.59	0.59	1.00	1.00	0.68	0.68
	Top Directions	0.00	0.00	1.00	0.00	0.00	0.00	0.50	0.50	0.50	0.25	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
	Bottom Flat	0.76	0.76	0.66	0.00	0.59	0.25	0.60	0.60	0.90	0.82	1.00	1.00	0.42	0.42	0.44	0.44	0.57	0.57

APPENDIX C

This appendix contains additional results for the input reproduction and preference detection trials of Chapter IV. These trials were used to identify an appropriate method for predicting a user's design preferences. Four methods were used in this investigation: Kohonen's Self Organizing Map (KSOM), Rough Set Reduct (RSR), Back-propagation Neural Network (BPNN), and the hybrid Back-prop Neural Network with Rough Set Reduct (BP-RSR) method. Results presented here are for populations S25M35(a), S25M50(a), S50M25(a), and S50M50(a). The following figures illustrate the predictions made by each of the four algorithms when presented with the user's selections shown in Figures 4.7 to 4.11.

Input Reproduction Trials

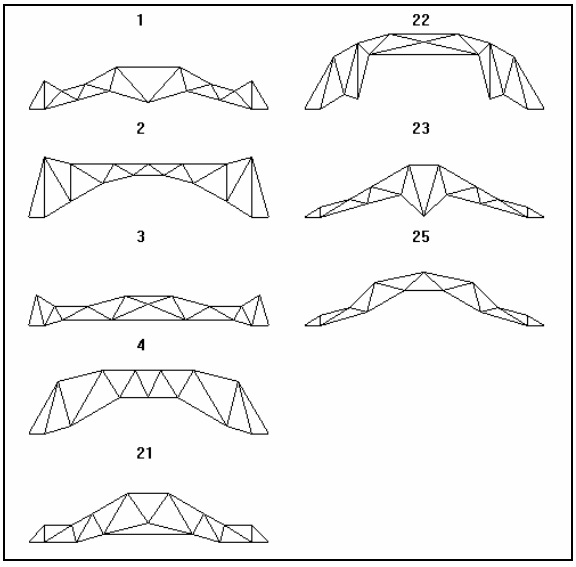


Figure C.1. KSOM predictions for population S25M35(a) during input reproduction trials.

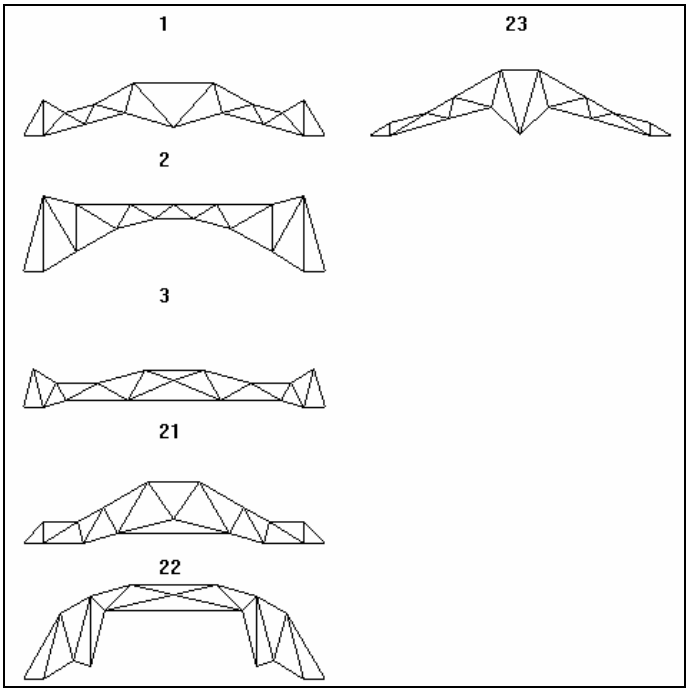


Figure C.2. RSR predictions for population S25M35(a) during input reproduction trials.

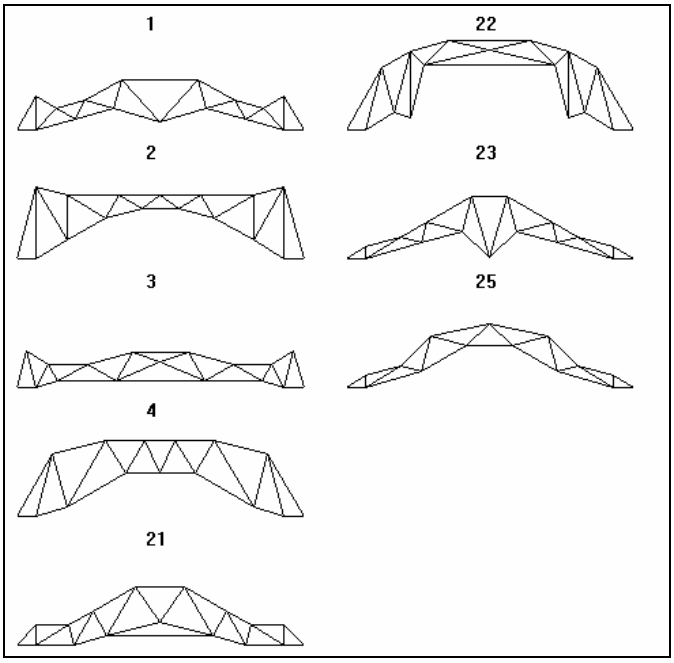


Figure C.3. BPNN predictions for population S25M35(a) during input reproduction trials.

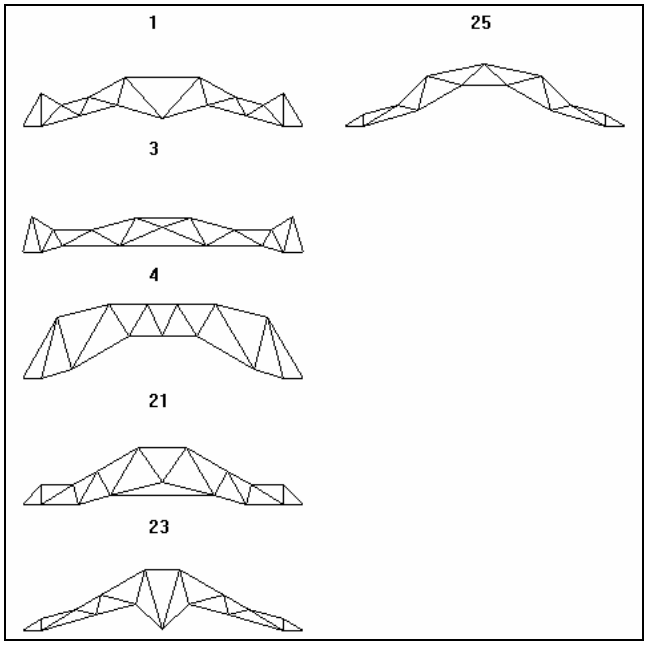


Figure C.4. BP-RSR predictions for population S25M35(a) during input reproduction trials.

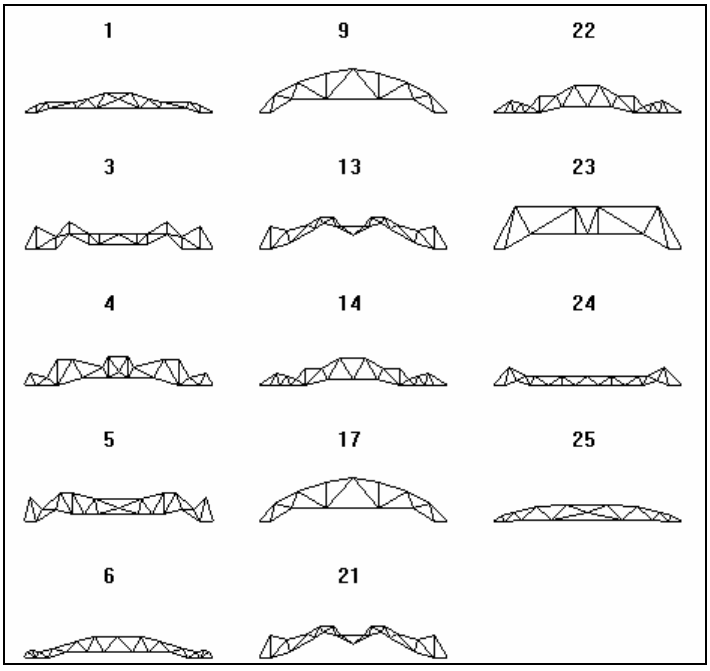


Figure C.5. KSOM predictions for population S25M50(a) during input reproduction trials.

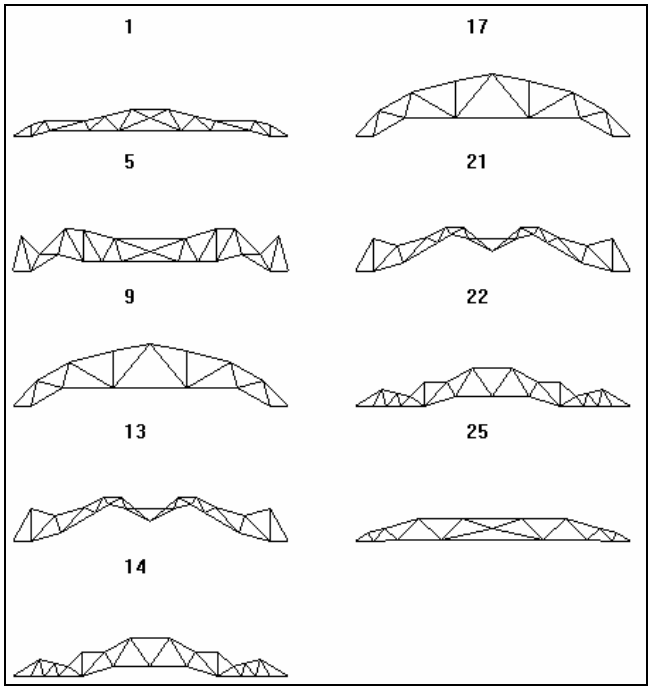


Figure C.6. RSR predictions for population S25M50(a) during input reproduction trials.

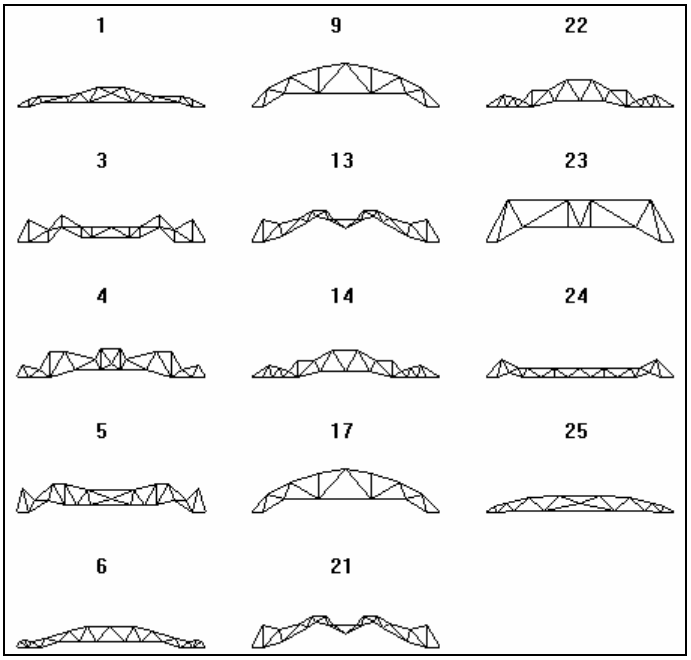


Figure C.7. BPNN predictions for population S25M50(a) during input reproduction trials.

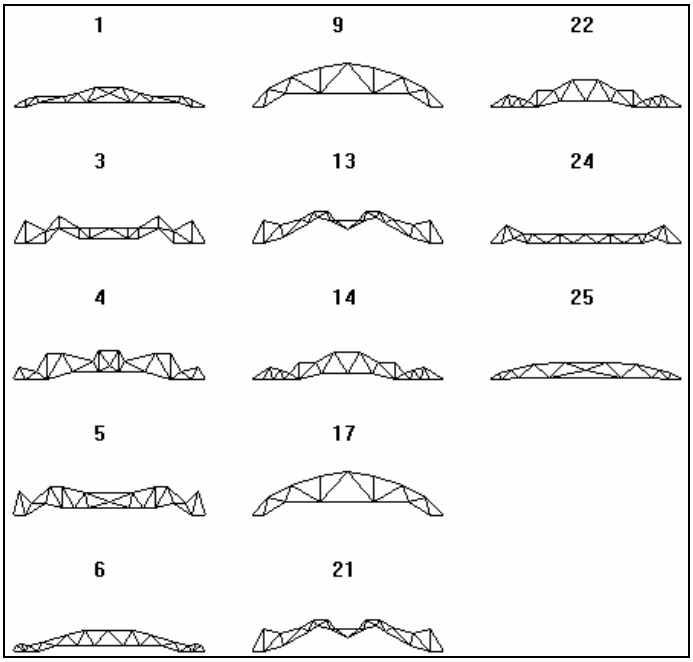


Figure C.8. BP-RSR predictions for population S25M50(a) during input reproduction trials.

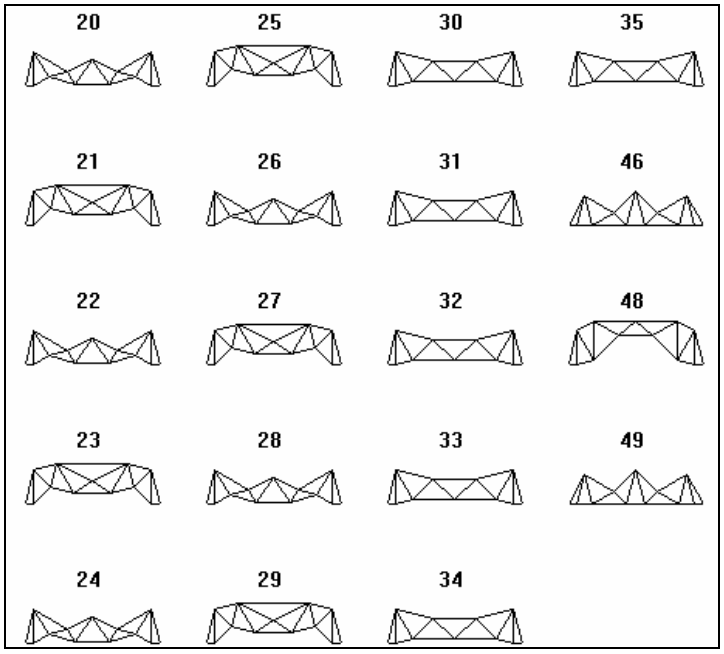


Figure C.9. KSOM predictions for population S50M25(a) during input reproduction trials.

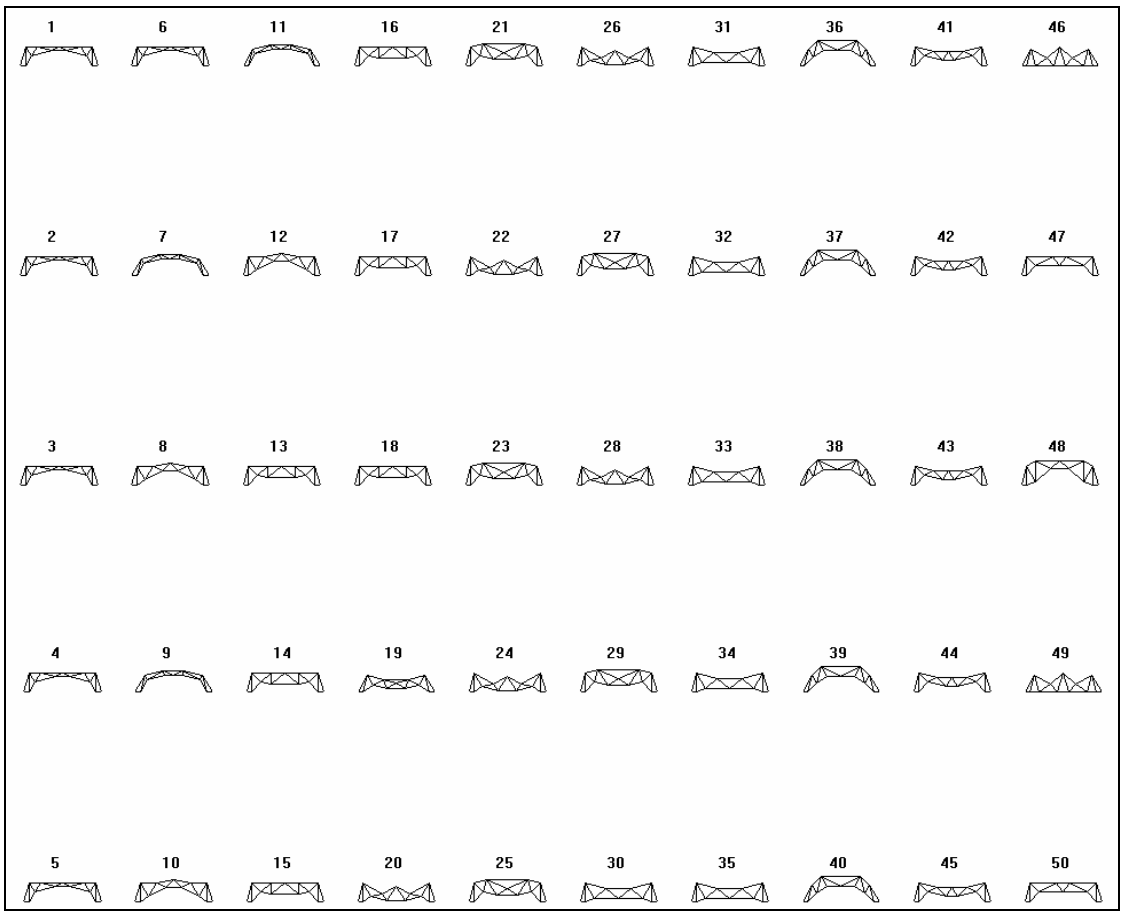


Figure C.10. RSR predictions for population S50M25(a) during input reproduction trials.

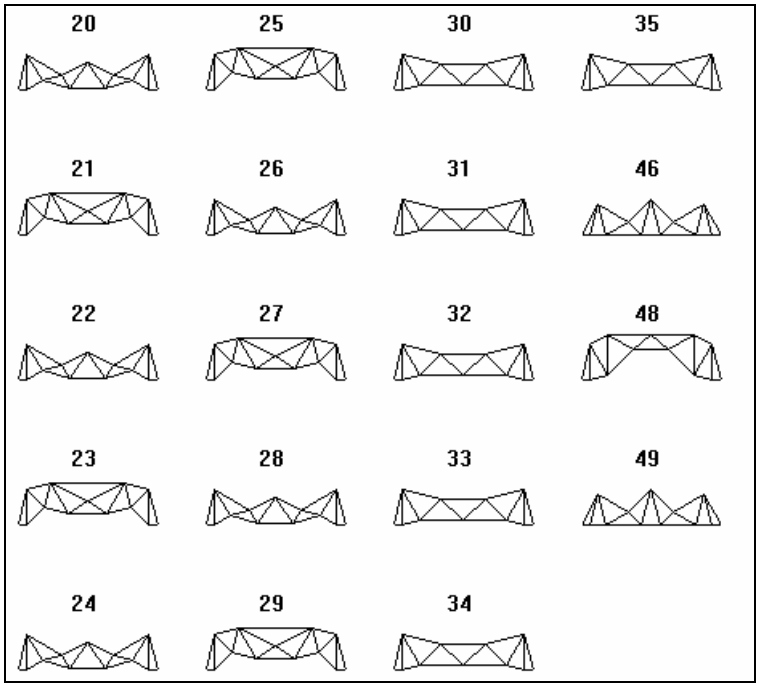


Figure C.11. BPNN predictions for population S50M25(a) during input reproduction trials.

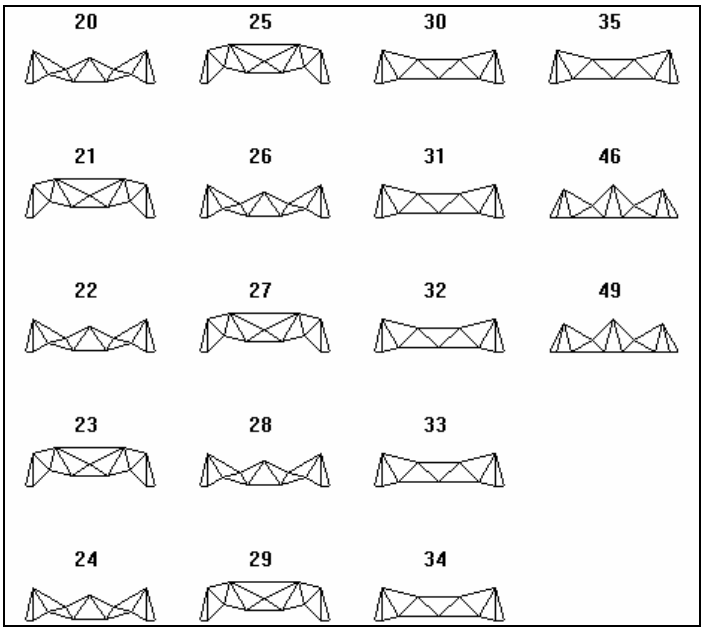


Figure C.12. BP-RSR predictions for population S50M25(a) during input reproduction trials.

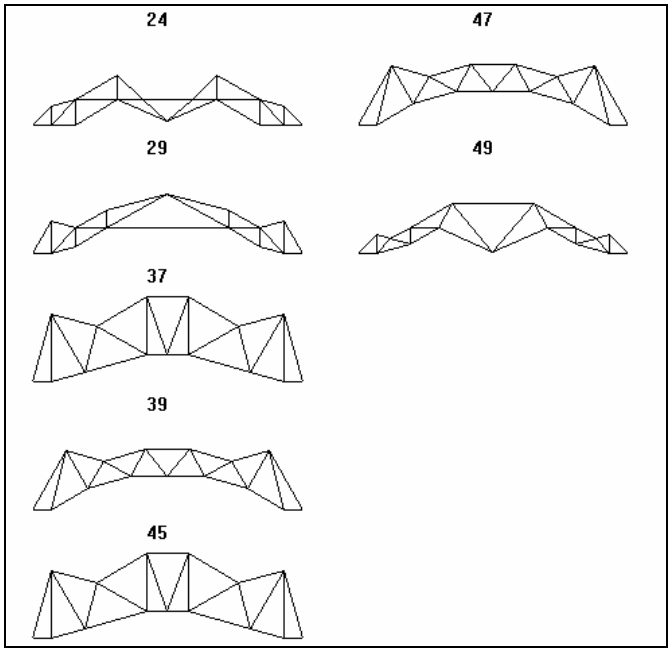


Figure C.13. KSOM predictions for population S50M50(a) during input reproduction trials.

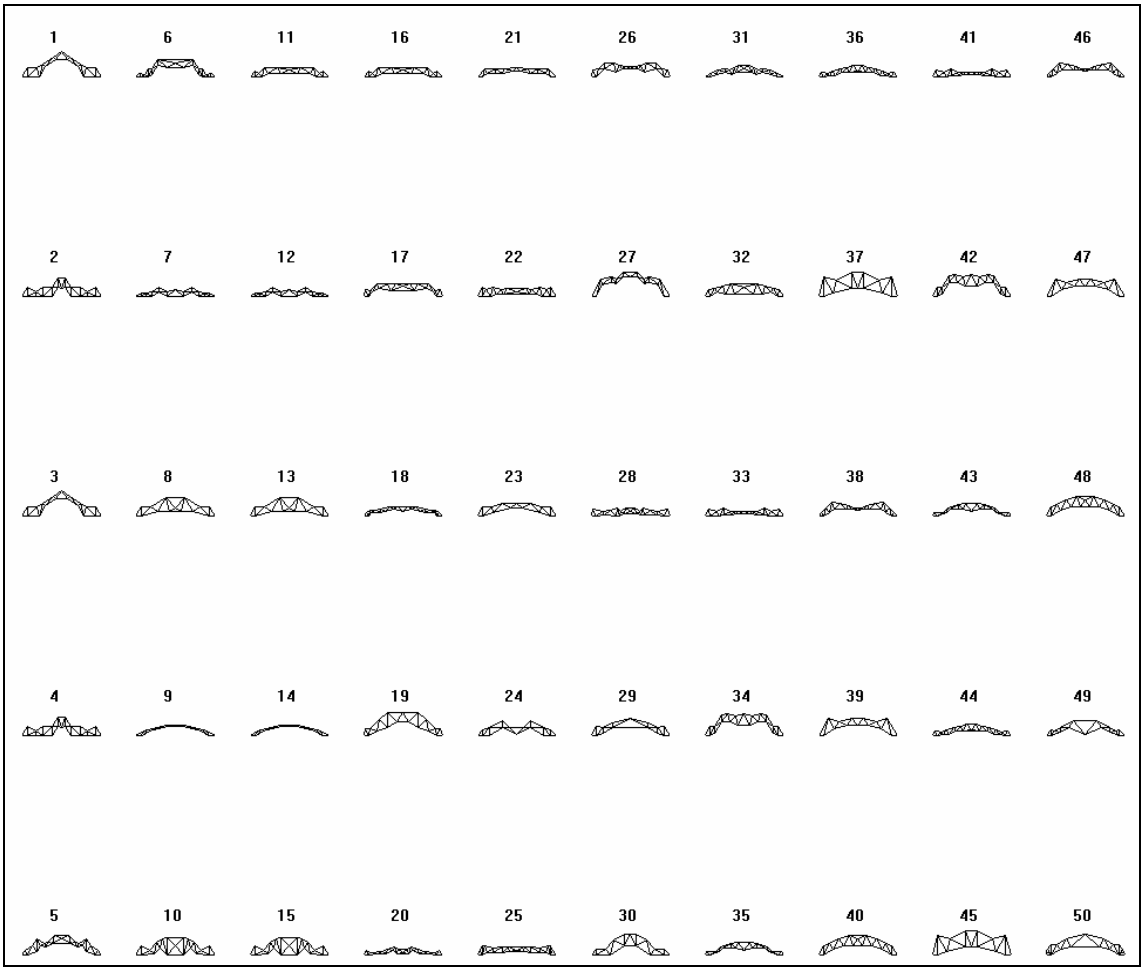


Figure C.14. RSR predictions for population S50M50(a) during input reproduction trials.

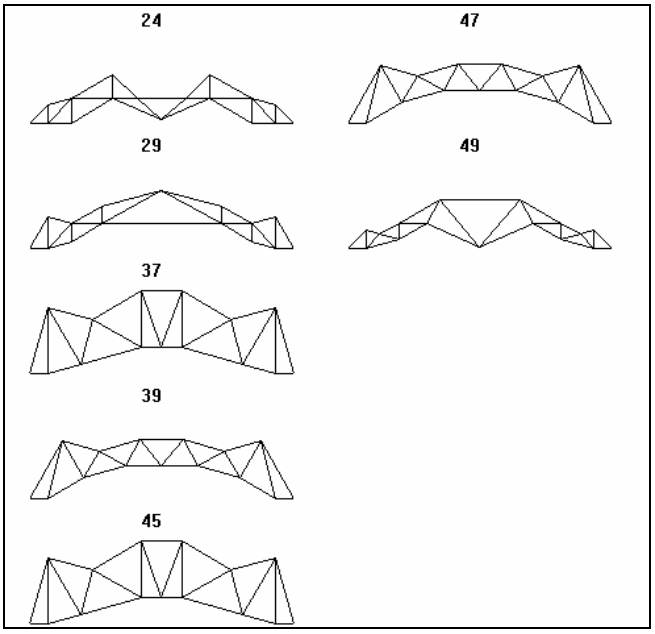


Figure C.15. BPNN predictions for population S50M50(a) during input reproduction trials.

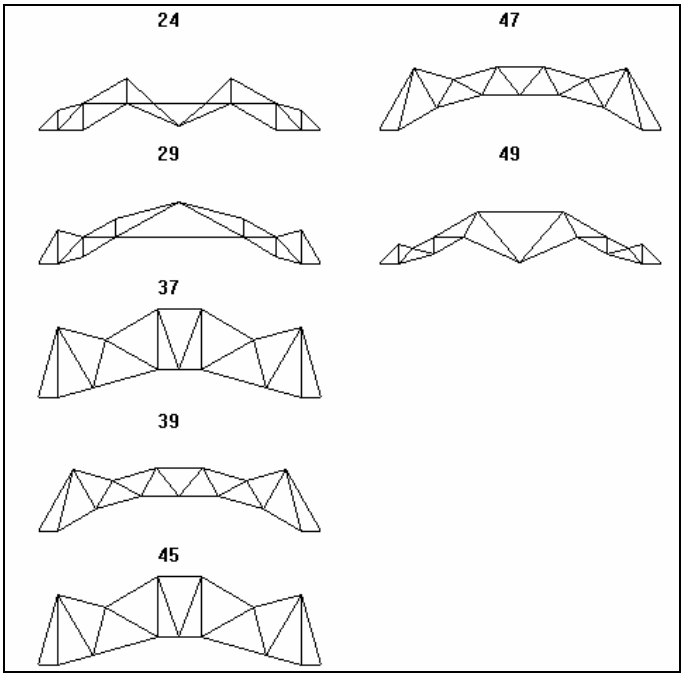


Figure C.16. BP-RSR predictions for population S50M50(a) during input reproduction trials.

Preference Detection Trials

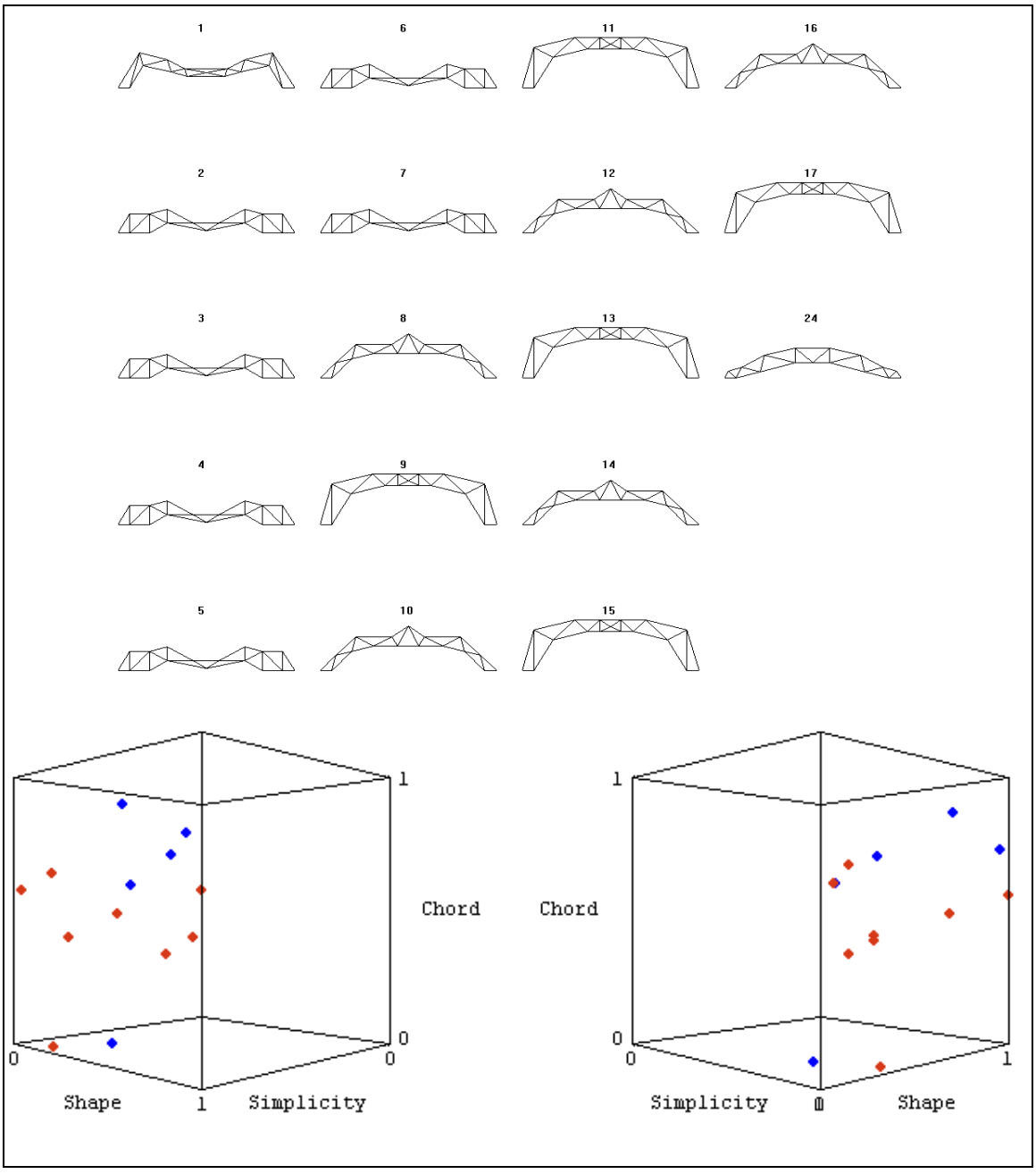


Figure C.11. KSOM predictions for population S25M35(b) during preference detection trials.

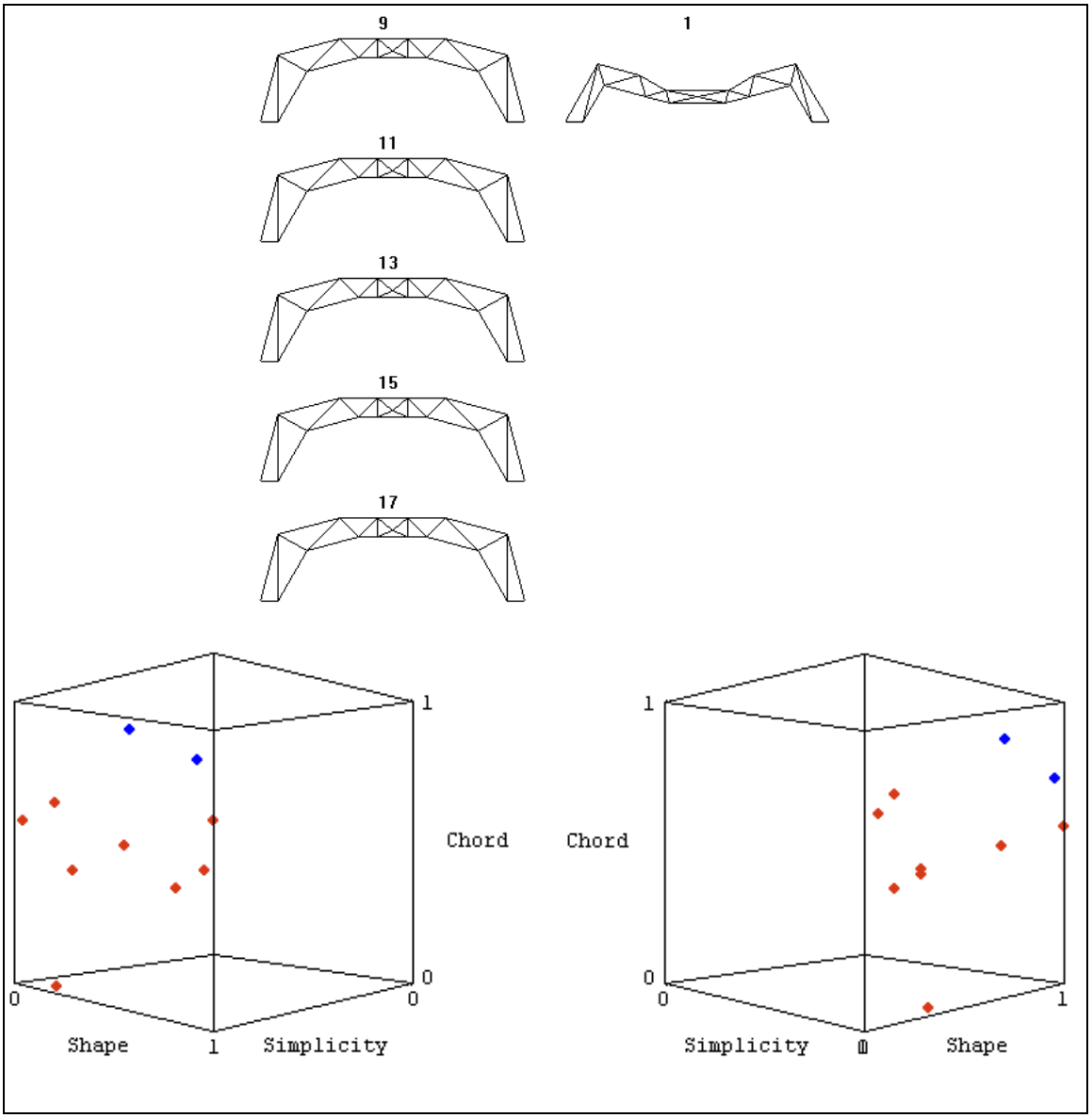


Figure C.12. RSR predictions for population S25M35(b) during preference detection trials.

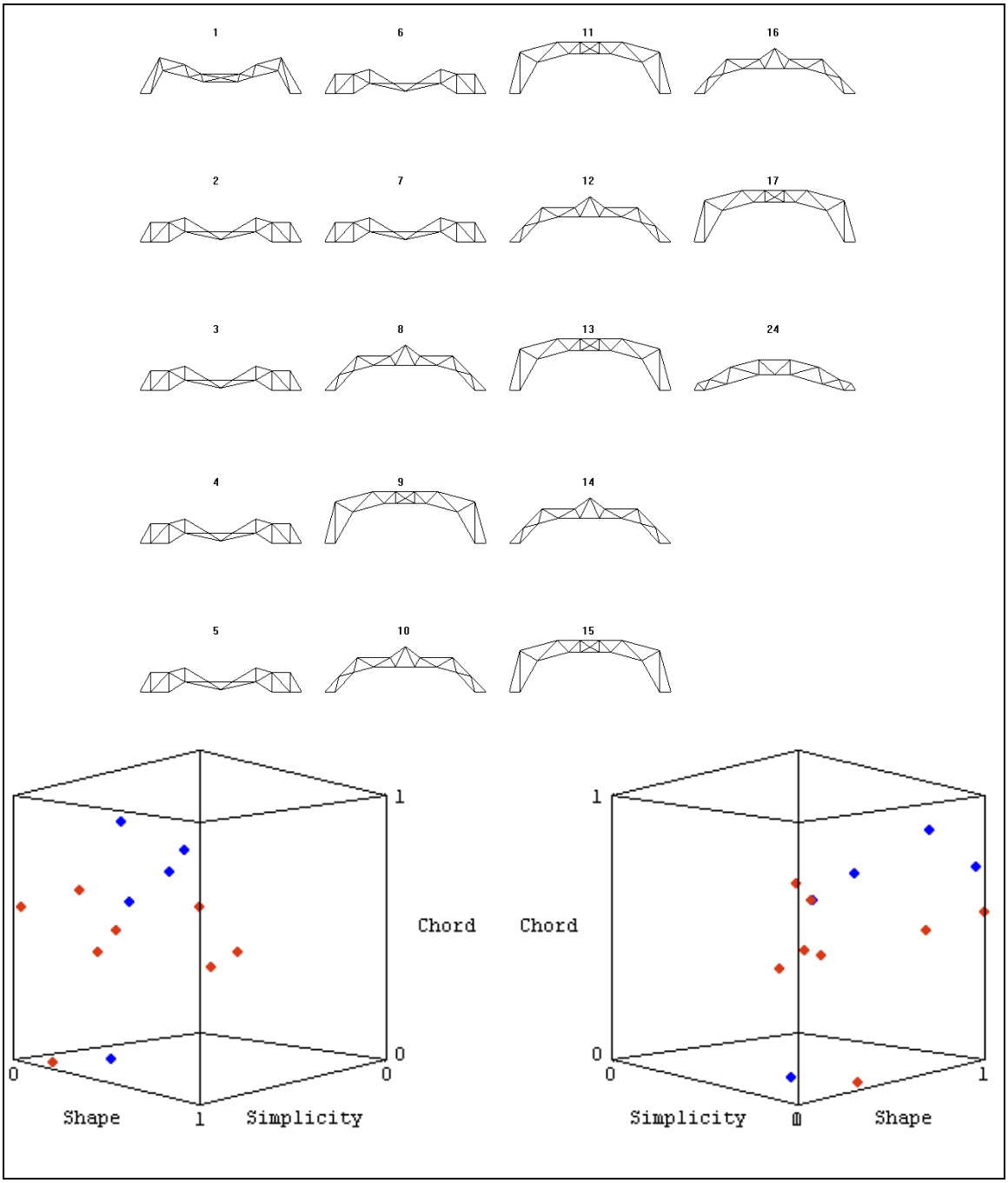


Figure C.13. BPNN predictions for population S25M35(b) during preference detection trials.

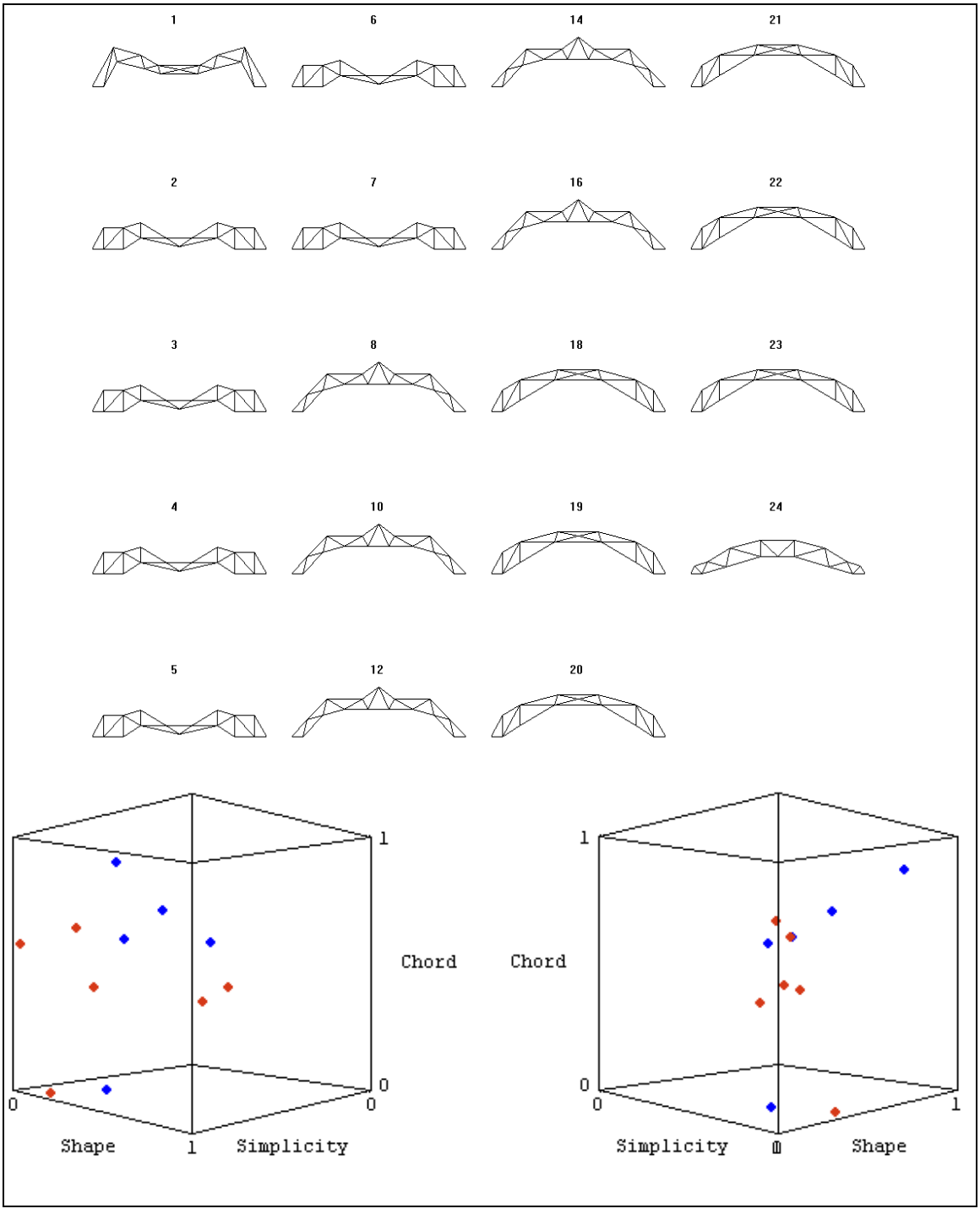


Figure C.14. BP-RSR predictions for population S25M35(b) during preference detection trials.

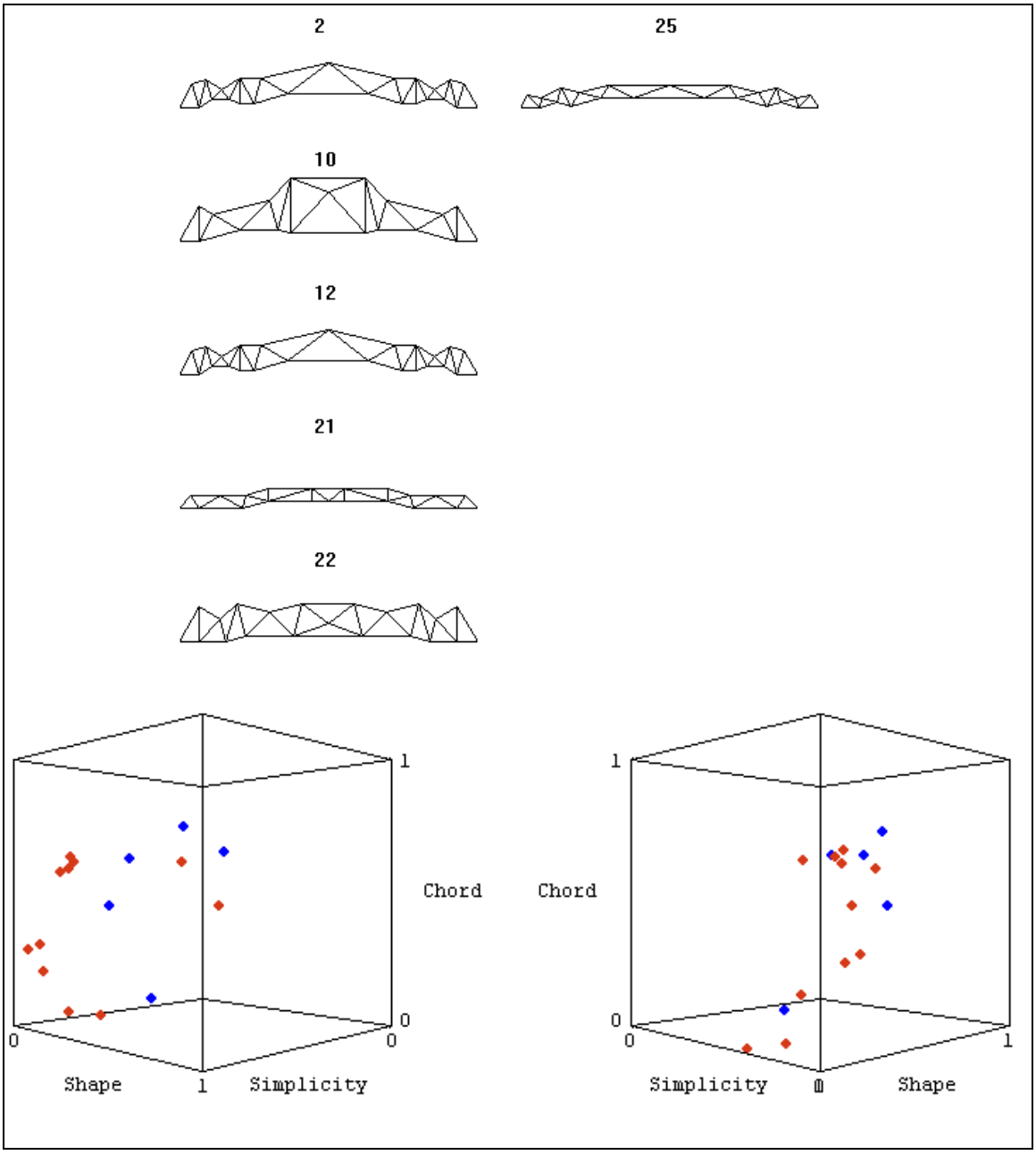


Figure C.15. KSOM predictions for population S25M50(b) during preference detection trials.

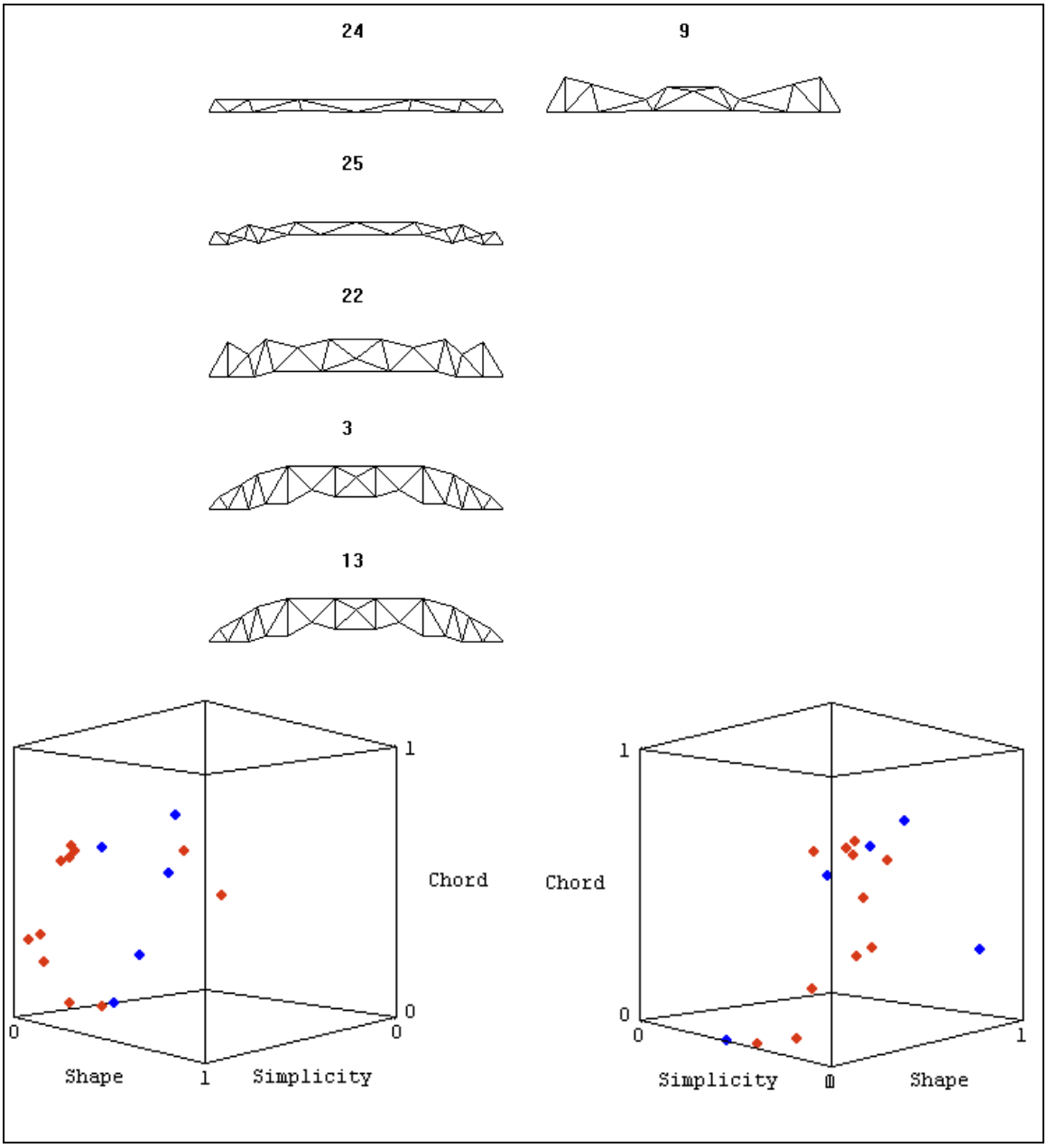


Figure C.16. RSR predictions for population S25M50(b) during preference detection trials.

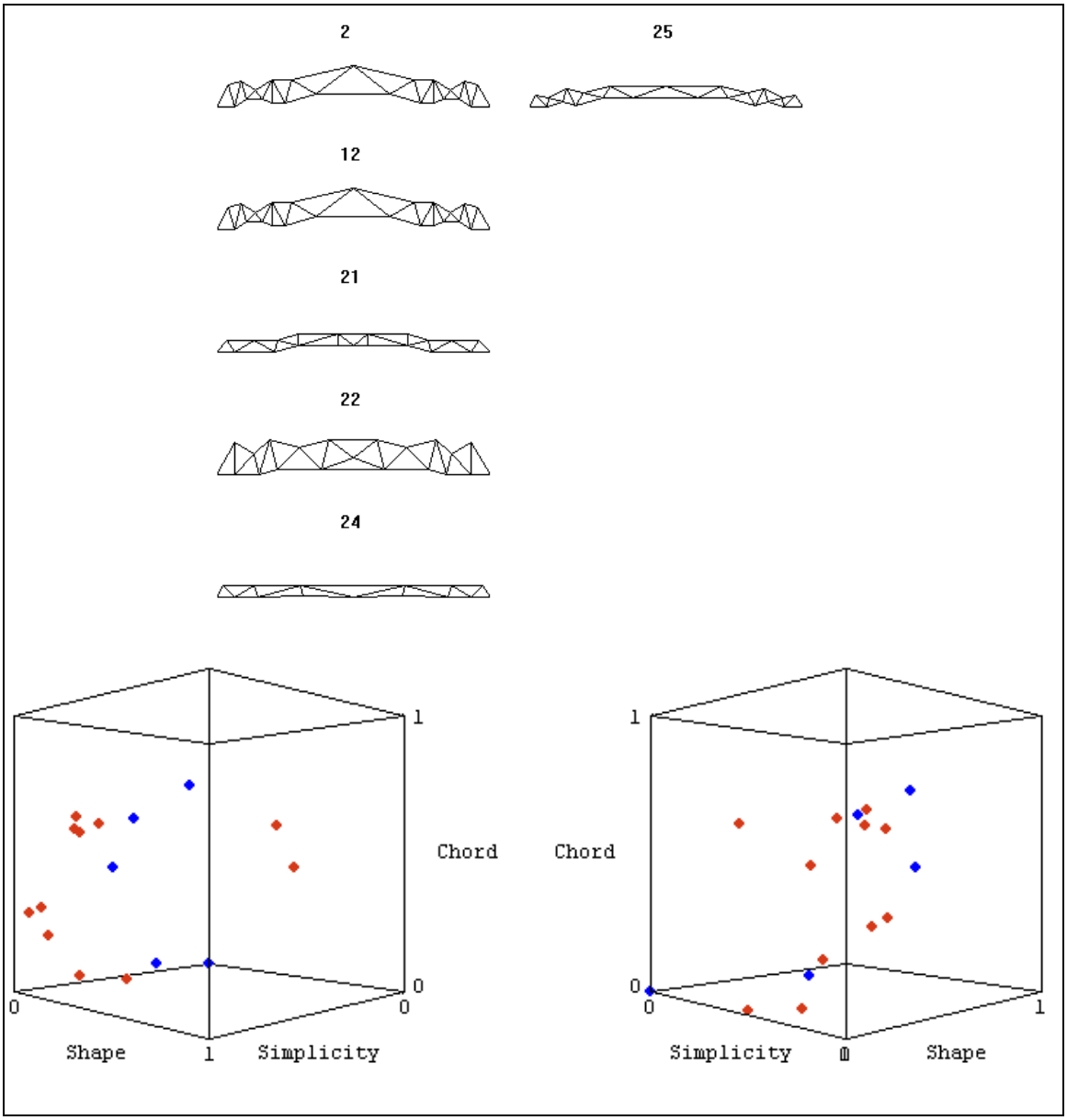


Figure C.17. BPNN predictions for population S25M50(b) during preference detection trials.

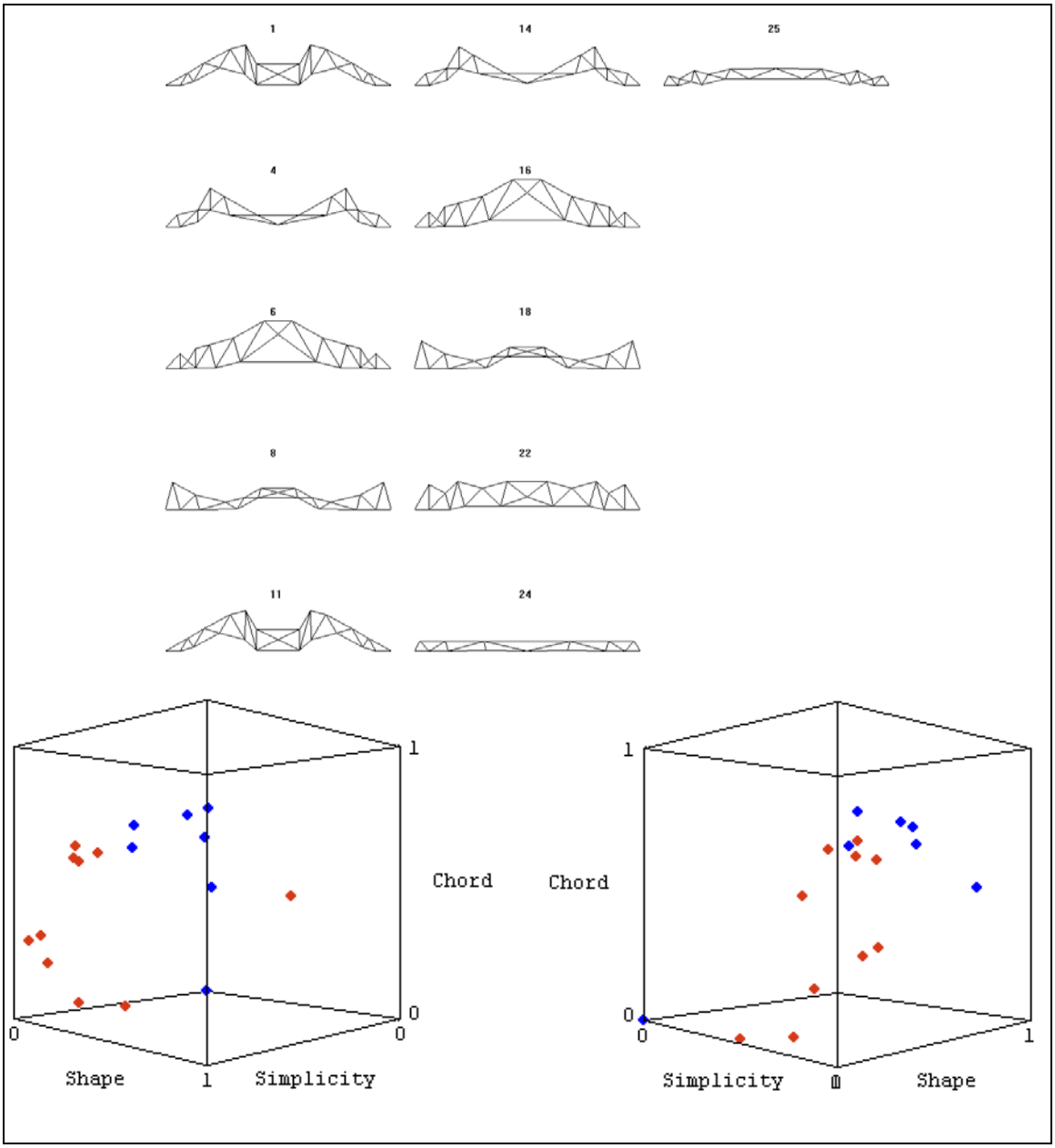


Figure C.18. BP-RSR predictions for population S25M50(b) during preference detection trials.

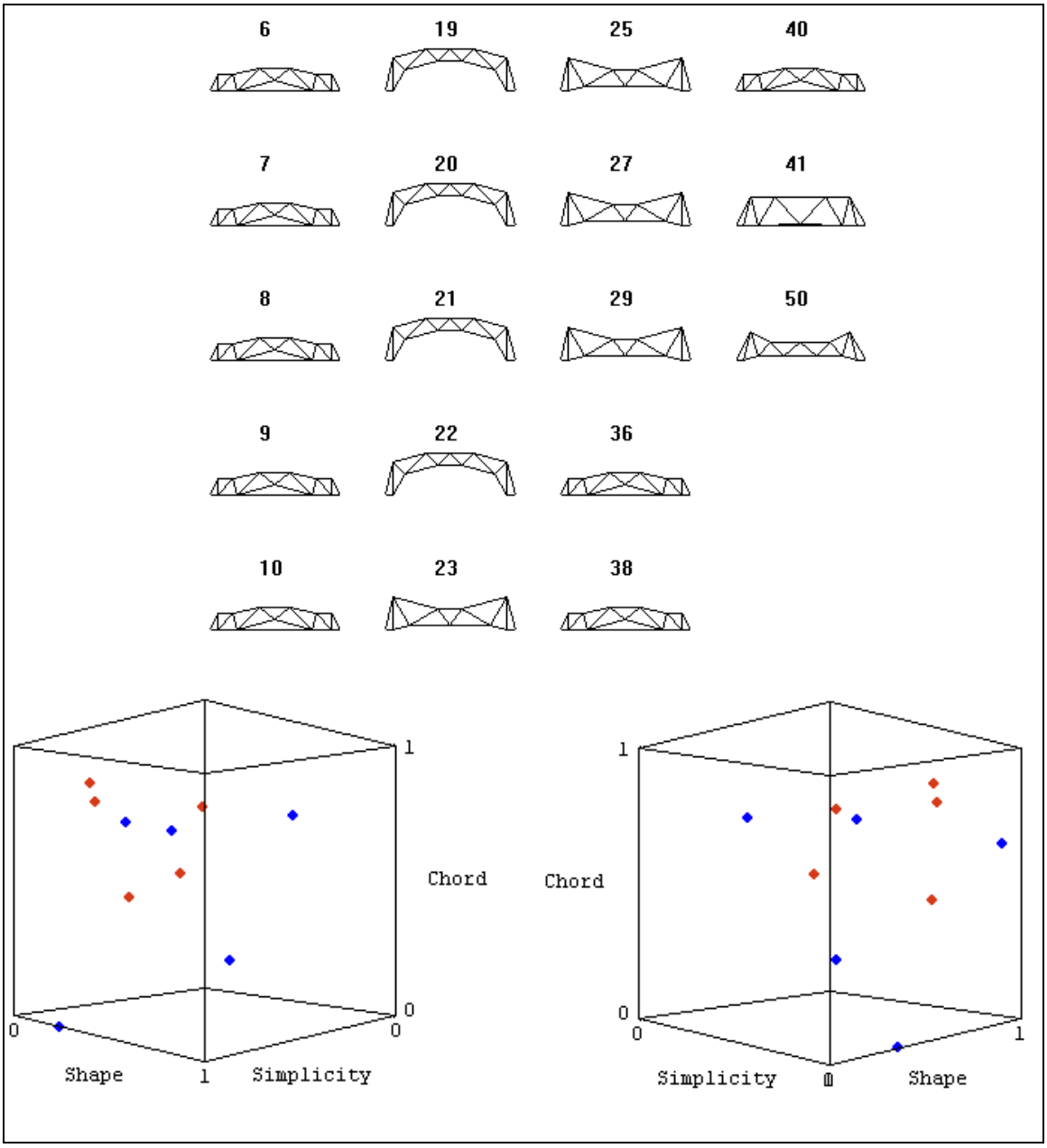


Figure C.19. KSOM predictions for population S50M25(b) during preference detection trials.

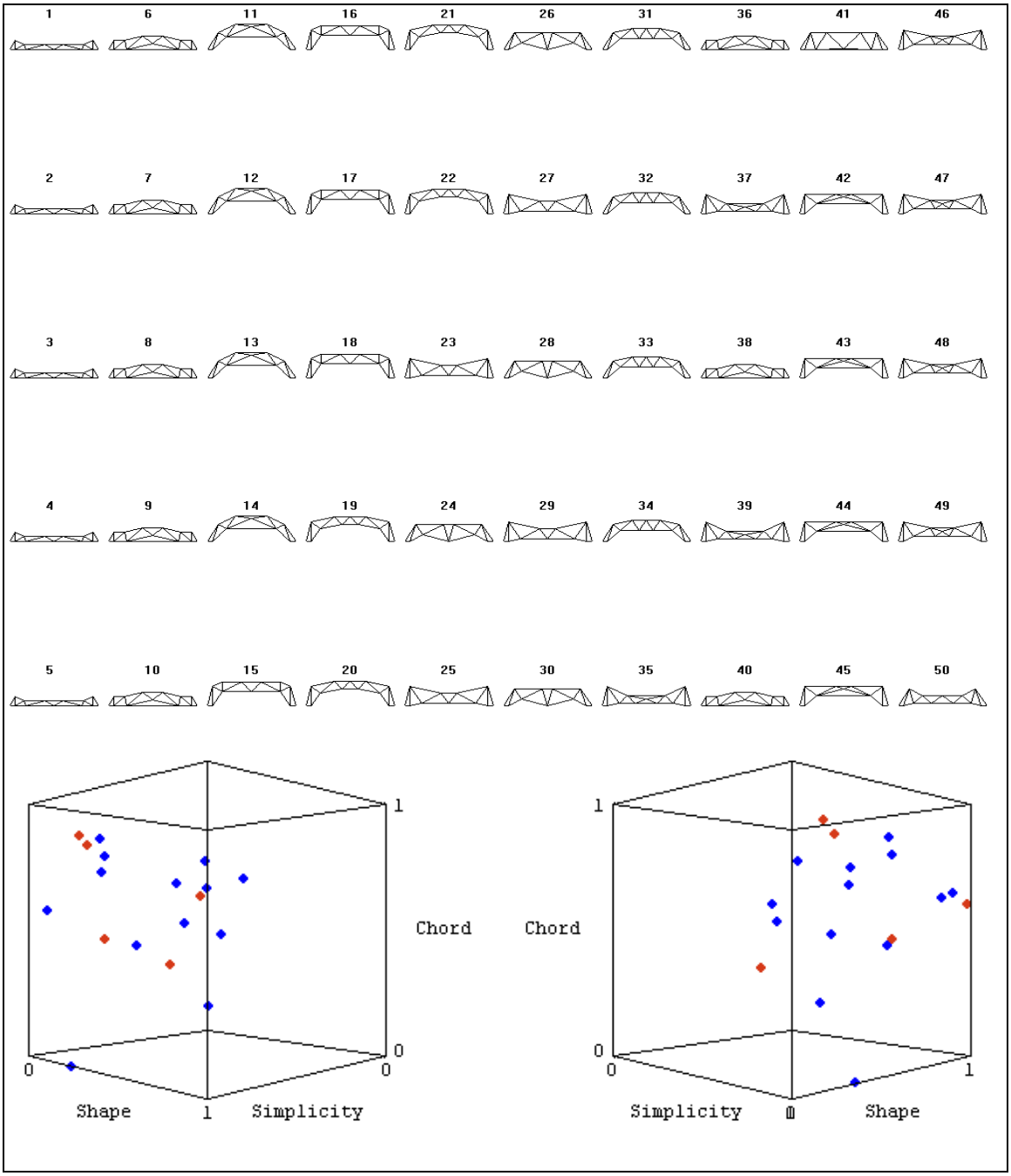


Figure C.20. RSR predictions for population S50M25(b) during preference detection trials.

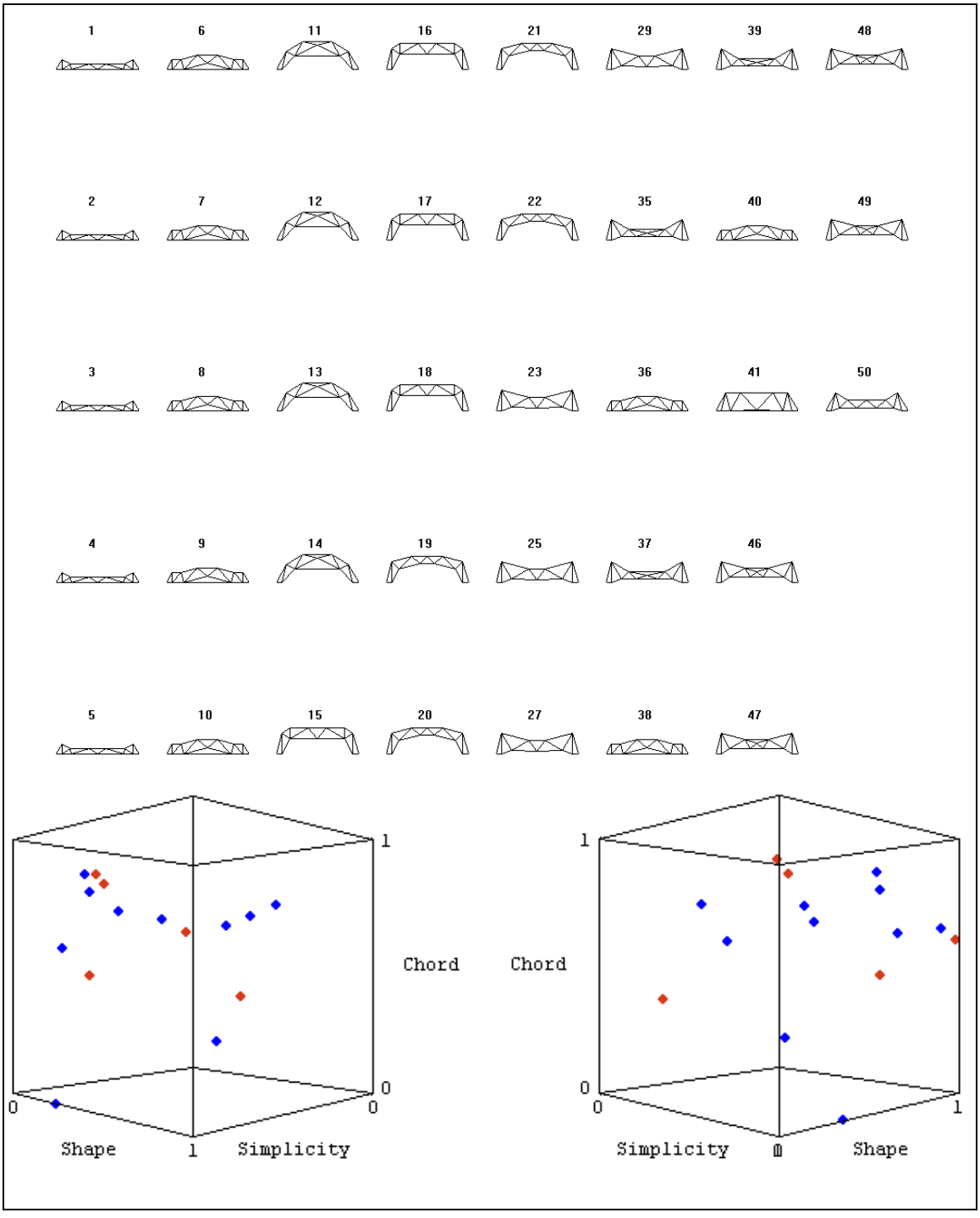


Figure C.21. BPNN predictions for population S50M25(b) during preference detection trials.

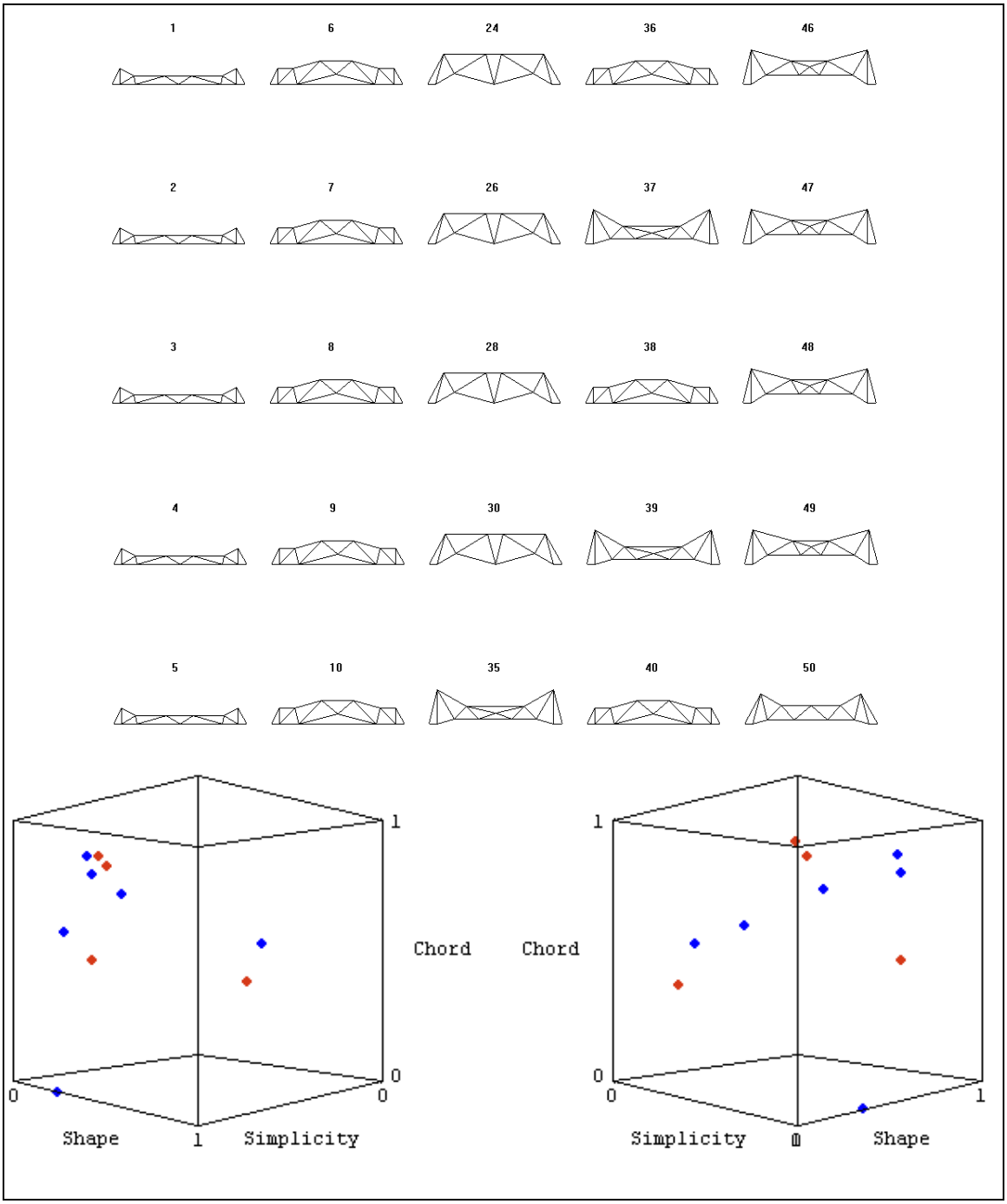


Figure C.22. BP-RSR predictions for population S50M25(b) during preference detection trials.

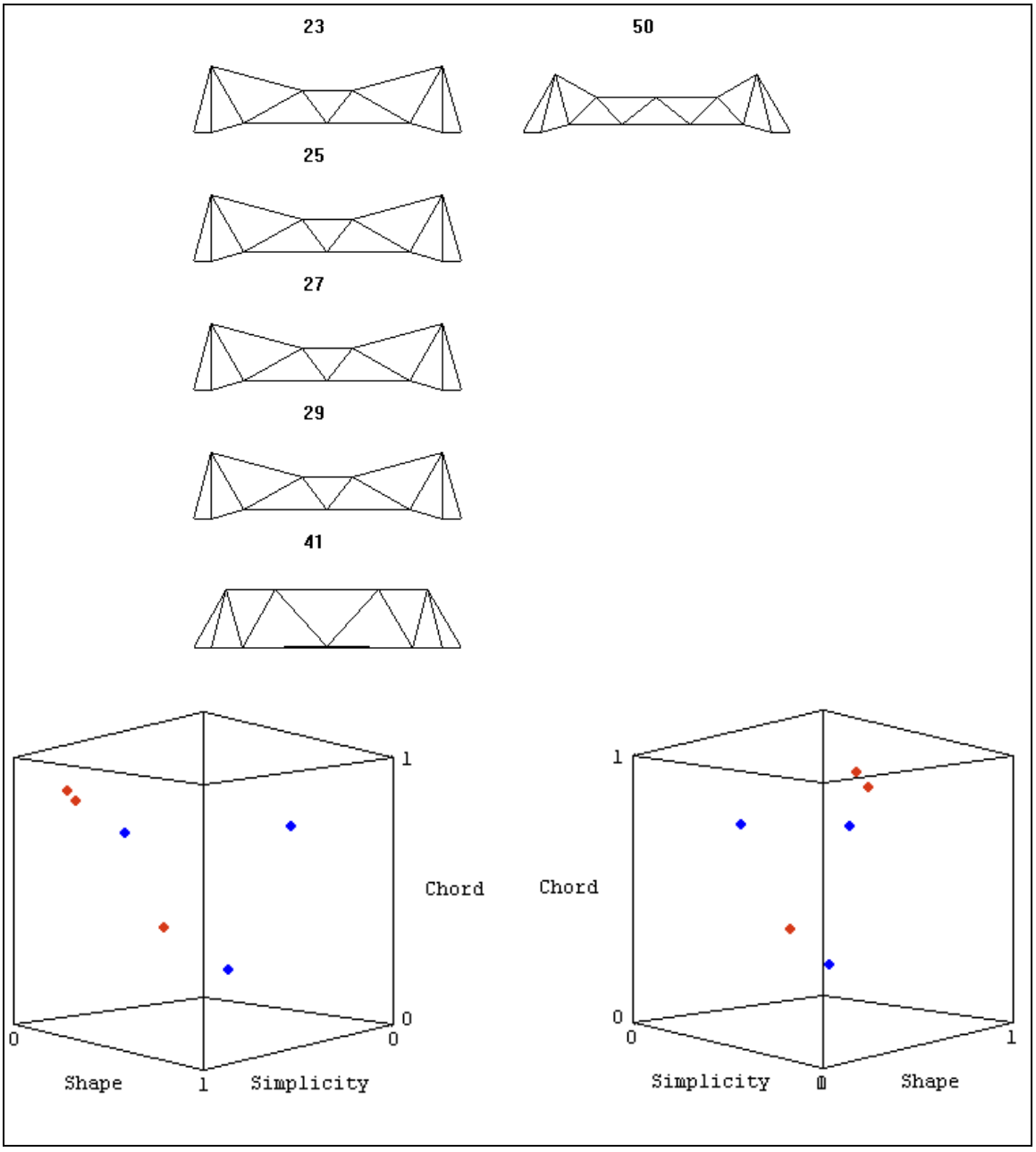


Figure C.23. KSOM predictions for population S50M25(b) during preference detection trials using alternate user selections.

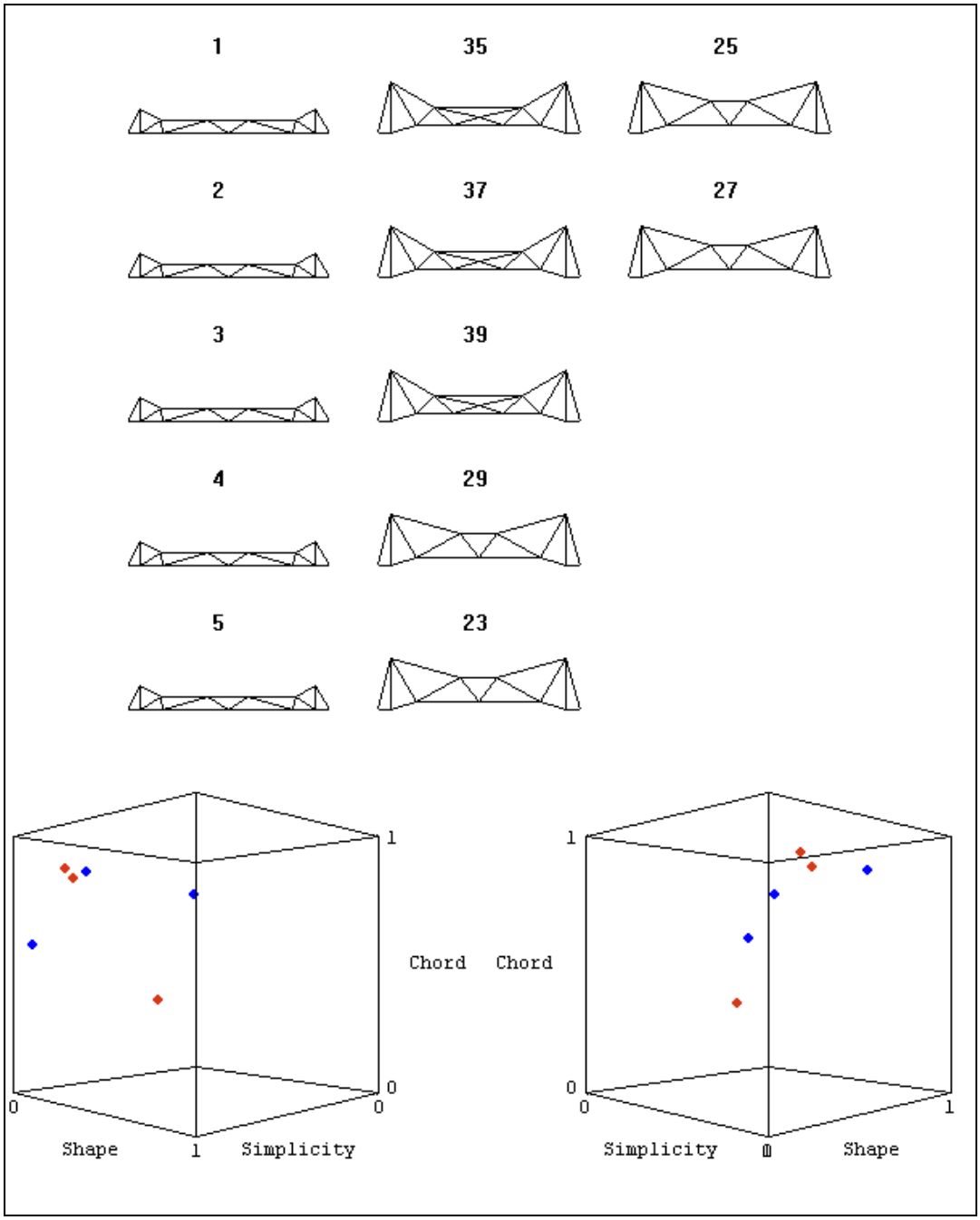


Figure C.24. RSR predictions for population S50M25(b) during preference detection trials using alternate user selections.

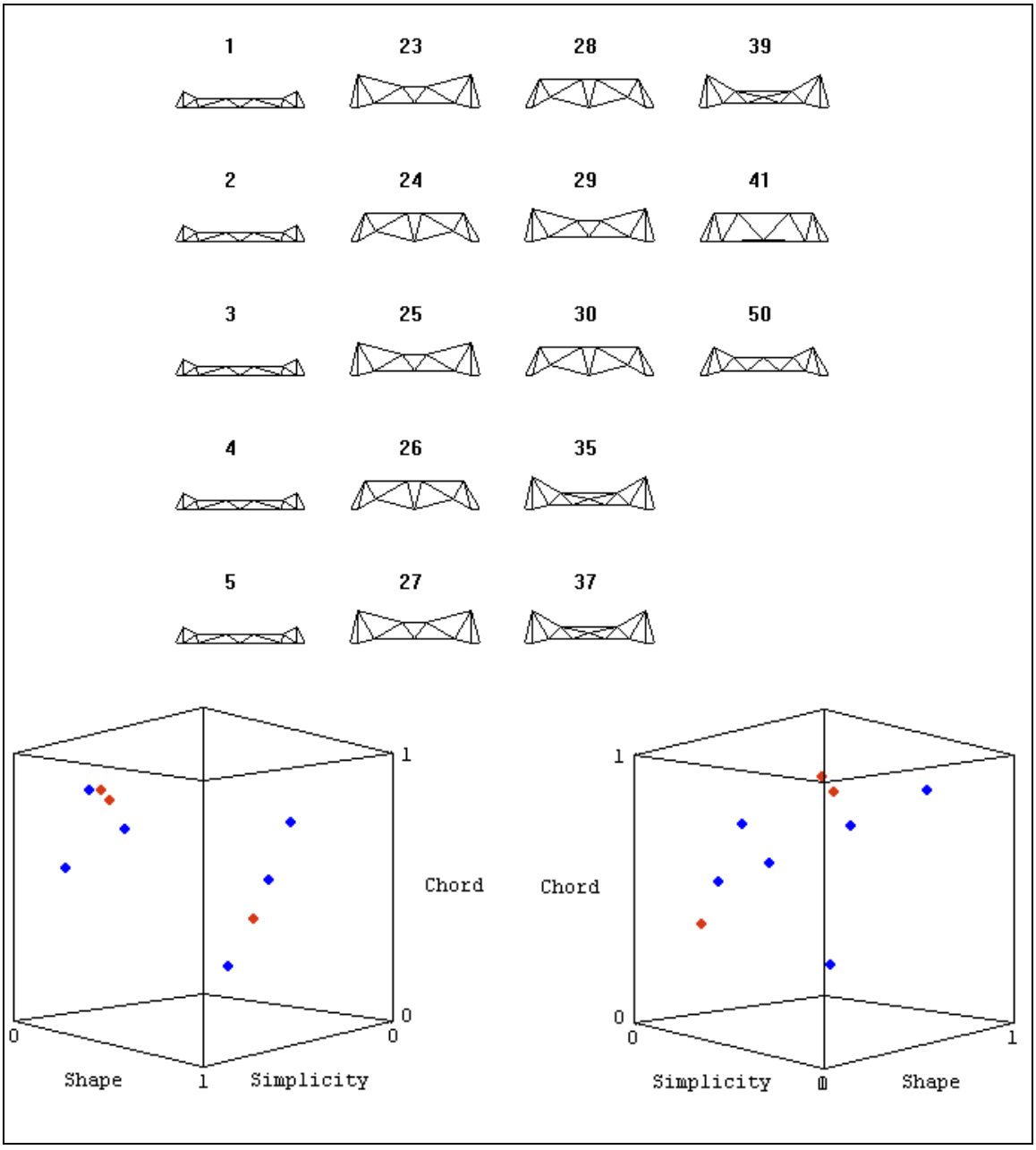


Figure C.25. BPNN predictions for population S50M25(b) during preference detection trials using alternate user selections.

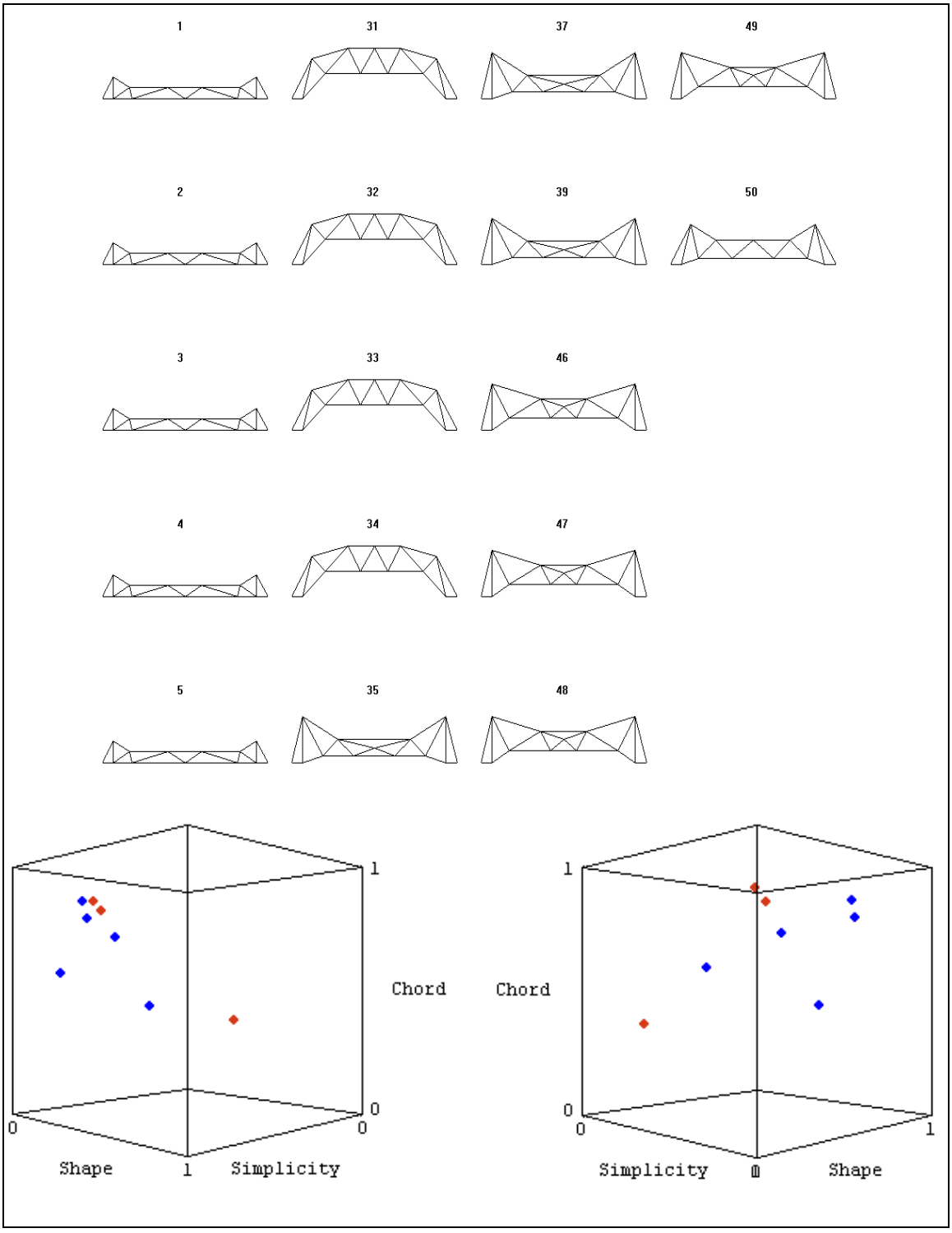


Figure C.26. BP-RSR predictions for population S50M25(b) during preference detection trials using alternate user selections.

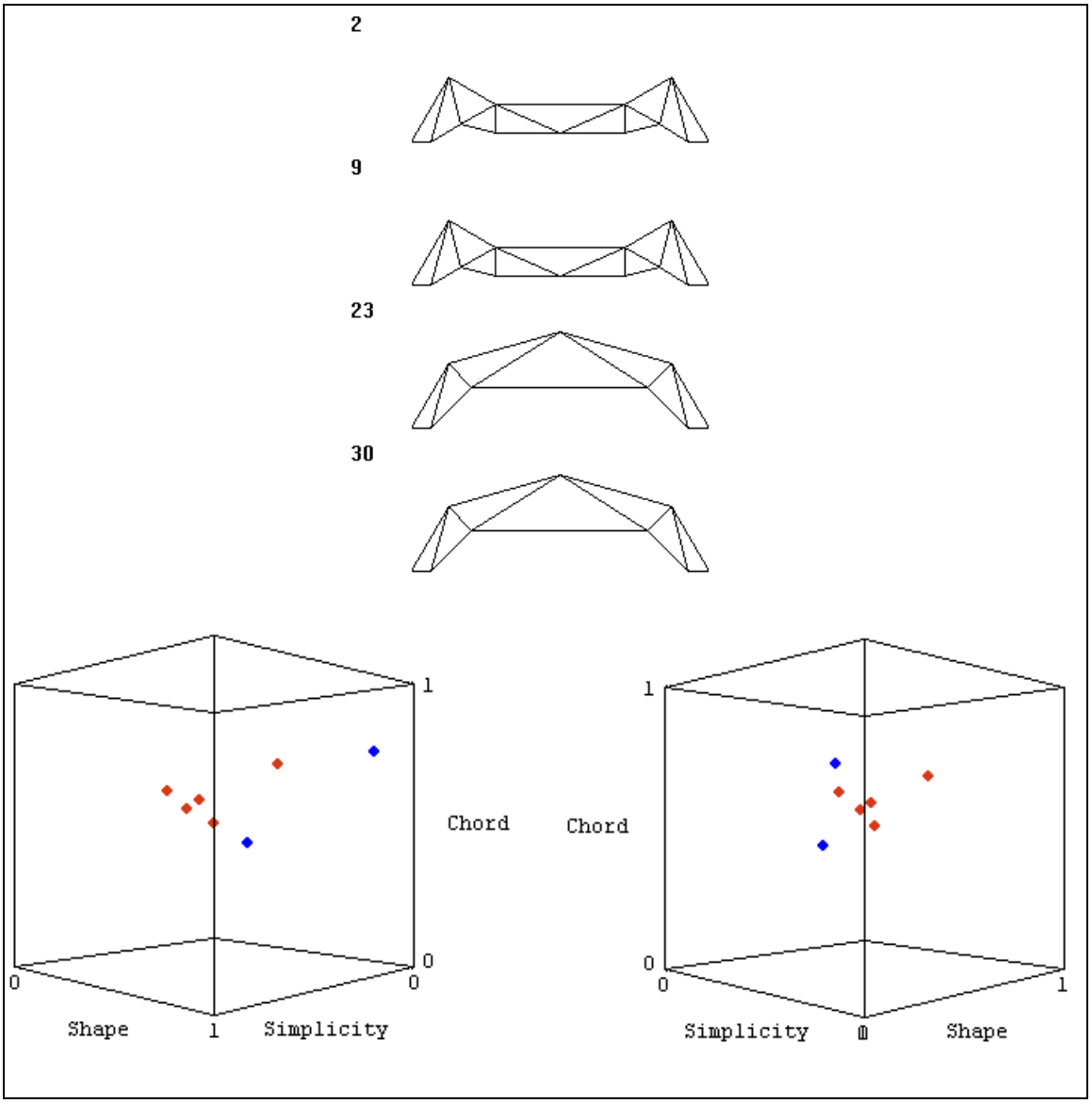


Figure C.27. KSOM predictions for population S50M50(b) during preference detection trials.

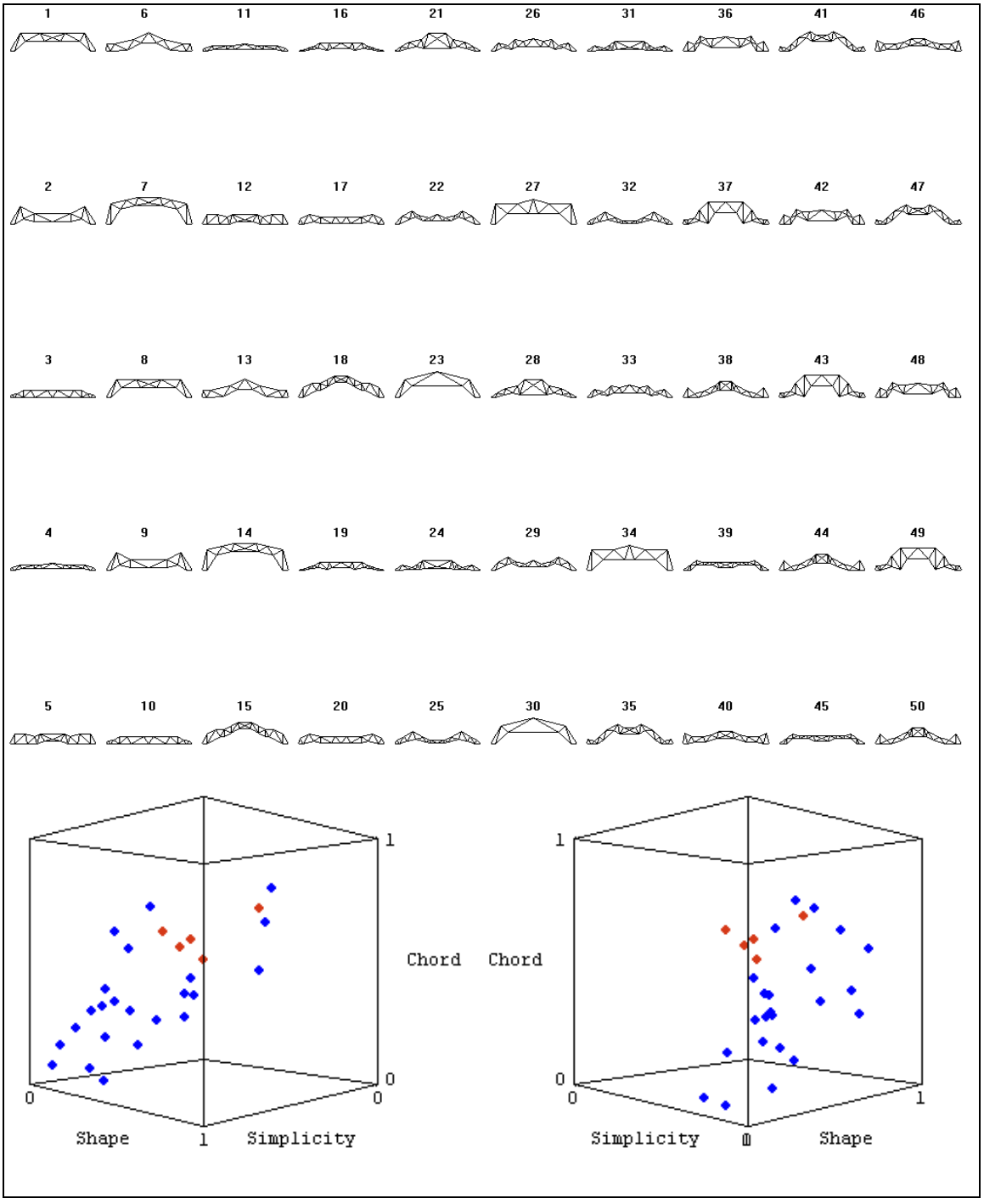


Figure C.28. RSR predictions for population S50M50(b) during preference detection trials.

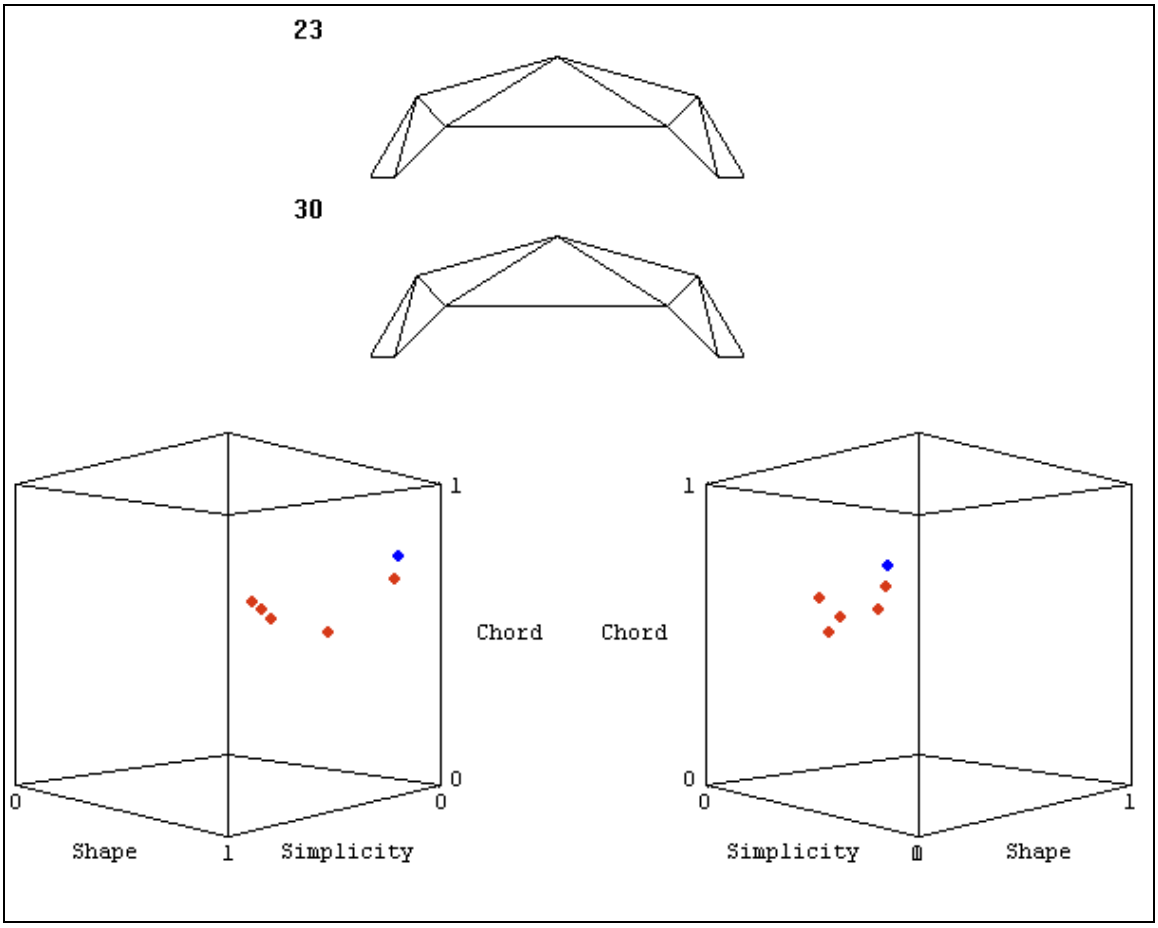


Figure C.29. BPNN predictions for population S50M50(b) during preference detection trials.

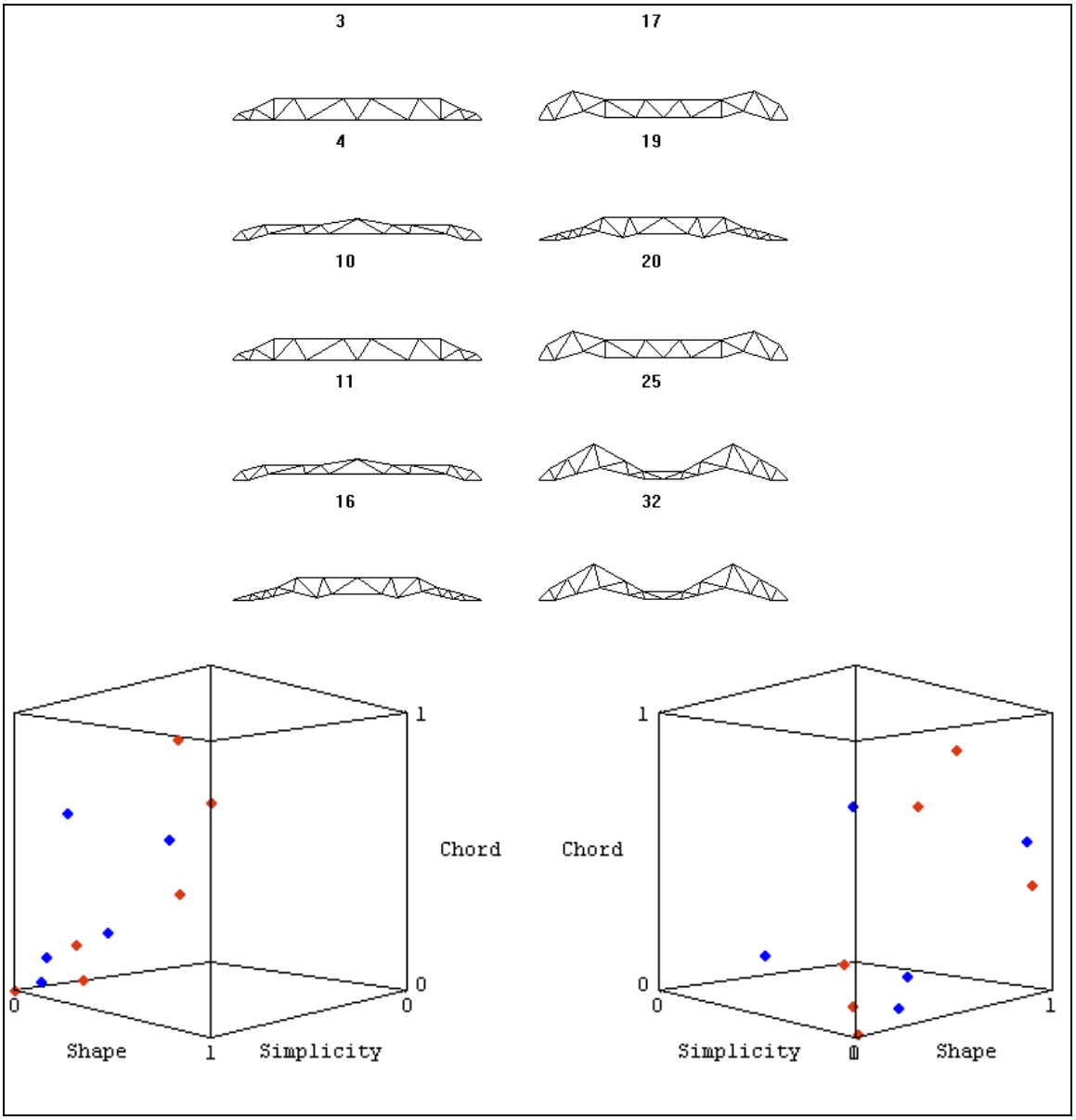


Figure C.30. BP-RSR predictions for population S50M50(b) during preference detection trials.

APPENDIX D

This appendix contains additional results for the mechanism and preference emphasis results of Chapter V. This chapter concerned the implementation of preference detection using the feature vector, one-dimensional Kohonen's Self-Organizing Map, and hybrid Back-prop Neural Network with Rough Set Reduct techniques discussed in the previous chapters. The **dominate( )** function as well as the **fitness**, **rankFitness**, and **compositePen** variables were considered as locations to include user preferences.

Mechanism Trials

Figures D.1 through D.6 present the unique rank one trusses for the 10th generation of population 40by15 for each of the six preference formulations considered during the mechanism trials. Figures D.7 through D.12 display equivalent results for 60by15par. User selections for these populations are presented in Figures 5.3 and 5.4.

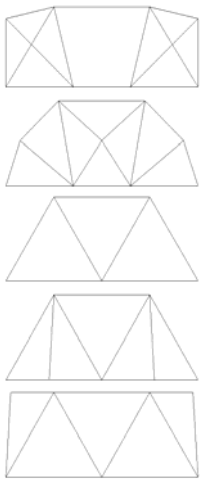


Figure D.1. 10th generation, rank one trusses for population 40by15 discovered during the Baseline trial.

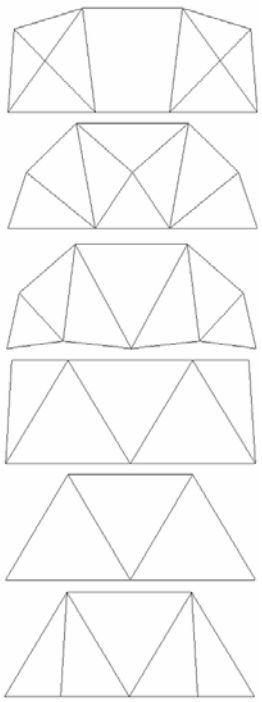


Figure D.2. 10th generation, rank one trusses for population 40by15 discovered during the Dominate trial.

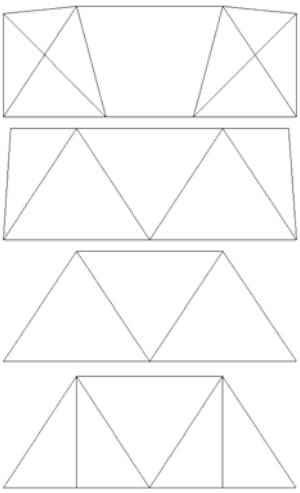


Figure D.3. 10th generation, rank one trusses for population 40by15 discovered during the CompositeOnly trial.

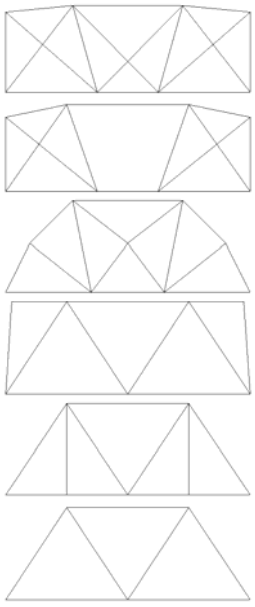


Figure D.4. 10th generation, rank one trusses for population 40by15 discovered during the Composite10 trial.

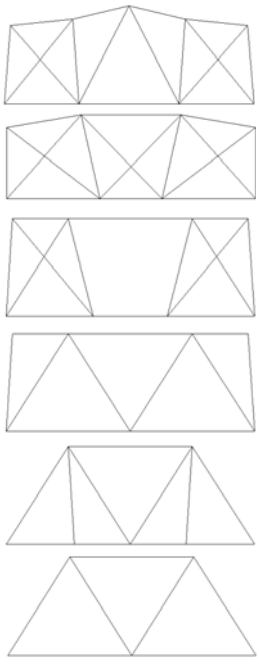


Figure D.5. 10th generation, rank one trusses for population 40by15 discovered during the Composite50 trial.

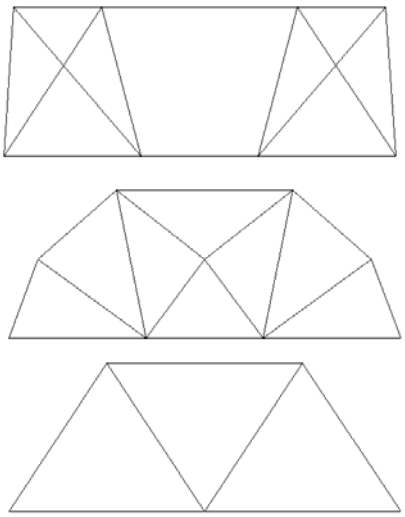


Figure D.6. 10th generation, rank one trusses for population 40by15 discovered during the Fitness trial.

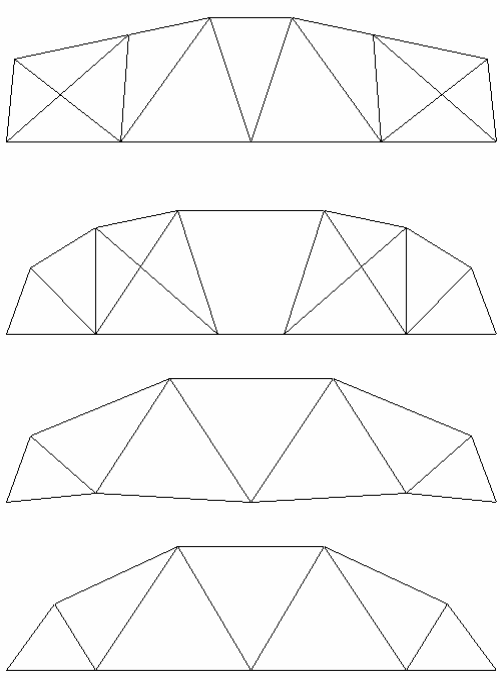


Figure D.7. 10th generation, rank one trusses for population 60by15Par discovered during the Baseline trial.

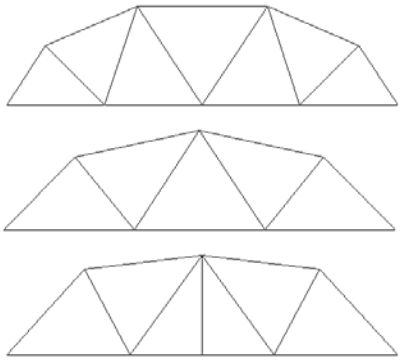


Figure D.8. 10th generation, rank one trusses for population 60by15Par discovered during the Dominate trial.

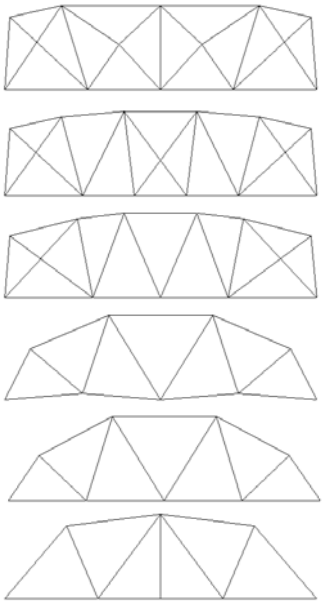


Figure D.9. 10th generation, rank one trusses for population 60by15Par discovered during the CompositeOnly trial.

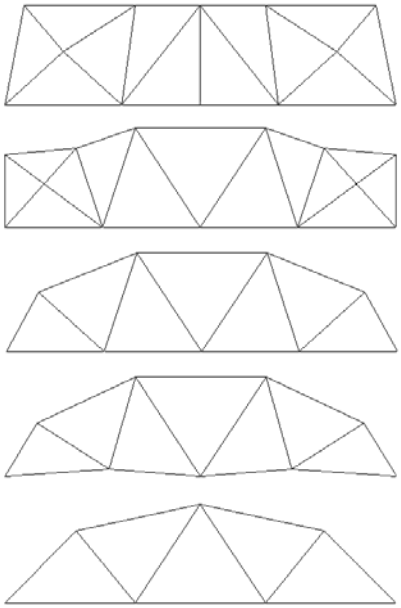


Figure D.10. 10th generation, rank one trusses for population 60by15Par discovered during the Composite10 trial.

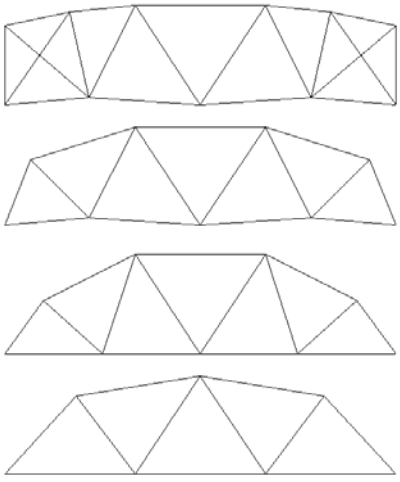


Figure D.11. 10th generation, rank one trusses for population 60by15Par discovered during the Composite50 trial.

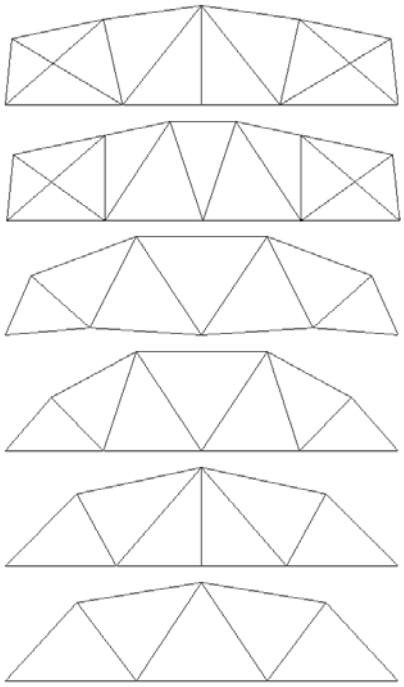


Figure D.12. 10th generation, rank one trusses for population 60by15Par discovered during the Fitness trial.

Figures D.13 and D.14 show Pareto fronts for population 40by15 and 60by15Par. The curves shown are for the Baseline and Composite10 trials and illustrate that structural performance did not suffer as a result of including user preferences in the IRR GA. Baseline analyses appear in red, and Composite10 results in yellow. Figures D.15 and D.16 show Pareto fronts for all of the mechanism trials for these two populations. These curves help show how adding user preferences increased the IRR GA's solution diversity. All units for these Figures are in pounds and inches.

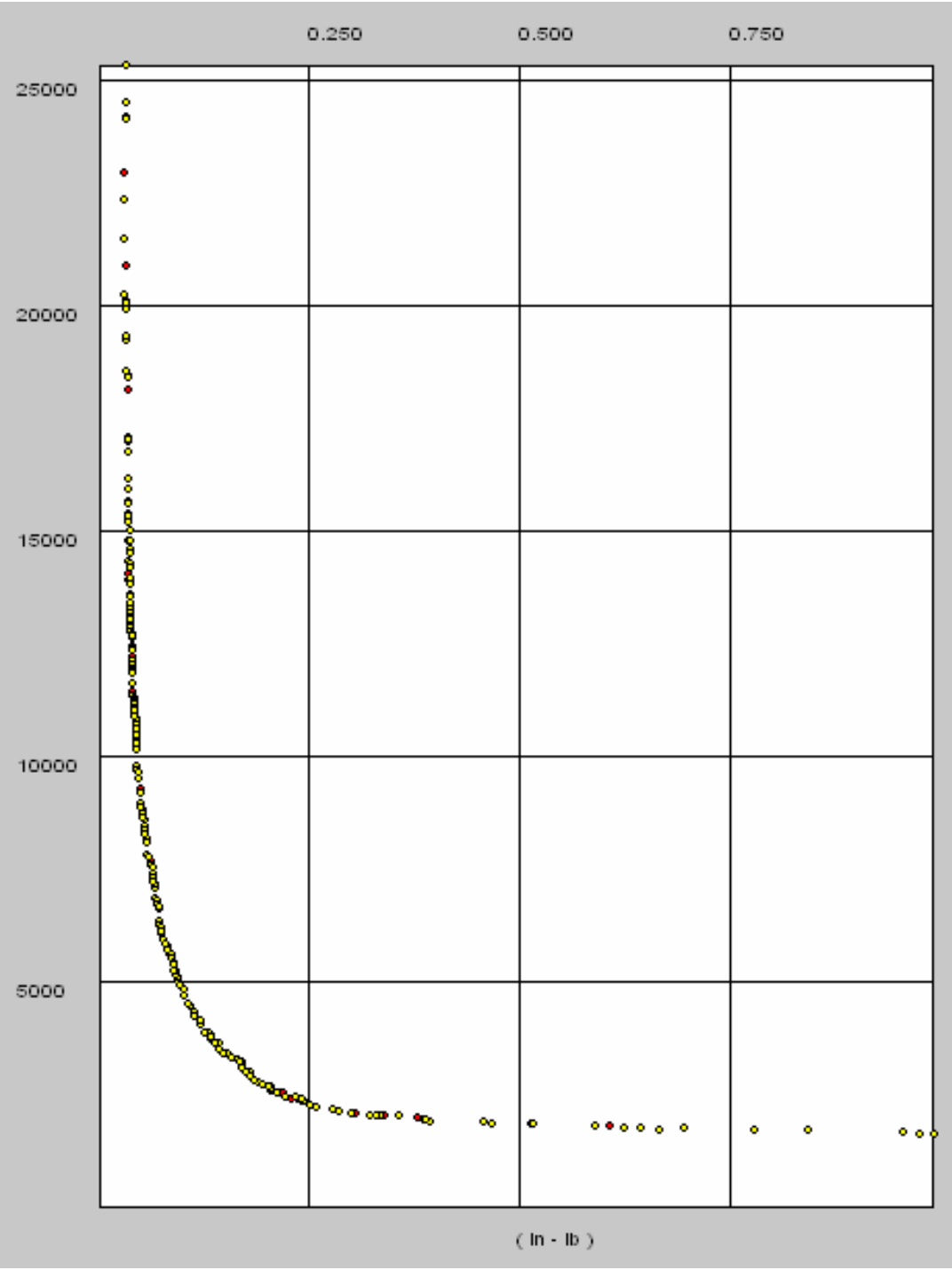


Figure D.13. 40by15 Pareto curves for the Baseline (red) and Composite10 (yellow) trials.

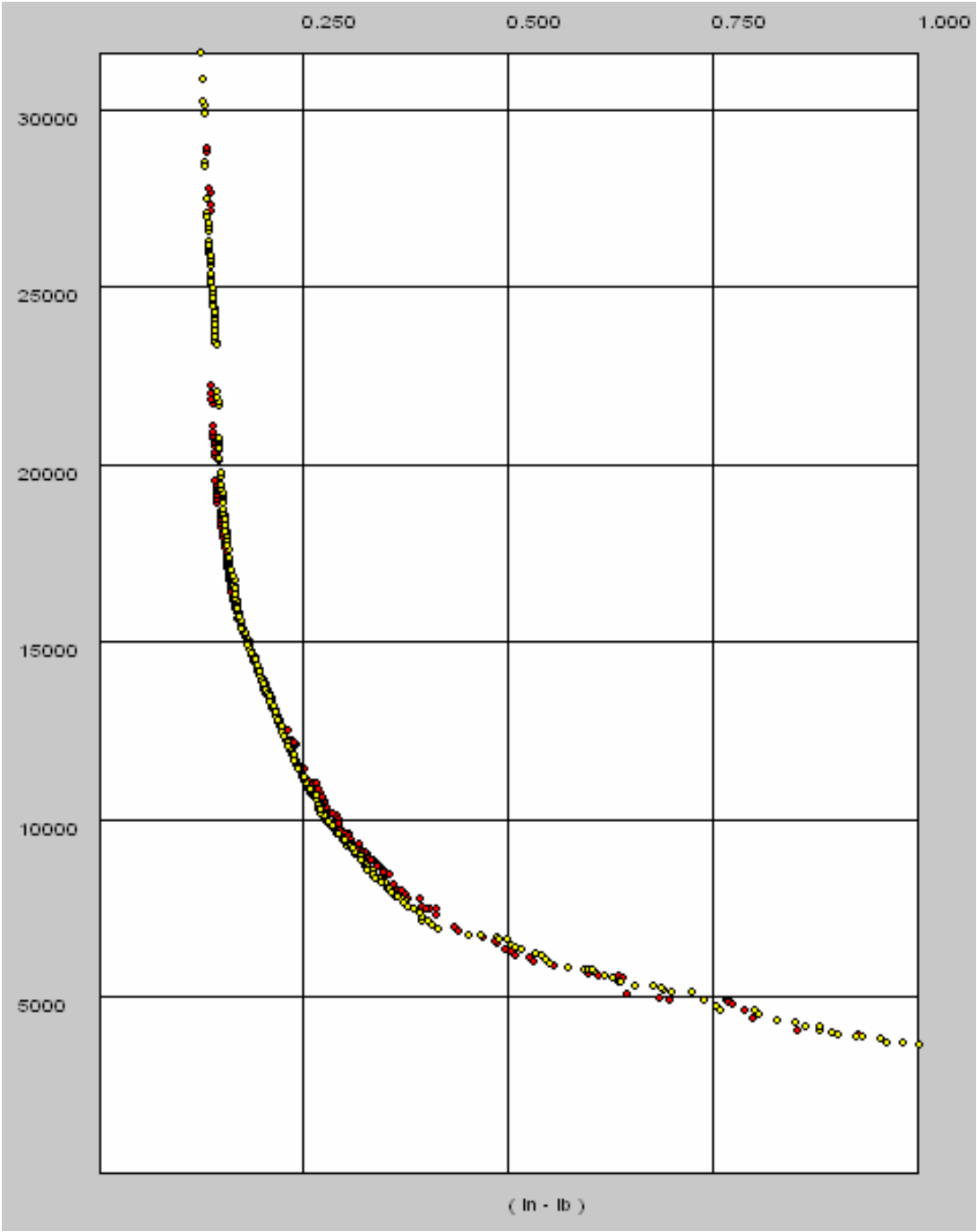


Figure D.14. 60by15Par Pareto curves for the Baseline (red) and Composite10 (yellow) trials.

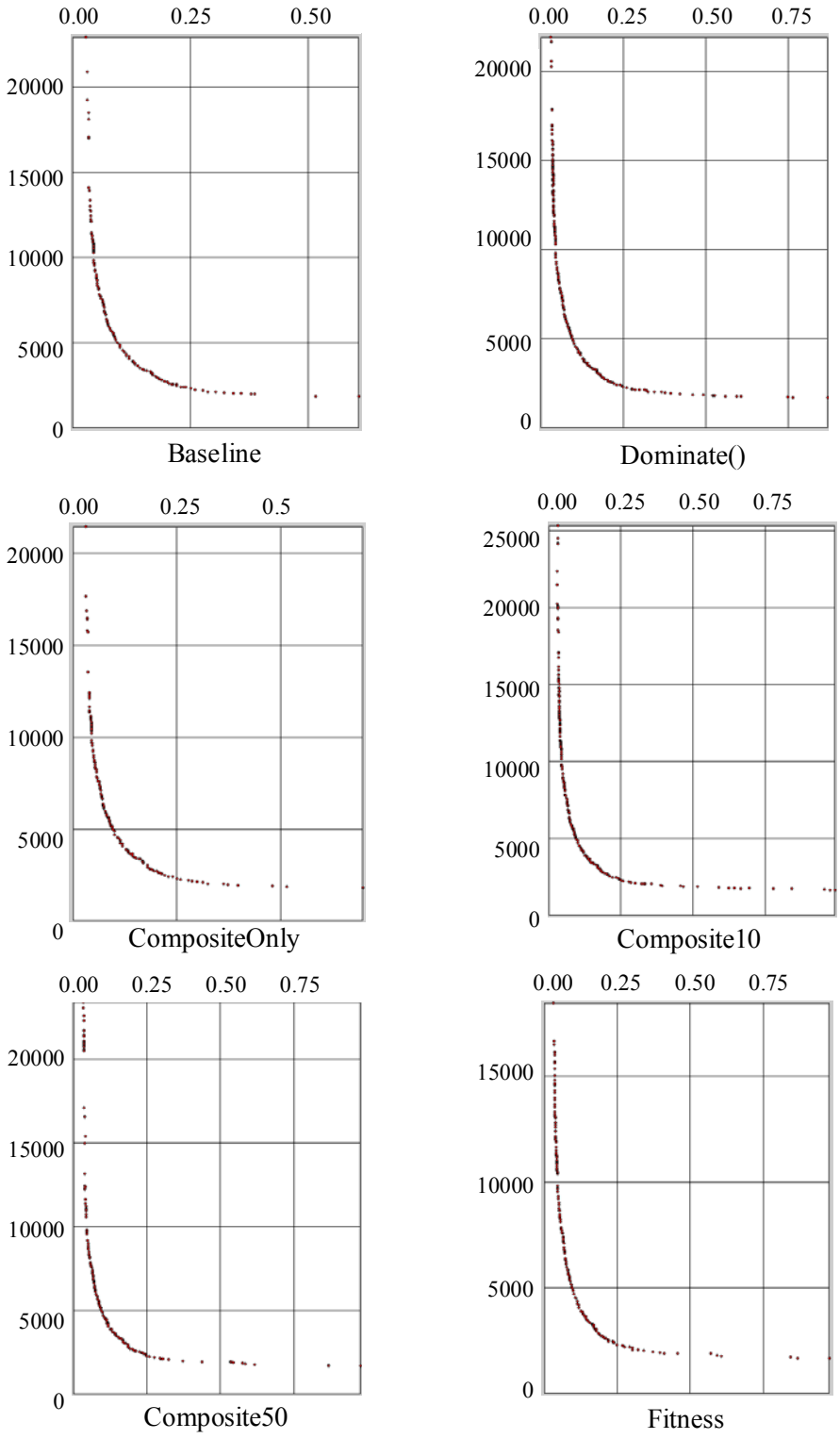


Figure D.15. 60by15 10th generation, rank one Pareto curves for all mechanism trials.

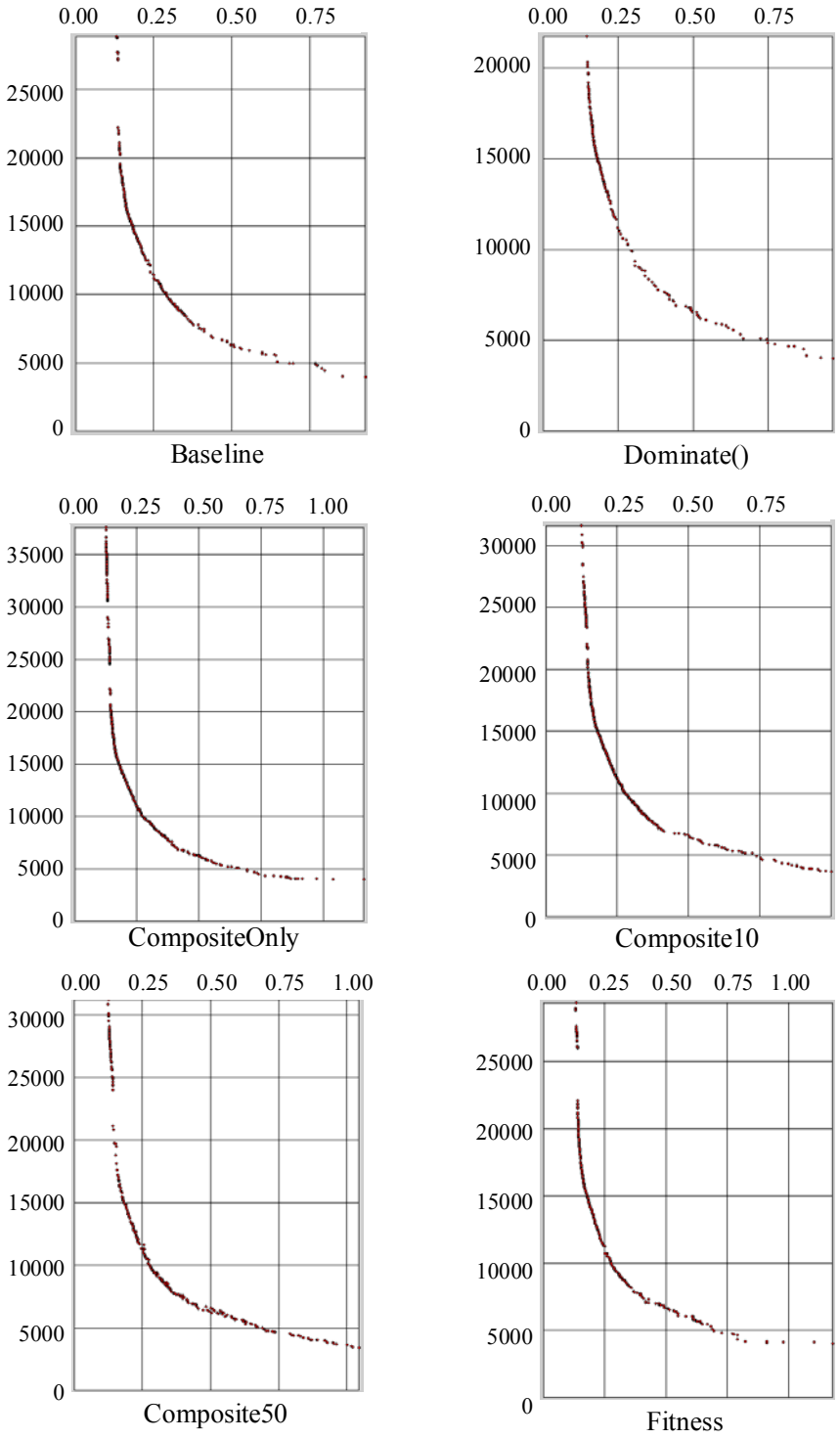


Figure D.16. 60by15Par 10th generation, rank one Pareto curves for all mechanism trials.

Preference Emphasis Trials

The preference emphasis trials concluded that normalizing the **preferred** attribute in the Composite10 definition by two instead of three would better model user preferences. Additional trials for the 60by15, 40by15, and 60by15Par trials were conducted to determine whether or not this change would affect the quality of structural results.

Pareto fronts for these new analyses are shown in Figures D.17 through D.19. The points in red represent results for normalization by two, and the yellow are normalization-by-three data. The superimposed curves illustrate that normalizing by two maintains structurally rigorous designs in the population while encouraging further explanation.

Figures D.20 and D.21 show the final truss forms present in the fronts of 60by15 and 60by15Par. Truss designs for 40by15 are presented in the text.

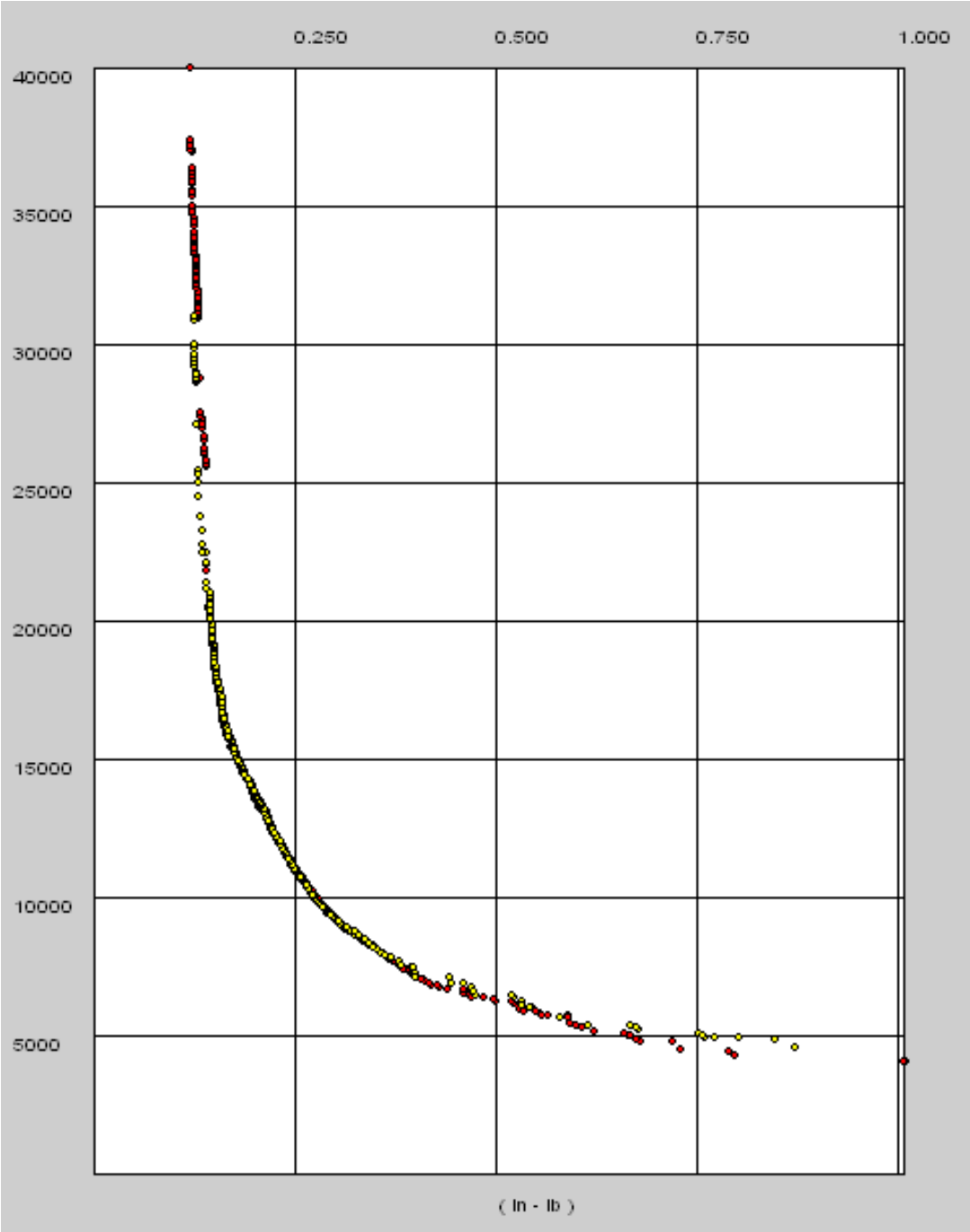


Figure D.17. 10th generation Pareto front for 60by15 with original selections when normalizing by two (red) and three (yellow).

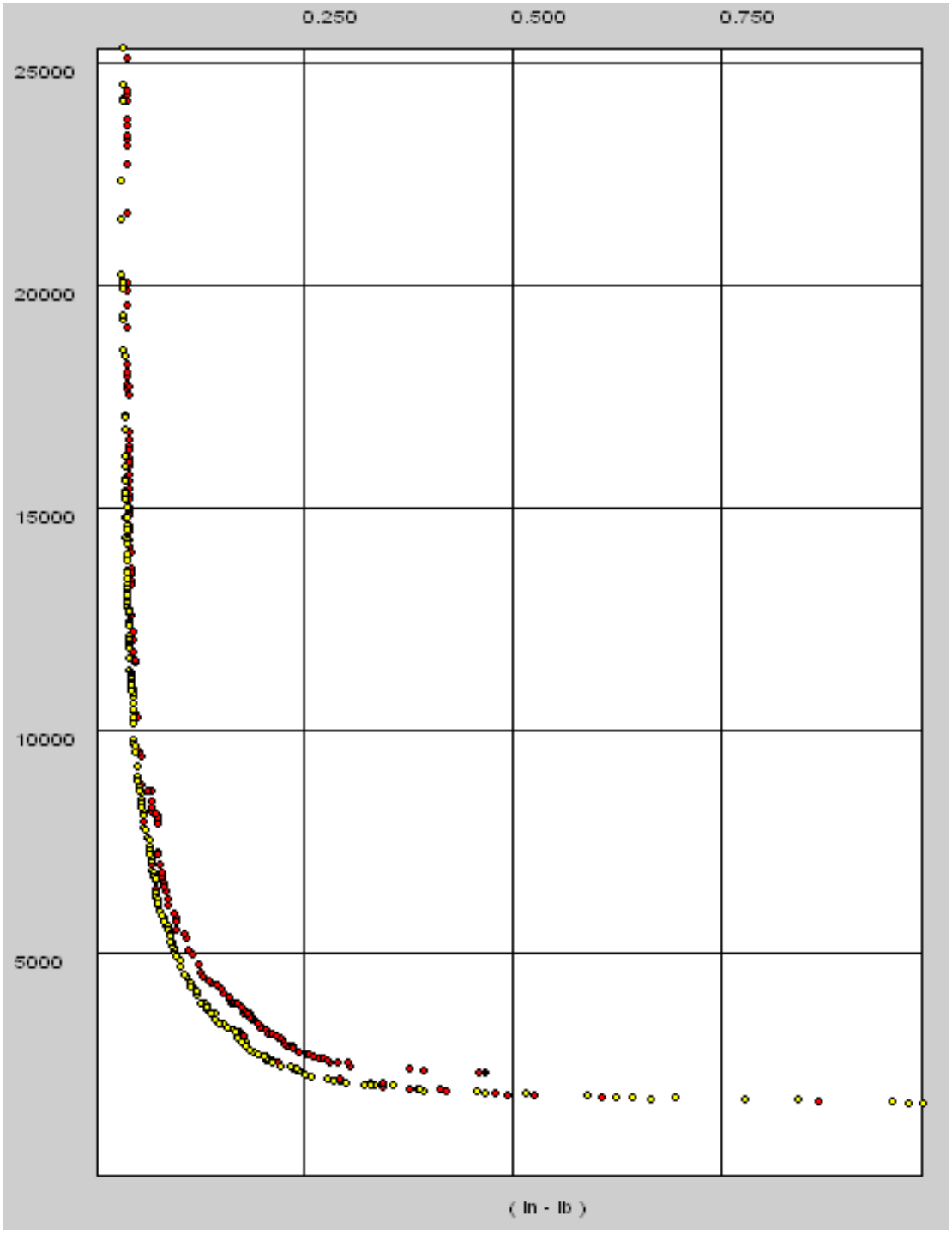


Figure D.18. 10th generation Pareto front for 40by15 when normalizing by two (red) and three (yellow).

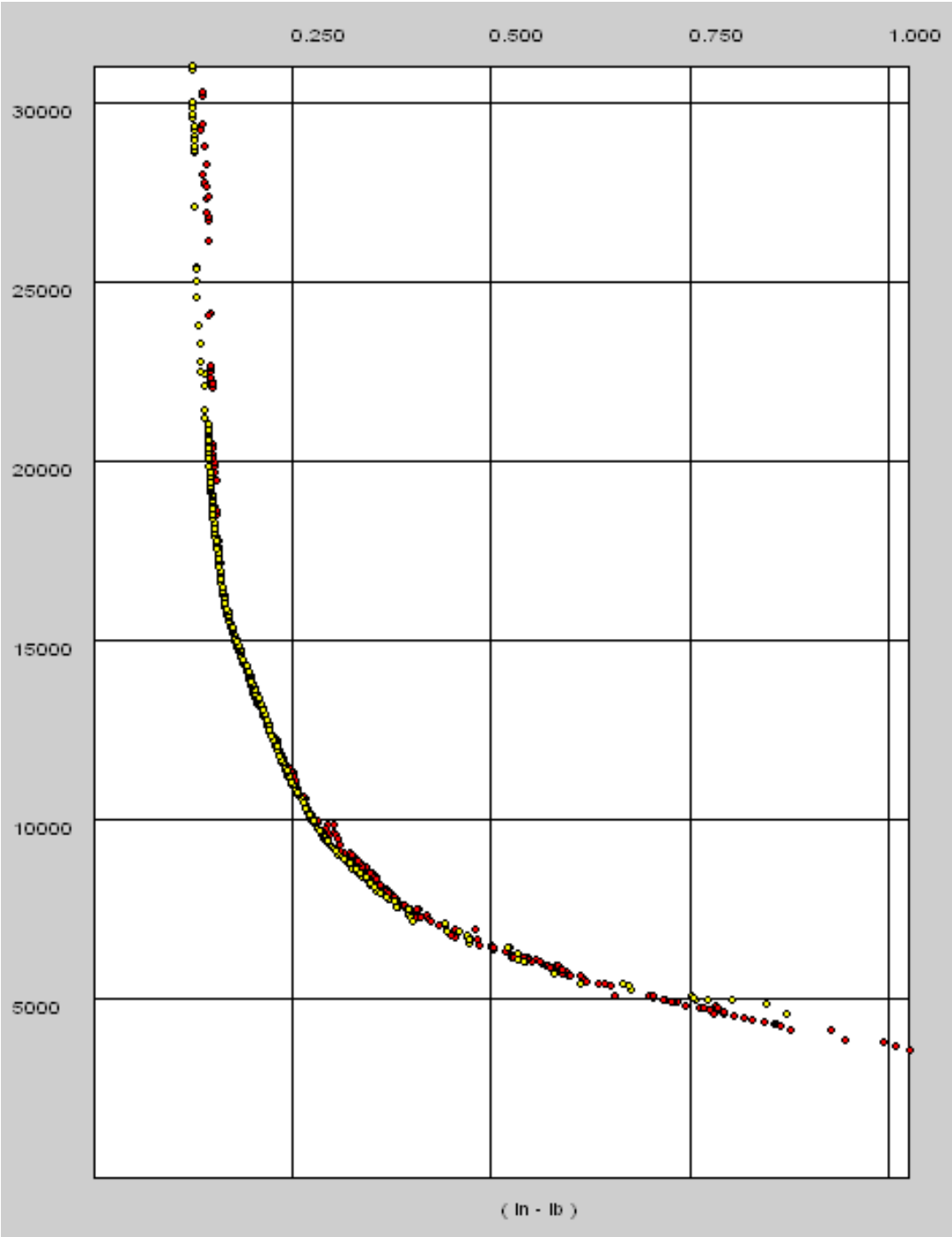


Figure D.19. 10th generation Pareto front for 60by15Par when normalizing by two (red) and three (yellow).

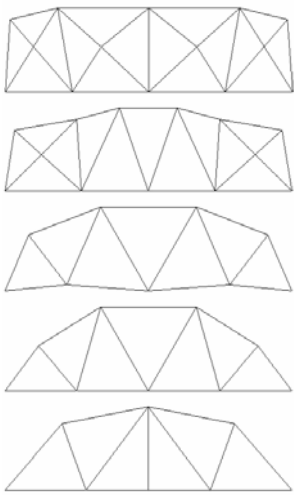


Figure D.20. 10th generation truss designs for 60by15 with original selections when normalizing by two.

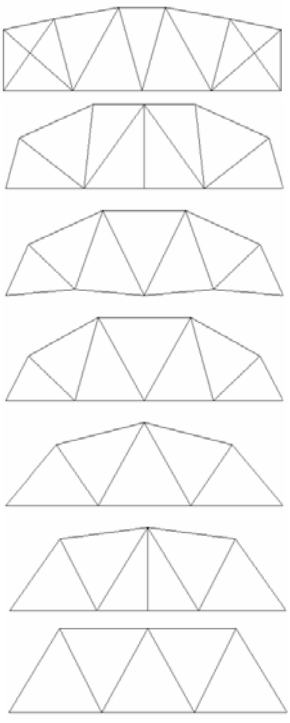


Figure D.21. 10th generation truss designs for 60by15Par when normalizing by two.

Four volunteers provided user selections for population 60by15 that were used as a final examination of how preferences affected the IRR GA. Results for two of these users are presented in the text. The remaining cases are shown here. Figure D.22 shows the preferences identified by Volunteer 3. These preferences were used to generate the 10^{th} generation truss designs and Pareto front shown in Figure D.23. Volunteer 4's preferences appear in Figure D.24, with corresponding results shown in D.25.

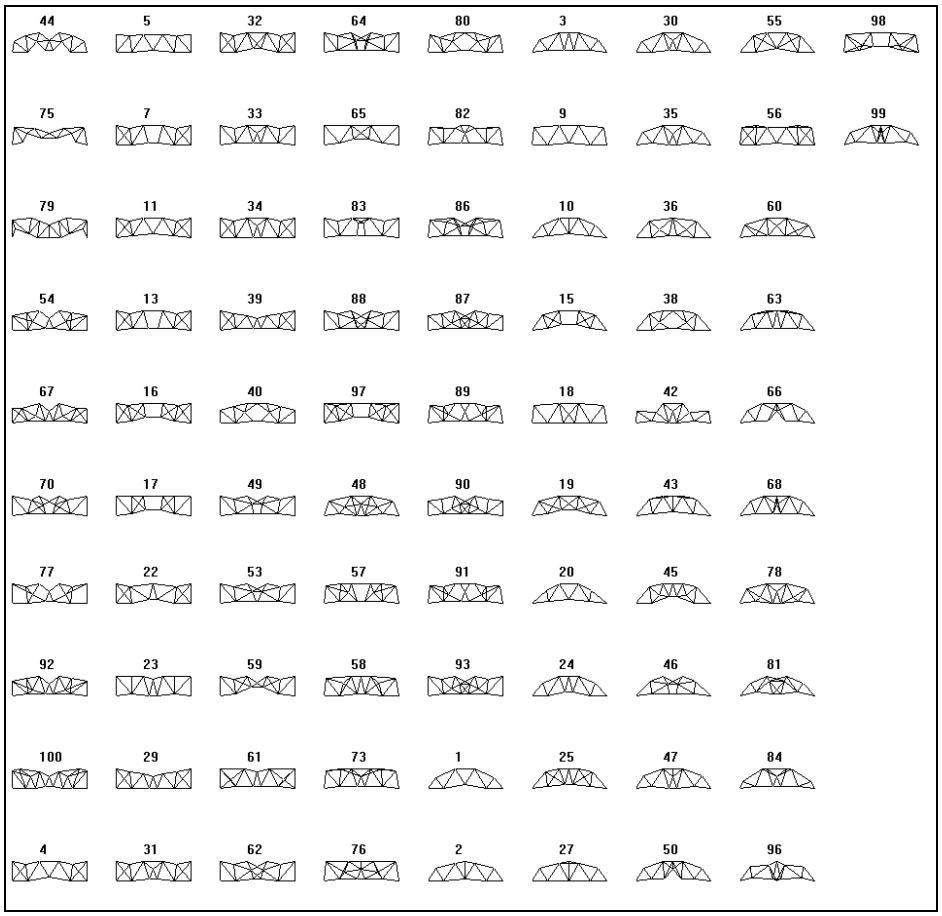


Figure D.22. Preferences identified by Volunteer 3.

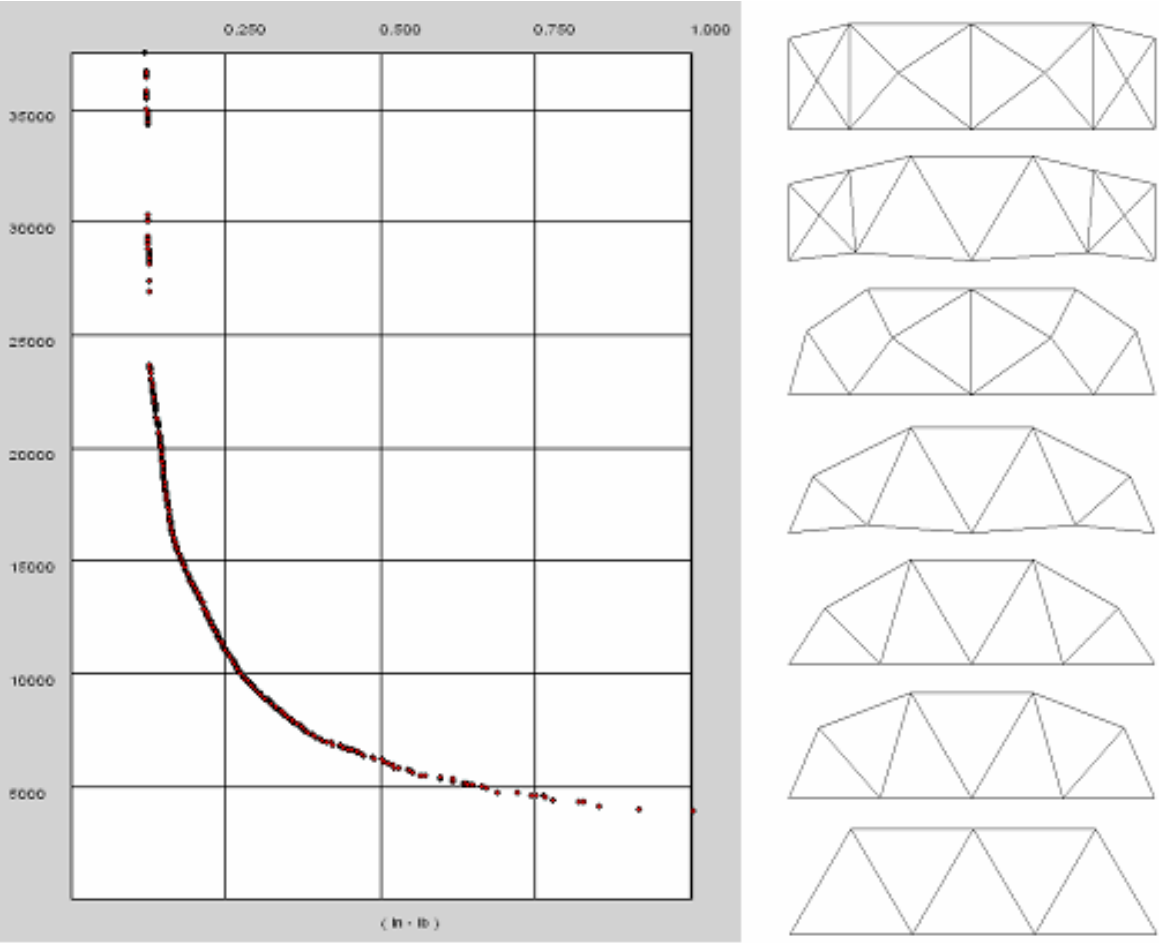


Figure D.23. 10th generation Pareto front and truss designs for Volunteer 3.

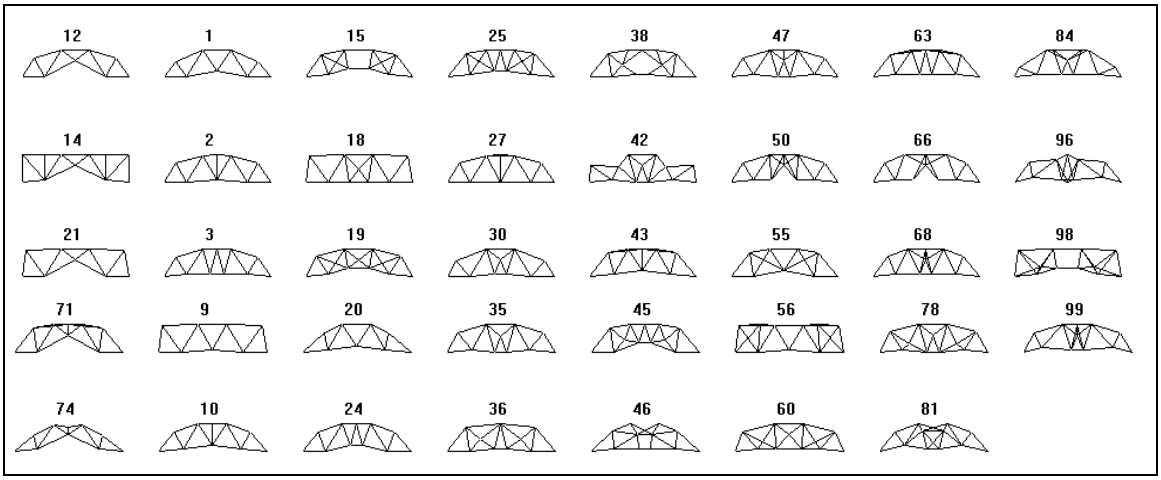


Figure D.24. Preferences identified by Volunteer 4.

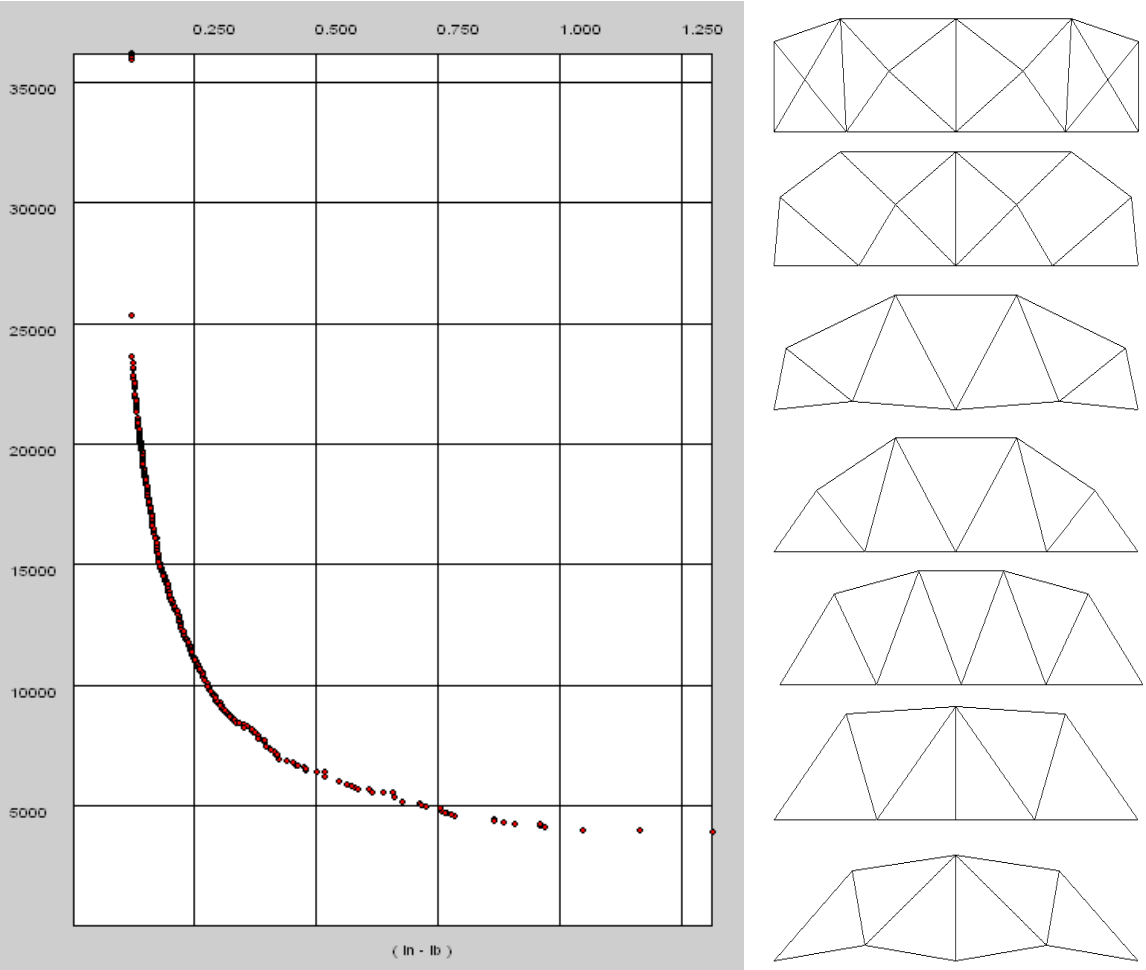


Figure D.24. 10th generation Pareto front and truss designs for Volunteer 4.

VITA

Name:	Breanna Michelle Weir Bailey
Address:	7514 Pine Green Lane Humble, TX 77346
Email Address:	breanna_weir_bailey@yahoo.com
Education:	B.S., Civil Engineering, Texas A&M University M.S., Civil Engineering, University of Illinois Urbana-Champaign Ph.D., Civil Engineering, Texas A&M University